

CONTROLLER, SYSTEM AND METHOD FOR VEHICLE CONTROLPRIORITY CLAIM

[0001] The present application claims the priority benefit of U.S. Provisional Patent Application No. 62/682,537, filed June 8, 2018.

BACKGROUND

[0002] Commuter train networks represent a rapidly growing industry. This rapid growth of rail commuter transit is accompanied by development of Automatic Train Operation (ATO) systems. Overall objectives of these systems are to reduce energy consumption and overall vehicle CO2 emission, improve timetable adherence, lessen wear-and-tear on train drive trains, ensure safe and robust system failure detection and fast subsequent response, and minimize the energy expenditure for a given trip.

BRIEF DESCRIPTION OF THE DRAWINGS

[0003] One or more embodiments are illustrated by way of example, and not by limitation, in the figures of the accompanying drawings, wherein elements having the same reference numeral designations represent like elements throughout. It is emphasized that, in accordance with standard practice in the industry various features may not be drawn to scale and are used for illustration purposes only. In fact, the dimensions of the various features in the drawings may be arbitrarily increased or reduced for clarity of discussion.

[0004] Fig. 1 is a diagram of a motion controlling part of a vehicle, in accordance with one or more embodiments.

[0005] Fig. 2 is a diagram of operations of the motion controlling part of Fig. 1, in accordance with one or more embodiments.

[0006] Fig. 3 is a diagram of a system, in accordance with one or more embodiments.

[0007] Fig. 4 is flow chart of a method, in accordance with one or more embodiments.

[0008] Fig. 5 is flow chart of a method, in accordance with one or more embodiments.

[0009] Fig. 6 is flow chart of a method, in accordance with one or more embodiments.

[0010] Fig. 7 is flow chart of a method, in accordance with one or more embodiments.

[0011] Fig. 8 is a block diagram of a computing platform, in accordance with one or more embodiments.

DETAILED DESCRIPTION

[0012] The following disclosure provides many different embodiments, or examples, for implementing different features of the provided subject matter. Specific examples of components and arrangements are described below to simplify the present disclosure. These are, of course, merely examples and are not intended to be limiting. For example, the formation or position of a first feature over or on a second feature in the description that follows may include embodiments in which the first and second features are formed or positioned in direct contact, and may also include embodiments in which additional features may be formed or positioned between the first and second features, such that the first and second features may not be in direct contact. In addition, the present disclosure may repeat reference numerals and/or letters in the various examples. This repetition is for the purpose of simplicity and clarity and does

not in itself dictate a relationship between the various embodiments and/or configurations discussed.

[0013] Further, spatially relative terms, such as “beneath,” “below,” “lower,” “above,” “upper” and the like, may be used herein for ease of description to describe one element or feature’s relationship to another element(s) or feature(s) as illustrated in the figures. The spatially relative terms are intended to encompass different orientations of an apparatus, object in use or operation, or objects scanned in a three dimensional space, in addition to the orientation thereof depicted in the figures. The apparatus may be otherwise oriented (rotated 90 degrees or at other orientations) and the spatially relative descriptors used herein may likewise be interpreted accordingly.

[0014] In some embodiments, a guideway is a track, rail, roadway, cable, series of reflectors, series of signs, a visible or invisible path, a projected path, a laser-guided path, a global positioning system (GPS)-directed path, an object-studded path or other suitable format of guide, path, track, road or the like on which, over which, below which, beside which, or along which a vehicle is caused to travel. In some embodiments, a vehicle is a railway vehicle (such as, a train) or a road vehicle.

[0015] In some embodiments, a controller, such as a vehicle on-board controller (VOBC), is configured to control movement of a vehicle along a path from a start location to a target location. For this purpose, at each node along the path, a control problem of the vehicle is solved over the remainder of the trip, and the solution, such as a driving profile, is applied to drive the vehicle to the next node. The process is repeated until the target location is reached. In at least one embodiment, the control problem is a convex second order cone problem. In one or more embodiments, a first driving profile of a leading vehicle is

sent to a trailing vehicle to enable a controller of the trailing vehicle to generate a second driving profile for controlling the trailing vehicle to follow the leading vehicle. Advantages in accordance with some embodiments include, but are not limited to, global optimality, robustness of the solution, and enforcement of a safety buffer between vehicles (e.g., trains) following each other on the same path (e.g., same track).

[0016] In some embodiments, the VOBC is implemented by running a background process on every vital machine having safety integrity level 4 (SIL 4) in the system which listens to communication traffic and collects key data as identified by a configuration profile of the VOBC. In some embodiments, SIL 4 is based on International Electrotechnical Commission's (IEC) standard IEC 61508. SIL level 4 means the probability of failure per hour ranges from 10^{-8} to 10^{-9} .

[0017] Fig. 1 is a diagram of a motion controlling part of a vehicle 100, in accordance with one or more embodiments. The motion controlling part of the vehicle 100 includes a controller 110 and a motoring and braking system 120. In some embodiments, the controller 110 comprises at least one processor or computing platform as described with respect to Fig. 8. In at least one embodiment, the controller 110 is implemented as part of a VOBC (not shown) of the vehicle 100. For example, the controller 110 is implemented at least in part by a processor of the VOBC which is programmed to perform operations or functionality of the controller 110 described herein. In at least one embodiment, the controller 110 is implemented in hardware separate from the VOBC of the vehicle 100. For example, the controller 110 is implemented at least in part by a processor outside the VOBC and programmed to perform operations or functionality of the controller 110 described herein. The motoring and braking system 120 includes a propulsion source configured to generate a force or acceleration to move the vehicle along a path. Examples of a propulsion source include, but are not limited to, an engine or an electric motor. The motoring

and braking system 120 further comprises a brake for decelerating and stopping the vehicle 100. Other movements of the vehicle 100 are also effected by the motoring and braking system 120 in various embodiments. For example, in embodiments where steering of the vehicle 100 (e.g., a road vehicle) is possible, the motoring and braking system 120 also includes a steering mechanism for steering the vehicle 100.

[0018] In some embodiments, the controller 110 is configured to apply an optimal control technique to drive the vehicle 100. In at least one embodiment, optimal control involves treating a control problem as a model-based optimization problem where the control law is computed through optimizing a given objective for a known process model (the model of the system, e.g., a vehicle, being controlled) under the presence of one or more constraints. For this purpose, the controller 110, also referred to as an optimal controller in one or more embodiments, is configured to solve an optimization problem based on a current state of the vehicle 100, at least one control objective, and at least one constraint.

[0019] In the example configuration in Fig. 1, the current state of the vehicle 100 includes a current speed (or velocity) V_0 of the vehicle 100, and a current position X_0 of the vehicle 100 on the path. In at least one embodiment, the current speed of the vehicle 100 is acquired by a speed sensor (not shown) in the motoring and braking system 120 or by an independent speed determination system using Doppler radar, camera (video odometry) and/or Light Detection And Ranging (LiDAR) equipment. The current position X_0 of the vehicle 100 is acquired, in one or more embodiments, by a sensor (not shown), such as a camera, a radar, a LiDAR scanner, or a radio frequency (RF) transceiver, that reads corresponding visible, radar, LiDAR or RF data embedded in a sign arranged along the path, in an arrangement known as a communication based train control (CBTC) system. Other configurations where the current state of the vehicle 100 include

other operational parameters of the vehicle 100 and/or how such operational parameters are acquired are within the scopes of various embodiments. For example, in at least one embodiment, the current state of the vehicle 100 includes a current acceleration of the vehicle 100 acquired by an acceleration sensor of the vehicle 100.

[0020] In the example configuration in Fig. 1, at least one control objective and at least one constraint to be used by the controller 110 in formulating and/or solving the optimization problem are included in a trip limits and objectives database 130, a track database 140, and a vehicle configuration database 150. One or more of the trip limits and objectives database 130, track database 140 and vehicle configuration database 150 is/are contained in a computer readable storage medium for access by the controller 110 or a processor thereof, as described with respect to Fig. 8.

[0021] At least one control objective is included in the trip limits and objectives database 130. Examples of at least one control objective include, but are not limited to, minimum amount of time to drive the vehicle from a start location to a target location on the path, minimum amount of energy consumption to drive the vehicle from the start location to the target location, and minimum excessive braking along the path. In at least one embodiment, the controller 110 is configured to solve the optimization problem with a mixed control objective, for example, a mixed time and energy objective. In this situation, the trip limits and objectives database 130 includes corresponding time and energy weights for the mixed time and energy objective. A further example includes a mixed objective of time, energy and excessive braking with three corresponding weights. In some embodiments, the controller 110 is configured to change or adjust, in real-time while the vehicle is travelling on the path, the control objective. In an example, the controller 110 changes the control objective from time to energy, or from time or energy to a mixed time and energy objective.

In another example, the controller 110 adjusts time and/or energy weights in a mixed time and energy objective to adapt the motion of the vehicle to new requirements from the system or a user. Other configurations for selecting and/or updating the control objective(s) are within the scopes of various embodiments.

[0022] Constraints in the optimization problem include one or more trip constraints, track constraints, and vehicle constraints. Trip constraints are limits related to the trip of the vehicle 100 along the path, and are included in the trip limits and objectives database 130. Examples of trip constraints include, but are not limited to, maximum and minimum arrival times at a location on the path, and constraints on braking. Track constraints are limits related to the path along which the vehicle 100 travels, and are included in the track database 140. Examples of track constraints include, but are not limited to, maximum allowable speed limit, friction, traction or grade profile of the path. Vehicle constraints are limits related to the vehicle 100 and are included in the vehicle configuration database 150. In at least one embodiment, vehicle constraints are parameters of a model of the vehicle 100 to be used by the controller 110 in the optimization problem. Examples of vehicle constraints include, but are not limited to, maximum braking force, maximum acceleration (or propulsion) force, vehicle mass, latencies/delays in the motoring and braking systems. Like the control objective, one or more constraints is/are changeable and/or adjustable by the controller 110 in real-time while the vehicle is travelling on the path. Other configurations for selecting and/or updating one or more constraints are within the scopes of various embodiments.

[0023] The controller 110 is configured to solve the optimization problem, using the current state from the motoring and braking system 120 and/or at least one sensor, at least one control objective with any applicable corresponding weight from the trip limits and objectives database

130, and one or more constraints from the trip limits and objectives database 130, track database 140 and vehicle configuration database 150. A solution for optimal control can be feasible or optimal. A feasible solution is one which satisfies the given constraints, for example, driving a vehicle from one location to another without over speeding. In some situations, there are various possible ways, i.e., various feasible solutions, to a control problem. An optimal solution is not only implicitly feasible but is the best feasible solution among the available feasible solutions. An optimal solution is achieved via minimizing or maximizing a control objective, for example, driving a vehicle from one location to another, without over speeding, in a minimum amount of time. This allows the number of feasible (possible) solutions to the control problem to be reduced from a large amount to a single high quality result. In some embodiments, the controller solved an optimization problem when an optimal solution has been achieved.

[0024] In some embodiments, optimization models are configured with an aim to guarantee feasibility and optimality of the resultant solution. This is achievable by presenting the optimal control law in a convex representation which guarantee that both feasible and optimal solutions are attainable for a feasible set of constraints. In addition, convex optimization problems are solvable far quicker and more efficiently than non-convex optimization problems. Nevertheless, in at least one embodiment, the optimization problem to be solved by the controller 110 is a non-convex optimization problem. In at least one embodiment, the optimization problem is a convex optimization problem. In at least one embodiment, the convex optimization problem is a convex second order cone problem. Due to the mathematical nature of the convex second order cone problem, it is possible to determine if the control problem is infeasible (e.g., arrival time, speed limit, or stopping point are impossible to achieve) in a fraction of the time it takes to solve one control iteration. Further, if the control problem is determined to be feasible, the solution is

guaranteed to be optimal. Examples of one or more non-convex optimization problems, convex optimization problems and convex second order cone problems, and their corresponding solutions, are described in Appendix 1 and/or Appendix 2.

[0025] In the example configuration in Fig. 1, the controller 110 is configured to solve the optimization problem in the time domain, in accordance with a clock signal from an internal clock circuit 160. For example, the controller 110 is configured to solve the optimization problem at every control cycle dictated by the clock signal from the clock circuit 160. In at least one embodiment, each control cycle is 100 ms or shorter. As a result, the optimization problem is solved multiple times at a plurality of discretization nodes along the path. Adjacent discretization nodes are arranged at a predetermined time interval corresponding to the clock signal. Physical distances between adjacent discretization nodes are, however, not necessarily equal due to non-linearities of various factors and/or parameters. In some situations, such non-linearities present one or more concerns when attempting to solve optimization problems in the time domain.

[0026] In another example configuration, the controller 110 is configured to solve the optimization problem in the position domain. For example, the controller 110 is configured to solve the optimization problem at every discretization node among a plurality of discretization nodes arranged at a predetermined distance along the path. Adjacent discretization nodes are therefore equally spaced. In some situations, discretization of the path and solving optimization problems in the position domain address one or more concerns met when attempting to solve optimization problems in the time domain.

[0027] Fig. 2 is a diagram of operations 200 of the motion controlling part in Fig. 1, in accordance with one or more embodiments. The operations 200 include a plurality of control iterations 210, 220, 230,... 250 along a path 202 of the vehicle from a start location 204 to a

target location 206. In the example configuration in Fig. 2, the path 202 is a track or rail on which the vehicle which is a train travels, the start location 204 is a start station, and the target location 206 is a stop station. However, other configurations are within the scopes of various embodiments. For example, in at least one embodiment, the start location 204 and/or the target location 206 is/are locations on the path 202 between which optimization of the travel of the vehicle is to be performed. The speed of the vehicle at the start location 204 and/or the target location 206 is not necessarily zero, i.e., the vehicle does not necessarily start or end its travel at the start location 204 or target location 206. The control iterations 210, 220, 230,... 250 are performed in the time domain or the position domain as described herein. In the time domain, the control iterations 210, 220, 230,... 250 correspond to various discretization nodes arranged at a predetermined time interval corresponding to the clock signal (control cycle) of the clock circuit 160. In the position domain, the control iterations 210, 220, 230,... 250 correspond to various discretization nodes distributed at a predetermined distance along the path 202.

[0028] At each of the control iterations 210, 220, 230,... 250, the controller 110 is configured to, based on a current state of the vehicle, generate a travel plan for a remainder of the path 202 from a current position of the vehicle to the target location 206. For example, at the first control iteration 210 at the start location 204, the current state of the vehicle includes the current speed V_0 and the current position X_0 as described with respect to Fig. 1. Based on this current state, the controller 110 is configured to perform an optimization problem to achieve at least one control objective under one or more constraints, as described with respect to Fig. 1. The solution of the optimization problem includes at least one driving profile 216 which includes for each of discretization nodes 217, 218 on a remainder of the path 202 between the current position, i.e., the start location 204, and the target location 206, at least one of time, position, speed or

acceleration of the vehicle. At the first control iteration 210, because the current position of the vehicle is the start location 204, the remainder of the path 202 is the path 202 itself. The time, position, speed or acceleration of the vehicle at each of the discretization nodes 217, 218 in accordance with the driving profile 216 is optimal to achieve the at least one control objective under the one or more constraints of the optimization problem. For example, when the optimization problem is performed in the time domain, the driving profile 216 includes a position profile including an optimal position of the vehicle at each discretization node (i.e., at each control cycle). In another example, when the optimization problem is performed in the position domain, the driving profile 216 includes a time profile including an optimal time when the vehicle arrives at each discretization node. In a further example, the driving profile 216 includes a speed or acceleration profile including an optimal speed or acceleration of the vehicle at each discretization node. One or more of the driving profile 216, e.g., a position profile, a speed profile and an acceleration profile, together define a travel plan for the vehicle over the remainder of the path 202. Other driving profiles besides the described time profile, position profile, speed profile and acceleration profile are within the scopes of various embodiments.

[0029] The controller 110 is further configured to control the motoring and braking system 120 of the vehicle to execute the travel plan until a next control iteration. For example, the first element of the driving profile 216, i.e., at least one of time, position, speed or acceleration of the vehicle, at the first discretization node 217 corresponding to the start location 204 is applied, in the form of a control vector Γ_1 in Fig. 1, by the controller 110 to control the motoring and braking system 120. At least one embodiment, the control vector represents an input force command to cause the motoring and braking system 120 to generate a propulsion or braking force to achieve the optimal time, position, speed or acceleration in the first element of the driving profile 216. The

motoring and braking system 120 maintains the propulsion or braking force until the next control iteration 220 (or next discretization node 227) is reached, at which time the controller 110 is configured to solve the optimization problem again.

[0030] Specifically, at the second control iteration 220, the new current state of the vehicle includes a new current speed and a new current position X_1 . The remainder of the path 202 is shortened, and includes a section between the new current position X_1 and the target location 206. The controller 110 is configured to perform the optimization problem again, based on the new current state and any update/adjustment to the control objective and/or constraints. The solution is a new driving profile for the remainder of the path 202 between the new current position X_1 and the target location 206. The first element of the new driving profile at the discretization node 227 is applied by the controller 110 to control the motoring and braking system 120 to move the vehicle until the next control iteration 230 (or next discretization node 237) is reached, at which time the controller 110 is configured to solve the optimization problem again.

[0031] Specifically, at the third control iteration 230, the current state of the vehicle is updated. The remainder of the path 202 is shortened, and includes a section between the new current position X_2 and the target location 206. The controller 110 is configured to perform the optimization problem again, based on the updated current state and any update/adjustment to the control objective and/or constraints. The solution is a further driving profile for the remainder of the path 202 between the new current position X_2 and the target location 206. The first element of the further driving profile at the discretization node 237 is applied by the controller 110 to control the motoring and braking system 120 to move the vehicle until the next control iteration (or next discretization node) is reached, at which time the controller 110 is configured to solve the optimization problem again. The above-described operations are repeatedly performed until the

vehicle reaches the target location 206. As a result, in at least one embodiment, an optimal travel of the vehicle at every discretization node along the path 202 is achievable, while taking into consideration various changes or adjustments in the control objective (e.g., from time to energy), trip constraints (e.g., updated maximum or minimum arrival time), track constraints (e.g., updated speed limit), and/or vehicle constraints (e.g., changed vehicle mass due to passengers getting on or off the vehicle).

[0032] Fig. 3 is a diagram of a system 300, in accordance with one or more embodiments. The system 300 is configured to control a plurality of vehicles 310, 320, one following another along a common path 330. As shown by the direction of motion arrow in Fig. 3, the vehicle 310 is a leading vehicle and the vehicle 320 is a trailing or following vehicle that follows the leading vehicle 310 on path 330. The leading vehicle 310 comprises a first controller 311, a motoring and braking system 312, a position sensor 313, a velocity (or speed) sensor 314, and an acceleration sensor 315. The trailing vehicle 320 comprises a second controller 321, a motoring and braking system 322, a position sensor 323, a velocity (or speed) sensor 324, and an acceleration sensor 325. In at least one embodiment, at least one of the first controller 311 and second controller 321 corresponds to the controller 110, at least one of the motoring and braking system 312 and motoring and braking system 322 corresponds to the motoring and braking system 120, and at least one of the position sensor 313, velocity sensor 314, acceleration sensor 315, position sensor 323, velocity sensor 324, acceleration sensor 325 corresponds to a respective sensor described with respect to Fig. 1. The first controller 311 communicates over a communication 317 with a wayside sign 331 arranged along the path 330, using a sensor (not shown), such as a camera, a radar, a LiDAR scanner, or an RF transceiver, in an CBTC arrangement as described herein. the second controller 321 communicates over a communication 327 with a wayside sign 332 arranged along

the path 330, using a sensor (not shown), such as a camera, a radar, a LiDAR scanner, or an RF transceiver, in an CBTC arrangement as described herein.

[0033] The first controller 311 comprises a first communication circuit 318, and the second controller 321 comprises a second communication circuit 328. The first communication circuit 318 and the second communication circuit 328 are configured to perform communication 340 with each other to transmit a first driving profile 316 from the first controller 311 to the second controller 321 as described herein. In at least one embodiment, the communication 340 is a one-way communication to reduce the workload and/or circuit complexity. In the example configuration in Fig. 3, the communication 340 is an over-the-air communication, for example, an RF channel, and the first communication circuit 318 and second communication circuit 328 comprise corresponding RF transmitting and receiving circuits and/or antennas, respectively. Other configurations of the communication 340 are within the scopes of various embodiments. For example, the communication 340 is a two-way communication in at least one embodiment.

[0034] The first controller 311 associated with the leading vehicle 310 is configured to generate the first driving profile 316, and control movement of the leading vehicle 310 along the path 330 based on the generated first driving profile 316. In at least one embodiment, the first controller 311 is configured to generate the first driving profile 316 by solving an optimization problem using the current state of the leading vehicle 310 acquired from at least one of the position sensor 313, velocity sensor 314, acceleration sensor 315, with respect to at least one control objective and one or more constraints as described with respect to Figs. 1-2. Other configurations are within the scopes of various embodiments. For example, the first driving profile 316 is generated in a manner different than solving an optimization problem, e.g., the first driving profile 316 is estimated or generated by another ATO technique. In at least one embodiment, the first

controller 311 is configured to apply the first driving profile 316 to control the motoring and braking system 312 to drive the leading vehicle 310, as described with respect to Figs. 1-2.

[0035] The second controller 321 associated with the trailing vehicle 320 is configured to receive the first driving profile 316 generated by the first controller 311 over the communication 340. The second controller 321 is further configured to generate a second driving profile 326 based on the received first driving profile 316, and control movement of the trailing vehicle 320 along the path 330 and following the leading vehicle 310 based on the generated second driving profile 326. In at least one embodiment, the second controller 321 is configured to generate the second driving profile 326 by solving an optimization problem using the current state of the trailing vehicle 320 acquired from at least one of the position sensor 323, velocity sensor 324, acceleration sensor 325, with respect to at least one control objective and one or more constraints as described with respect to Figs. 1-2. However, in addition to the one or more constraints described with respect to the trip limits and objectives database 130, track database 140 and vehicle configuration database 150, the second controller 321 solve the optimization problem using the first driving profile 316 as an additional constraint. Specifically, the first driving profile 316 includes information on a position of the leading vehicle 310 at a given time. The additional constraint used by the second controller 321 in solving the optimization problem ensures that the trailing vehicle 320 is at least a distance d away from the position of the leading vehicle 310 at the given time.

[0036] In at least one embodiment, at each control iteration, the second controller 321 receives the first driving profile 316 and uses the first driving profile 316 as a constraint to generate the second driving profile 326 through solving an optimization problem. In at least one embodiment, at each control iteration, the first controller 311 generates the first driving profile 316 through solving an optimization problem, and sends the first driving profile 316 to the second controller 321.

In some embodiments, the optimization problems solved by the first controller 311 and the second controller 321 differ in at least one control objective and/or one or more constraints. For example, a control objective of the first controller 311 is time, whereas a control objective of the second controller 321 is a mixed time and energy objective. Examples of optimization problems solved by the first controller 311 and/or the second controller 321 to control the trailing vehicle 320 to follow the leading vehicle 310 at an intended separation are described in Appendix 1 and/or Appendix 2.

[0037] In at least one embodiment, more than two vehicles are included in a fleet of vehicles following each other on the path 330. In an example, a further trailing vehicle (not shown) follows the trailing vehicle 320. A third controller (not shown) associated with the further trailing vehicle is configured to receive the second driving profile 326 generated by the second controller 321, generate a third driving profile based on the received second driving profile 326, and control movement of the further trailing vehicle along the path 330 and following the trailing vehicle 320 based on the generated third driving profile, in a manner similar to the described operations of the second controller 321. In at least one embodiment, the leading vehicle 310 is catching up with a further leading vehicle (not shown) and becomes a trailing vehicle. In this situation, the first controller 311 receives a driving profile from the further leading vehicle and uses the received driving profile as a constraint in solving an optimization problem to update the first driving profile 316, in a manner similar to the described operations of the second controller 321. An example of this situation is met where the path 330 is a closed loop, and the leading vehicle 310 being a first vehicle of the fleet is catching up with a last vehicle and becomes a trailing vehicle. The controller on each vehicle in the fleet repeatedly updates its driving profile based on a driving profile received from the vehicle immediately ahead, and sends the updated driving profile to the vehicle

immediately behind. As a result, the vehicles in the fleet follow each other at an intended separation in an optimal manner.

[0038] The above description includes examples, and other configurations are within the scopes of various embodiments.

[0039] For example, in some embodiments, the controller 110, 311 or 321 is configured to, at each control iteration, estimate a parameter of the vehicle, and use the estimated parameter in generating the travel plan for the vehicle over the remainder of the path. For example, instead of using a constant parameter retrieved from the vehicle configuration database 150, the parameter of the vehicle is constantly updated through estimation which further improves optimality of the optimization problem solution, i.e., the driving profile of the vehicle. In at least one embodiment, the parameter of the vehicle to be updated includes the mass or weight of the vehicle, which changes, for example, when passengers get on or off the vehicle. In another example, the parameter of the vehicle to be updated is a parameter that changes as the vehicle ages through normal wear-and-tear. Examples of parameter estimation techniques are described in Appendix 2.

[0040] In some situations, time delays are observed in operations of the motoring and braking system 120. For example, when the control vector is outputted by the controller 110 to the motoring and braking system 120 to drive the vehicle, there are time delays from when the control vector is received until the propulsion source, such as, a motor, or the brake generates the corresponding motoring or braking force corresponding to the control vector. The motoring time delay associated with the propulsion source and the braking time delay associated with the brake are similar in some situations, and different in other situations. One or more embodiments take such time delays into account to further improve optimality of the solution, i.e., the driving profile

generated through optimization. Examples of embodiments that consider time delays of a motoring and braking system are described in Appendix 3. In the described example embodiments, at each control iteration, the controller determines, between the current position and the target location, a predicted position of the vehicle corresponding to at least one delay of the motoring and braking system. The controller next solves an optimization problem to determine a first part of the travel plan between the predicted position and the target location. The controller then determines, based on the first part of the travel plan, a second part of the travel plan between the current position and the predicted position, for example, through interpolation.

[0041] In some embodiments, to limit wear-and-tear on the braking system, hard braking is reserved for an emergency or a modelling error. For example, the constraints to be used for optimization comprises a first brake constraint and a second brake constraint. The second brake constraint corresponds to a harder brake operation (or force) than the first brake constraint. In at least one embodiment, the second brake constraint corresponds to the performance limit, e.g., the maximum braking force, of the braking system, whereas the first brake constraint corresponds to a softer braking operation (or force) to improve passenger comfort, stop the vehicle without overshoot in the event of a modelling error, or adjust to a new shorter stopping distance. The controller is configured to solve the optimization problem constrained by the first (soft) brake constraint. The objective function of the controller penalizes braking above the soft braking constraint/level. In response to a determination that the optimization problem constrained by the first brake constraint is feasible, the controller is configured to control the motoring and braking system of the vehicle based on the solution of the optimization problem. On the other hand, in response to a determination that the optimization problem constrained by the first brake constraint is infeasible (i.e., a modelling error exists and extra braking is required to satisfy the constraints of the

optimization problem), the controller is configured to solve the optimization problem constrained by the second brake constraint corresponding to the harder brake operation. As discussed herein, in embodiments where the optimization problem is a convex second order cone problem, it is possible to determine if the control problem is infeasible (e.g., arrival time, speed limit, or stopping point are impossible to achieve) in a fraction of the time it takes to solve one control iteration, and therefore, it is possible to trigger the emergency brake thus saving valuable time and increasing the chances of a safe arrival.

[0042] In some embodiments, to optimize the control of a fleet of vehicles, only a one-way communication is needed from a leading vehicle to a trailing vehicle to drive the entire fleet optimally. In at least one embodiment, each vehicle is running at the optimal Automatic Speed Control (ASC), and transmitting its forecasted optimal control profile, e.g., a driving profile including at least one of position, speed, velocity, etc..., for its entire remaining distance to the target location to its immediately following vehicle. The following vehicle adds the received driving profile into the constrain set for its controller for optimization at the following vehicle. Each following vehicle assumes that it is possible for the leading vehicle to come to an immediate stop at any point in its forecasted trajectory to account for any possible way in which the vehicle is to stop (normal stop, crash, etc...). In at least one embodiment, this allows to optimally control an entire fleet of vehicles with a basic control architecture and allows to move vehicles closer together, achieve higher passenger comfort, reduce more wear-and-tear on the vehicles and conserve significantly more energy than is possible with moving-block CBTC.

[0043] A practical application of some embodiments is CBTC systems configured to operate in ATO, driverless or unattended modes of operation where the VOBC commands the requested propulsion and braking effort along with coasting to the vehicle's control systems. A further practical application of some embodiments is for off-line generation of speed profiles to improve the capabilities of existing speed controllers. In at least one embodiment, the controller is used for larger scale control room routing and management of complete vehicle fleets or other mobile logistical planning. While trains are a practical application of some embodiments, at least one embodiment has a practical application in road vehicles, such as autonomous cars, travelling on a guideway, especially in the form of a fleet of vehicle one following another as described herein.

[0044] Compared to alternative approaches, one or more embodiments provide one or more advantages discussed herein. Specifically, an alternative approach involves ad-hoc techniques which are tuned to a specific vehicle. The underlying architecture is usually a PID (Proportional-Integral-Derivative) where a current present measurement set (speed, acceleration, position) is fed back into a controller which plans only its present control decision without factoring in the future consequences of its action. The PID attempts to track a per-defined speed profile. Thus, a present control decision may cause an over speed further down the line. Since a PID is a linear controller that is not designed for non-linear systems, it is usually patched with other ad-hoc techniques to work to an acceptable level in an advanced non-linear system. This patch-on-patch design model for a controller is neither sustainable nor robust. From the controller tuning point of view, it is difficult to tune a linear controller on a vehicle mainly due to

the complex behavior and system interactions of the vehicle. Additionally, it is difficult to understand and debug control properties of such a linear controller. These factors make it difficult to ensure that a linear controller can cope under all possible conditions. The ad-hoc control strategies are not using optimization techniques and are difficult to design in a manner where the required constraints such as time-table adherence and energy consumption levels can be satisfied. Such controllers depend heavily on their tuning and are often additionally limited by the simplifications to the vehicle model which are required for their function. As control objectives and models become more complex, alternative control architectures become less able to keep up and eventually become completely unable to meet the vehicle control requirements.

[0045] To address one or more concerns observed in the ad-hoc techniques, a further alternative approach involves controllers based on optimal control. Such optimal controllers largely suffer from the inability to solve the control law fast enough to be usable as automatic speed controller. Therefore, due to the large numerical computation requirements for these controllers, they are utilized as offline planning algorithms for existing controllers, or as driver assist systems. For example, a driver assist system computes switching points between motoring/speed-hold/coast/brake states, and informs the driver when they should change modes. It takes a few seconds to re-optimize the switching curve profile whereas an ASC in ATO operation needs to operate at frequencies in the order of 10Hz. The alternative approach also greatly simplifies the underlying vehicle and trip model and does not account for the possibility of running behind another vehicle. The alternative approach is not real-time capable and cannot

drive a vehicle on its own but instead relies on the driver following the switching curves that is generates.

[0046] In contrast, one or more embodiments allow for real-time optimal and adaptive control of the vehicle to achieve goals defined by the operator while satisfying defined constraints in a driverless or unattended modes of operation. In at least one embodiment, the controller repeats optimization each and every control cycle in real-time and returns a solution substantially faster than 100ms.

[0047] Since the type of controller in some embodiments is model based, it is possible to mathematically determine the most optimal course of action (e.g., driving profile) for the vehicle. In addition, at least one embodiment allows for hard constraints (including, but not limited to, speed limits, arrival time limits, control limits, etc...) to be numerically specified and mathematically guarantees that the control law will adhere to the hard constraints assuming that the constraints are feasible. Therefore, control objectives such as minimum time or constrained arrival time and energy optimal driving strategies are possible. These capabilities are not available in the described alternative approaches.

[0048] In some embodiments, the real-time capability to re-optimize the entire driving profile up to the target location, e.g., the next station, at every control iteration allows an energy and time minimized driving strategy. The controller in accordance with some embodiments does not need to be supplied with a speed curve and is therefore much easier to deploy in the field. In addition, it is robust (less sensitive) to changes in the trip requirements. For instance, if a vehicle running a PID controller

must be delayed for any reason to a next platform the PID controller will bring the vehicle to a complete stop, wait the delay time, and then resume following its reference speed curve. This is greatly inefficient, uncomfortable to passengers, and detrimental to the railway operators reputation. One or more embodiments mitigate this by immediate re-optimizing the entire driving profile over the remainder of the intended travel and continued driving in the newly calculated (adaptive) most optimal manner.

[0049] The controller in some embodiments is a model-based controller which within itself knows the entire model of the vehicle it is controlling. Therefore, the controller plans the most efficient and safe courses of action, but is also easy to adapt to different vehicles. This removes the need for the ad-hoc patch-on-patch work-flow that is present with controllers in accordance with alternative approaches, and allows for a streamlined ease to configure and understand the controller in at least one embodiment.

[0050] The convexity of the optimization problem handled by a controller in at least one embodiment and/or the algorithmic design provide(s) the controller's capability of real-time operation. In contrast to optimal vehicle speed control methods in alternative approaches, the controller in at least one embodiment is fully capable of real-time operation on minimal hardware.

[0051] The controller in some embodiments also presents an inter-vehicle communication and optimization method allowing many vehicles to follow one close to another in an optimal manner, e.g., during high service hours. The controller in some embodiments allows for much smoother and more energy efficient driving than is currently possible with controllers in alternative approaches.

[0052] Some embodiments further provide one or more advantages including, but not limited to, huge energy savings and guaranteed timetable adherence along due to optimal constrained driving, greatly reduced tuning effort for deployment in the field, and self-tuning and internal optimization strategy permitting the controller to adapt to the vehicle and does not deteriorate in performance as the vehicle ages.

[0053] Fig. 4 is flow chart of a method 400, in accordance with one or more embodiments. In at least one embodiment, the method 400 is performed by one or more of the controller 110, first controller 311 and second controller 321.

[0054] At operation 415, a convex optimization is performed to generate a travel plan for the vehicle along a path. The convex optimization comprises a second order cone problem. For example, the controller 110 solves a convex optimization problem which is a second order cone problem, as described with respect to Fig. 1.

[0055] At operation 425, a motoring and braking system of the vehicle is controlled to execute the generated travel plan to move the vehicle along the path. For example, the motoring and braking system 120 is controlled to execute the travel plan generated by the controller 110 to move the vehicle along the path, as described with respect to Fig. 1.

[0056] Fig. 5 is flow chart of a method 500, in accordance with one or more embodiments. In at least one embodiment, the method 500 is performed by one or more of the controller 110, first controller 311 and second controller 321.

[0057] As indicated at loop operation 505, the method 500 includes operations 515 and 525 which are repeatedly performed for each of a plurality of control iterations along a path

from a start location to a target location. For example, the control iterations in the method 500 corresponding to the control iterations 210, 220, 230,..., 250, which are performed at various locations $X_0, X_1, X_2, \dots, X_N$ along the path 202 from a start location 204 to a target location 206, as described with respect to Fig. 2.

[0058] At operation 515, a travel plan for a remainder of the path from a current position of the vehicle to the target location is generated. For example, at the control iteration 220, a travel plan for a remainder of the path 202 from a current position X_1 of the vehicle to the target location 206 is generated, as described with respect to Fig. 2.

[0059] At operation 525, a motoring and braking system is controlled to execute the generated travel plan until a next control iteration. For example, the first element 227 of the travel plan at the discretization node 227 is applied by the controller 110 to control the motoring and braking system 120 to move the vehicle until the next control iteration 230 (or next discretization node 237) is reached. In at least one embodiment, the above-described operations are repeatedly performed until the vehicle reaches the target location 206, as described with respect to Fig. 2.

[0060] Fig. 6 is flow chart of a method 600, in accordance with one or more embodiments. In at least one embodiment, the method 600 is performed by the first controller 311 and second controller 321. The left-side part of the flow chart in Fig. 6 includes operations on a trailing vehicle, such as the trailing vehicle 320 in Fig. 3. The right-side part of the flow chart in Fig. 6 includes operations on a leading vehicle, such as the leading vehicle 310 in Fig. 3.

[0061] On the leading vehicle side, at operation 615, a first driving profile is generated for the leading vehicle. For example, the first controller 311 of the leading vehicle 310 generates a first driving profile 316 for the leading vehicle 310, as described with respect to Fig. 3. In at least one embodiment, the first driving profile 316 is generated by solving an optimization problem as described with respect to Fig. 1. In at least one embodiment, the first driving profile 316 is otherwise generated, without solving an optimization problem. The first driving profile 316 is transmitted to the trailing vehicle, as indicated at 630.

[0062] At operation 625, movement of the leading vehicle along the path is controlled based on the generated first driving profile. For example, the first controller 311 controls the motoring and braking system 312 of the leading vehicle 310 to move the leading vehicle 310 along the path, as described with respect to Fig. 3.

[0063] On the trailing vehicle side, at operation 645, the first driving profile generated by the first controller is received. For example, the second controller 321 of the trailing vehicle 320 receives the first driving profile 316 over a communication 340, as described with respect to Fig. 3.

[0064] At operation 655, a second driving profile is generated based on the received first driving profile. For example, the second controller 321 solves an optimization problem with the first driving profile 316 as a constraint, to obtain the second driving profile 326, as described with respect to Fig. 3.

[0065] At operation 665, movement of the trailing vehicle along the path and following the leading vehicle is controlled based on the generated second driving profile. For example, the second controller 321 controls the motoring and braking system

322 of the trailing vehicle 320 to move the trailing vehicle 320 along the path 330 and following the leading vehicle 310, as described with respect to Fig. 3.

[0066] Fig. 7 is flow chart of a method 700, in accordance with one or more embodiments. In at least one embodiment, the method 700 is performed by the first controller 311 and second controller 321. The left-side part of the flow chart in Fig. 7 includes operations on a trailing vehicle, such as the trailing vehicle 320 in Fig. 3. The right-side part of the flow chart in Fig. 7 includes operations on a leading vehicle, such as the leading vehicle 310 in Fig. 3.

[0067] On the leading vehicle side, as indicated at loop operation 705, the method 700 includes operations 715 and 725 which are repeatedly performed for each of a plurality of control iterations along a path from a start location to a target location. Likewise, on the trailing vehicle side, as indicated at loop operation 735, the method 700 includes operations 745, 755 and 765 which are repeatedly performed for each of a plurality of control iterations along the path from the start location to the target location. In at least one embodiment, the control iterations of the leading vehicle corresponds in a one-to-one relationship to the control iterations of the trailing vehicle. Other configurations are within the scopes of various embodiments. In an example, the control iterations for at least one of the leading vehicle 310 or the trailing vehicle 320 in the method 700 corresponding to the control iterations 210, 220, 230,... 250, which are performed at various locations X0, X1, X2,...XN along the path 202 from a start location 204 to a target location 206, as described with respect to Fig. 2.

[0068] On the leading vehicle side, at operation 715, a travel plan for a remainder of the path from a current position of the leading vehicle 310 to the target location is

generated by the first controller 311, in a manner similar to operation 515 of the method 500.

[0069] At operation 725, the motoring and braking system 312 of the leading vehicle 310 is controlled to execute the generated travel plan until a next control iteration, in a manner similar to operation 525 of the method 500.

[0070] As indicated at 730, a driving profile corresponding to the generated travel plan of the leading vehicle 310 is transmitted to the trailing vehicle 320.

[0071] On the trailing vehicle side, at operation 745, the driving profile is received from the first controller.

[0072] At operation 755, a travel plan for a remainder of the path from a current position of the trailing vehicle 320 to the target location is generated by the second controller 321, in a manner similar to operation 515 of the method 500, with the received driving profile as a constraint.

[0073] At operation 765, the motoring and braking system 322 of the trailing vehicle 320 is controlled to execute the generated travel plan until a next control iteration, in a manner similar to operation 525 of the method 500.

[0074] One or more advantages achievable in the system 100 and/or system 300 by one or more of the controller 110, first controller 311 and second controller 321 is/are also achievable in one or more of the methods 400, 500, 600 and 700.

[0075] The described methods include example operations, but they are not necessarily required to be performed in the order shown. Operations may be added, replaced, changed order, and/or eliminated as appropriate, in accordance with the spirit and scope of embodiments of the disclosure. Embodiments that combine different features and/or different embodiments are within the scope of the disclosure and will be apparent to those of ordinary skill in the art after reviewing this disclosure.

[0076] Fig. 8 is a block diagram of a computing platform 800, in accordance with one or more embodiments. In some embodiments, one or more of the controller 110, first controller 311, second controller 321, a VDBC of any one or more of the vehicle 100, leading vehicle 310 and trailing vehicle 320 is/are implemented as one or more computing platform(s) 800.

[0077] The computing platform 800 includes a specific-purpose hardware processor 802 and a non-transitory, computer readable storage medium 804 storing computer program code 803 and/or data 805. The computer readable storage medium 804 is also encoded with instructions 807 for interfacing with the vehicle on which the computing platform 800 is installed. The processor 802 is electrically coupled to the computer readable storage medium 804 via a bus 808. The processor 802 is also electrically coupled to an I/O interface 810 by the bus 808. A network interface 812 is electrically connected to the processor 802 via the bus 808. The network interface 812 is connected to a network 814, so that the processor 802 and/or the computer readable storage medium 804 is/are connectable to external elements and/or systems via the network 814.

[0078] In some embodiments, the processor 802 is a central processing unit (CPU), a multi-processor, a distributed processing system, an application specific integrated circuit (ASIC), and/or a suitable hardware processing unit.

[0079] In some embodiments, the processor 802 is configured to execute the computer program code 803 and/or access the data 805 stored in the computer readable storage medium 804 in order to cause the computing platform 800 to perform as one or more components of the system 100 and/or system 300, and/or to perform a portion or all of the operations as described in one or more of the methods 400, 500, 600 and 700. For example, the computer program code 803 includes one or more algorithm or model for causing the processor 802 to solve optimization problems or estimate a parameter of the vehicle. The computer readable storage medium 804 includes one or more of the trip limits and objectives database 130, track database 140 and vehicle configuration database 150 with at least one control objective and one or more constraints for the optimization problems and/or parameter estimation.

[0080] In some embodiments, the processor 802 is hard-wired (e.g., as an ASIC) to cause the computing platform 800 to perform as one or more components of the system 100 and/or system 300, and/or to perform a portion or all of the operations as described in one or more of the methods 400, 500, 600 and 700.

[0081] In some embodiments, the computer readable storage medium 804 is an electronic, magnetic, optical, electromagnetic, infrared, and/or a semiconductor system (or apparatus or device). For example, the computer readable storage medium 804 includes a semiconductor or solid-state memory, a magnetic tape, a removable computer diskette, a random access memory (RAM), a read-only memory (ROM), a rigid magnetic disk, and/or an optical disk. In some embodiments using optical disks, the computer readable storage medium 804 includes a compact disk-read only memory (CD-ROM), a compact disk-read/write (CD-R/W), and/or a digital video disc (DVD).

[0082] In some embodiments, the I/O interface 810 is coupled to external circuitry. In some embodiments, the I/O interface 810 includes a keyboard, keypad, mouse, trackball, trackpad, and/or cursor direction keys for communicating information and commands to processor 802. In at least one embodiment, the I/O interface 810 is coupled to a communication circuit for vehicle-to-vehicle communication as described with respect to Fig. 3.

[0083] In some embodiments, the network interface 812 allows the computing platform 800 to communicate with network 814, to which one or more other computing platforms are connected. The network interface 812 includes wireless network interfaces such as BLUETOOTH, WIFI, WIMAX, GPRS, or WCDMA; or wired network interface such as ETHERNET, USB, or IEEE-1394. In some embodiments, the method 300A and/or method 300B is/are implemented in two or more computing platforms 800, and various executable instructions and/or data are exchanged between different computing platforms 800 via the network 814.

[0084] By being configured to execute some or all of functionalities and/or operations described with respect to Figs. 1-7, the computing platform 800 enables the realization of one or more advantages and/or effects described with respect to Figs. 1-7.

[0085] In some embodiments, a controller for a vehicle comprises a processor configured to repeatedly perform operations, at each control iteration among a plurality of control iterations along a path of the vehicle from a start location to a target location, by solving an optimization problem. The operations include, based on a current state of the vehicle, generating a travel plan for a remainder of the path from a current position of the vehicle to the target location. The operations further include controlling a motoring and braking system of the vehicle to execute the generated travel plan until a next control iteration among the plurality of control iterations.

[0086] In some embodiments, a system for controlling a plurality of vehicles along a common path comprises a first controller associated with a leading vehicle among the plurality of vehicles, and a second controller associated with a trailing vehicle among the plurality of vehicles. The first controller is configured to generate a first driving profile, and control movement of the leading vehicle along the path based on the generated first driving profile. The second controller is configured to receive the first driving profile generated by the first controller, generate a second driving profile based on the received first driving profile, and control movement of the trailing vehicle along the path and following the leading vehicle based on the generated second driving profile.

[0087] In some embodiments, a method of controlling a vehicle comprises performing convex optimization to generate a travel plan for the vehicle along a path, and controlling a motoring and braking system of the vehicle to execute the generated travel plan to move the vehicle along the path. The convex optimization comprises a second order cone problem.

[0088] It will be readily seen by one of ordinary skill in the art that the disclosed embodiments fulfill one or more of the advantages set forth above. After reading the foregoing specification, one of ordinary skill will be able to affect various changes, substitutions of equivalents and various other embodiments as broadly disclosed herein. It is therefore intended that the protection granted hereon be limited only by the definition contained in the appended claims and equivalents thereof.

Appendix 1

A Convex Real-time Capable On-line Optimal Controller for a Commuter Train

Abstract—This paper proposes an on-board optimal controller that drives a train between two stations in an optimal time, energy efficient, or mixed-objective manner, while adhering to a set of system-specific constraints. To this end at each step along the track, the train control problem is formulated and solved as a constrained optimization problem over the remainder of the trip, while utilizing the latest train sensor data. The optimization problem is a convex Second Order Cone Program. It incorporates knowledge of the track profile and relevant velocity and propulsion/braking constraints in the computation of the optimal propulsion/braking commands. It features an option to enforce a safety buffer between the train and another leading train on the track. The resulting convex optimization problem can be efficiently solved using a simple numerical solver, ensuring global optimality, and robustness of the solution. Real-time performance and simulated closed-loop control results are presented, for a realistic vehicle and advanced trip model on desktop and embedded computer architectures.

Index Terms—Optimal Control, Minimum Time Control, Minimum Energy Control, Commuter Train, Numerical Optimization, Convex Optimization, Second Order Cone Program, Real-time, Sparse Optimization

I. INTRODUCTION

Global commuter train networks drive an everincreasing 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 [1] and in China the government is presently spending a record 1.25% percent of its GDP on investments in new rail infrastructure [2]. 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 [3]. Rail transportation is so prolific that the railway transportation industry has accounted for 2% of the total energy consumption in Europe alone [4].

This rapid growth of rail commuter transit has motivated development of reliable and efficient on-board control systems for passenger locomotives. The overall objectives of these controllers are to reduce energy consumption and overall vehicle CO_2 emission, improve timetable adherence, and lessen wear-and-tear on train drive trains [4], [5]. Automatic Train Operation (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) [6].

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 [7]. Conventional ATO control techniques in the railway industry mainly rely on PID based strategies, where the ATO controller attempts to follow a speed trajectory between a set of stations [8]. Due to the lack of any process knowledge and a simple gain multiplier feedback, the PID is a reactive controller as opposed to a planning controller and is a compromise [9]. In recent years, there has been an increasing interest in moving away from classical control strategies to more advanced control architectures, to achieve increasingly complicated control objectives [10] and enhance operation safety [11].

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 with simple control objectives [10]. In principle, optimal controllers should be able to adhere to state and input limits for an entire control horizon [12]. 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. Additionally, optimal control techniques would have full knowledge of a locomotive, allowing for more meaningful control.

While the theory of optimal control is well established, many practical optimal control problems especially those with input and state constraints are notoriously difficult to solve analytically. A possible solution is to formulate the problem in the discrete-time domain and convert it to a conventional optimization, which can then be solved numerically. This approach has been popular in dealing with processes with slow cycles, such as chemical engineering processes where control decision are made at slow update rates in the order of minutes and hours, allowing for real-time computation of a solution [12], [7]. Research in the optimal control of commuter trains has been pursued using both numerical optimization and analytical control formulation. Analytical methods are generally based on simplified and often unrealistic system operational model so that a solution can be obtained. The existing numerical approaches allow for more realistic vehicle and trip models but usually suffer from unreliable global minima, and are computationally too involved for real-time control [13].

In the analytical realm of control, Ichikawa was the first to solve the optimal control problem on a simplified train model through Pontryagin's maximization principle [14]. Since then, analytical methods have greatly extended but still generally suffer from an inability to find a solution when subjected to more advanced nonlinear vehicle models, track and trip constraints [15].

Numerous research has focused on treating the train control problem as time optimal or energy optimal switching problems where the transitions for a MOTOR-SPEEDHOLD-COAST-BRAKE strategy are presented as an analytical optimization. *Howlett et al.* formulated a theoretical optimal energy control strategy that was based on local optimization around grade transition [16] and then global optimization to determine the optimal control switching points along a track. The controller was not presented in a closed-loop online form and required several numeric perturbations of the optimal profile to come up with a feasible solution with simplistic grade models and constant speed limits [17], [16]. *Howlett et al.* also formulated an energy optimal off-line 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. Methods of analytically or numerically solving this optimal control problem were not addressed.

Matsura and Miyake [18] 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 they used was significantly advanced involving complicated speed limits, grades, regenerative braking and even advanced dynamic modeling for cross-car forces. 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. The numerical approaches in [19], [7] have faced similar limitations. To reduce the on-line computation time, they propose 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. Such an approach would require very substantial off-line computations and would be unable to adapt to changes in trip conditions on the fly.

This paper presents a novel convex formulation of the train optimal control problem which can be solved at sufficiently high update rates on an embedded computer, to obtain a globally optimal solution for real-time control. Unlike other numerical optimal control techniques in the literature, the method presented here uses realistic train model, track grade profiles, and operational constraints, e.g. speed limits, arrival times, etc., both with and without sharing the track with another vehicle. In addition, the convexity of the problem formulation assures global optimality, thus supporting the notion of a unique control solution, when feasible [20]. The controller is implemented with a direct interface to a robust second-order cone problem solver (ECOS) and is tested with an advanced train model, sensor and model uncertainty on non-trivial guideway and trip designs on x86-64 machines and embedded ARM computers. The timing results indicate that

the proposed controller in its current form can execute in real-time for on-line optimal control of a commuter train using an embedded computer.

The rest of this paper is organized as follows. In Section II, vehicle dynamics models are developed. In Section III, an optimal controller is formulated as non-convex and convex optimization problems. Section IV discusses implementation and presents test results with the convex controller. Section V concludes this paper and presents potential future research directions.

TABLE I: Notation

$k \in \mathbb{Z} \geq 0$	Discretization size
$V_{max} \in \mathbb{R}^k \geq 0$	Max allowable speed limit vector (m/s)
$Y_0, X_0 \in \mathbb{R}$	Current train speed (m/s) and position (m)
$V_f \in \mathbb{R}$	Speed at the arrival point (0m/s for station a stop).
$G \in \mathbb{R}^{k-1}$	Discrete grade acceleration (m/s ²)
$g = 9.8$	Gravitational Constant (m/s ²)
$X \in \mathbb{R}^k$	Discrete position vector (m)
$d_x \in \mathbb{R}^{k-1} \geq 0$	Diff of the discrete position vector
$w_t, w_e \in \mathbb{R} \geq 0$	Time and energy weights for the mixed objective
$w_{brk}, w_{ovr} \in \mathbb{R} \geq 0$	Overbraking and overspeeding objective weights
$T_{min}, T_{max} \in \mathbb{R} \geq 0$	Min/Max limits on arrival time into a station
$T_{start} \in \mathbb{R}$	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 two trains (m)
$\theta(x)$	Track inclination (rad)
f_{ctrl}	Closed loop control frequency (Hz)
$\epsilon \in \mathbb{R} > 0$	Very small positive number
$A(v, l)$	Traction and braking function (m/s ²)
$C(v)$	Motion resistance function (m/s ²)
$F_{brk} \in \mathbb{R} \leq 0$	Maximum braking force
$F_{prop} \in \mathbb{R} \geq 0$	Maximum propulsion force
$c_0, c_1, c_2 \in \mathbb{R}$	Davis equation coefficients
$r_0, r_1, r_2 \in \mathbb{R}$	Tractive loss quadratic equation coefficients
$U \in \mathbb{R}^{k-1}$	Control input force(N)
$u \in \mathbb{R}^{k-1}$	Control input acceleration ($u = U/m$) (m/s ²)
$t, v, a \in \mathbb{R}^k$	Time (s), speed (m/s) and acceleration (m/s ²)
$\rho \in \mathbb{R}^{k-1}$	Inverse speed (s/m)
$\alpha \in \mathbb{R}^{k-1}$	Position domain acceleration (1/s)
$\gamma, \gamma_+ \in \mathbb{R}^{k-1}$	Γ/v , and $\max(\Gamma/v, 0)$ respectively (1/s)
$c_t, c_e, c_{brk}, c_{ovr} \in \mathbb{R}$	Time, trip energy, brake reserve, and overspeed costs
$f(x) \in \mathbb{R}$	Minimization Objective
$m \in \mathbb{R}$	Vehicle mass
$D_{st} \in \mathbb{R}$	How close to stay away from V_{max} (m/s)
$D_{brk} \in \mathbb{R}$	How much brake to reserve(m/s ²)
$b, h \in \mathbb{R}^{k-1}$	Soft speed and brake violation variables
E_f, X_s, T_f	Simulated energy (J/kg), position (m) and time (t)
$P^*, D^* \in \mathbb{R}$	Optimal primal and dual objective values
μ, λ	Dual equality and inequality variables

II. 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.

In the time domain, the train control optimization horizon is the discretized time vector t for $t = 0 \dots t_h$ with a fixed sampling rate [7]. A 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, grade and speed limit models thus requiring a reformulation into the position domain.

It has been shown by *Albrecht et. al* that the advanced distributed vehicle model for a multicar train can be in total represented as a point-mass model in the position domain [21]. Thus the physics of a train are formulated as a discrete position point mass system that has, at position $X[i]$ (for $i = 1 \dots k$), an applied propulsion or braking command force Γ_i (N), motion resistance $C(v_i)$, and track grade acceleration forces $G[i]$ [22], [21]. In the position domain, the vehicle dynamics can be modeled as follows:

$$t_{i+1} = t_i + \frac{d_x}{v_i} \quad (1a)$$

$$v_{i+1} = v_i + \alpha_i d_x \quad (1b)$$

$$\alpha_i = \frac{\partial v_i}{\partial x} = \frac{a_i}{v_i} = \frac{A(v_i, \Gamma_i) + C(v_i) + G[i]}{v_i} \quad (1c)$$

$u = A(v, \Gamma)$ represents the acceleration due to motoring or braking input forces in m/s^2 that is felt by the vehicle.

The control horizon in the position domain is along the train position vector X [23]. Therefore the entire distance between two stations can be discretized into a constant vector X of length k , with a finite differences vector d_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. Therefore, given a position domain formulation of a controller, the arrival time (t_k) can be minimized as an optimization objective.

Energy is also simple to calculate in this domain as it is trivially $\Gamma^T d_x$ - 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 α (1/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 [23], [7], [24]. The grade acceleration $G[i]$ for a point mass vehicle is generated as lookup table for the function of the vertical track inclination as:

$$G[i] = g \sin(\theta(X[i])) \quad (2)$$

1) *Davis Equation*: The motion resistance model $C(v_i)$ can be expressed as the Davis equation with constant coefficients c_0, c_1, c_{v2} [22]. 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_1, c_{v2} [22], either through the optimal parameter estimation of vehicle test data or semi-empirically through the Armstrong and Swift resistance model equation [25]:

$$C(v_i) = c_0 + c_1 v_i + c_{v2} v_i^2 \quad (3)$$

2) *Traction Loss*: Electric motor tractive effort loss is a common behavior of electric motors on locomotives and usually looks like Fig. 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. There is general agreement in the literature 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 $R(v)$ [26]. As a complete model, given that the mass of the vehicle is m and the maximum braking force is F_{min} , the traction acceleration is modeled as:

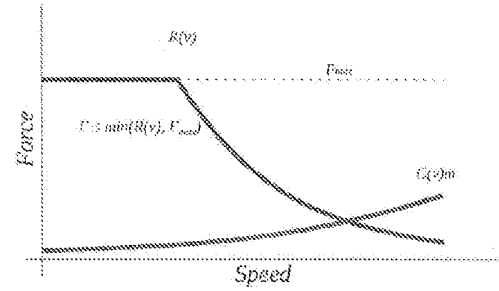
$$A(v_i, \Gamma_i) = \frac{\min(\Gamma_i, R(v_i))}{m} \quad \text{for } F_{min} \leq \Gamma \leq F_{max} \quad (4)$$

In the railway literature, there are many 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:

$$R(v_i) = r_0 + r_1 v_i + r_2 v_i^2 \quad (5)$$

where r_0, r_1, r_2 are constant polynomial coefficients provided by a vehicle manufacturer or through parameter estimation from experimental data.

Fig. 1: Traction Loss and Vehicle Resistance



3) *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 go to infinity which is a substantial problem in control. This can be resolved by constraining the vehicle velocity to remain larger than an arbitrarily small value (ϵ) as: $v \geq \epsilon$, where ϵ is small enough to practically mean that the train is stopped at $v = \epsilon$.

III. OPTIMIZATION FORMULATION

A. Non-Convex Position Domain Optimization 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 - a very challenging task due to advanced train models, complex guideway constraint and advanced objectives. 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^+). This process is repeated until the train safely arrives at the target.

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:

$$\underset{\substack{t, v \in \mathbb{R}^k \\ \Gamma, \Gamma^+ \in \mathbb{R}^{k-1}}}{\text{minimize}} \quad f(z) = w_t t_k + w_e \frac{d_x^T \Gamma^+}{m} \quad (6a)$$

$$0 \leq \Gamma^+ \quad \Gamma \leq \Gamma^+ \quad (6b)$$

$$\epsilon \leq v_i \leq V_{max}[i] \quad i = 2, \dots, k-1 \quad (6c)$$

$$v_1 = V_0 \quad v_k = V_f = 0 \quad (6d)$$

$$t_{k+1} = t_i + \frac{d_x[i]}{v_i} \quad i = 1, \dots, k-1 \quad (6e)$$

$$t_1 = T_{start} \quad (6f)$$

$$T_{min} \leq t_k \leq T_{max} \quad (6g)$$

$$F_{min} \leq \Gamma \leq F_{max} \quad (6h)$$

$$\Gamma_i \leq r_0 + r_1 v_i + r_2 v_i^2 \quad i = 1, \dots, k-1 \quad (6i)$$

$$v_{k+1} = v_k + (c_0 + G[t] + \frac{\Gamma_i}{m} + c_1 v_k + c_2 v_k^2) \frac{d_x[i]}{v_i} \quad (6j)$$

for $i = 1, \dots, k-1$

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 smart tuning for a spectra of control objectives simply by the selection of weights to achieve a pure time, energy or mixed objective. This enables the controller, for example, in 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.

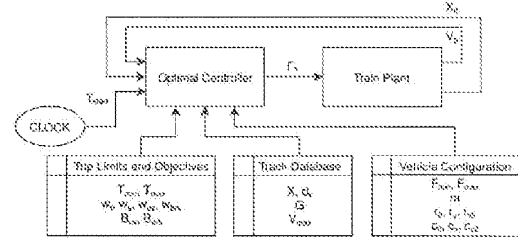
An additional auxiliary variable Γ^+ is added to extract only the positive control applied to the train for use in the energy calculation in the objective. The total trip energy per unit mass can then be evaluated as $(d_x^T \Gamma^+)/m(J)$. The unit energy is more useful than the actual trip energy since the objective function should not place more emphasis in energy minimization for a heavier vehicle than it would for a light vehicle. The time objective of the cost function minimizes the arrival time t_k . The mixed control objective is defined in Eqn. 6a where Γ^+ is defined as an epigraph transformation and is equivalent to the motoring input applied on the vehicle (Eqn. 6b). 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 avoid control singularities in the optimization and prevent driving in reverse (Eqn. 6c). The initial velocity is then constrained as the current measured velocity of the train (V_0), defining a terminal constraint. 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. Therefore, the terminal velocity constraints are set to be Eqn. 6dm. The time at each segment along the track can be evaluated as in Eqn. 6e (the same as Eqn. 1a) 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 the recursive (Eqn. 6f). Thus, the arrival time at $X[k]$ can be constrained from above and below by minimum and maximum allowable trip traversal time limits (Eqn. 6g). The braking limit and constant traction region of the traction loss motoring model ($A(v_i, \Gamma_i)$) are defined as the inequalities (Eqn. 6h). The nonlinear region of the traction loss model ($R(v)$) is also introduced as an inequality constraint (Eqn. 6i). Finally, the dynamic model for the vehicle velocity (Eqn. 1c) is constrained, resulting in the recursive equality (Eqn. 6j).

The full control problem can be defined as a closed-loop version of Eqn. 6. In this way a horizon x is discretized along the entire track spanning from the start station to the stop station, resulting in a constant N , d_x , V_{max} , and G . The closed-loop control can then be determined via solving the optimization problem in Eqn. 6 for the optimal Γ^* that satisfies the constraints. The first element of Γ^* (Γ_1) is then extracted and applied to the vehicle. In the next feedback cycles, X_0 , T_{start} and V_0 would be measured and fed back resulting in shortening the control horizon for X , d_x , V_{max} , and G .

The closed-loop control architecture is depicted in Fig. 2. The controller resolves the optimization in Eqn. 6 at each $\frac{1}{f_{rate}}$ seconds until the train arrives and stops at the target. The closed-loop re-optimization should help deal with modeling uncertainties and external disturbances all while maintaining optimal driving behavior and rapidly adapting to changing problem specifications.

Fig. 2: The Architecture of the Closed-loop Optimal Controller.



The track, model parameters and trip objective weights are all loaded from databases at each optimization cycle. Therefore as the track horizon shortens, the controller pulls up the remaining speed and grade profiles for the remainder of the trip from the database. Should the vehicle model or trip specifications change during operation, the updated parameters can be supplied to the optimal controller which can then react optimally. 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.

B. Numerical Solutions to the Non-Convex Control Problem

The non-convex formulation was realized and tested with the IPOPT solver (a large scale nonlinear programming interior point solver) compiled with the MUMPS linear subsolver [27]

and compiled under Linux with the efficient and parallelisable openBLAS linear algebra routine under Linux 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.

In practical testing, the IPOPT solver could solve small scale problems with a coarse horizon to a small duality gap and relatively quickly. Regardless, the performance was not reliable in closed loop in terms of optimization time, solution quality and solver feasibility. Therefore, since real-time control is a primary design criteria of proposed closed-loop control architecture, a convex reformulation of the optimization problem will be introduced next.

C. Convex Optimization Formulation

The optimization problem in Eqn. 6 is non-convex, and therefore, finding a globally optimal solution for it is not guaranteed. This issue may be mitigated to a degree 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 [28]. 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 [29]. Convex problems largely benefit from their highly scalable nature [30], [31], 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.

The problem in Eqn. 6 can be approximated by a convex a second-order cone problem with some reformulation and relaxation, allowing not only for an approach that does not suffer from the above issues but also allows for invaluable extensions to the problem. This is achieved 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:

$$\underset{\rho, \gamma, \gamma^+, t \in \mathbb{R}^{k-1}}{\text{minimize}} \quad f(x) = c_t + c_v = w_t t_k + \frac{d_x^T(w_v \gamma^+)}{m} \quad (7a)$$

$$c \leq v_i \leq V_{max}[i] \quad i = 2, \dots, k-1 \quad (7b)$$

$$x_1 = V_0 \quad v_k = V_f = 0 \quad (7c)$$

$$t_1 = T_{start} \quad (7d)$$

$$T_{min} \leq t_k \leq T_{max} \quad (7e)$$

$$t_{i+1} = t_i + \rho v_i d_x[i] \quad i = 1, \dots, k-1 \quad (7f)$$

$$\left\| \frac{2}{v_i - \rho v_i} \right\|_2 \leq v_i + \rho v_i \quad i = 1, \dots, k-1 \quad (7g)$$

$$F_{min} \rho \leq \gamma \leq F_{max} \rho \quad (7h)$$

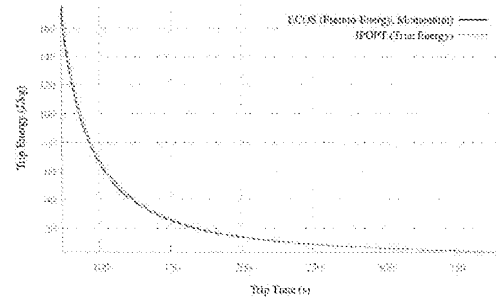
$$0 \leq \gamma^+ \quad \gamma \leq \gamma^+ \quad (7i)$$

$$\gamma_i \leq r_0 \rho_i + r_v + r_{v2} v_i \quad i = 1, \dots, k-1 \quad (7j)$$

$$v_{i+1} = v_i + ((c_0 + G[r])\rho_i + \frac{\gamma_i}{m} + c_v + c_{v2} v_i) d_x[i] \quad i = 1, \dots, k-1 \quad (7k)$$

The objective function remains unchanged from Eqn. 6a for the minimum time cost. Since Γ is no longer an optimization variable in the convex reformulation, an energy efficient similar objective of $c_v = (d_x^T w_v \gamma^+)/m$ is presented instead, where γ^+ is the non-negative of γ ($\gamma^+ = \max(0, \gamma)$). Physically, this new objective represents a momentum. While this term does not truly represent energy, the new mixed objective still yields a very similar behavior to the actual mix time-energy optimal objective in the sense of providing a trade-off between time and energy through selection of the gains. To demonstrate this point, Fig. 3 plots the energy-time curve for the optimal solutions for the same trip with the two different formulations, and for $w_v/w_t = 0.01 \dots 100$. To obtain these results, the non-convex problem was run through a multistart strategy to increase the probability of converging to the globally optimal solution. As is evident, the optimal energy-time curves are almost identical, meaning that the convex formulation effectively provides the same trade-off between energy and time as that of the original formulation. Interestingly, due to the globally optimal nature of the convex formulation, in some regions of Fig. 3, its solution is slightly more optimal than the non-convex formulation. It should be noted that given different physical meanings and scaling of the energy minimization objectives in the convex and non-convex case, it is expected that a different w_v/w_t ratio is required in the convex case for converging to the same optimal point as the non-convex case.

Fig. 3: Optimal Solutions with Convex and Non-Convex Objectives.



Eqn. 7b-e remain unchanged from Eqn. 6. ρ is substituted into the recursive time equation Eqn. 6e to obtain Eqn. 7f. $\rho = 1/v$ 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 of Eqn. 7g. With downwards pressure on ρ from the minimum time objective ($c_t = w_t t_{k-1}$) and no other interfering constraints on ρ , it can empirically be seen that for any given feasible problem where $w_k \gg 0$, $\rho^* = 1/v^*$ holds at optimum. It should also be noted that through empirical testing, downwards pressure from the

minimum energy objective (c_e), for cases where $w_k = 0$, can also reduce the $p^* - 1/v^*$ gap to a small margin (but not true equality) such 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 outside the range of feasible t_k) the controller will violate the activity of $p^* = 1/v^*$ resulting in a situation where $p^* > 1/v^*$, 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 would continue functioning sub-optimally but safely until feasible trip specifications are supplied. 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 remains beyond the scope of this paper and a subject for future research.

Given that $p = 1/v$ and $\gamma = \Gamma/v$, Eqn. 6h can trivially be converted to the equivalent set of inequalities defined by Eqn. 7h. The non-negative control input is required for the energy efficient objective function and so it must be isolated. Eqn. 7i is equivalent to Eqn. 6b due to the fact that v is strictly positive as a result of Eqn. 7b. The non-linear traction loss region model in Eqn. 6i is converted to convex form through the substitution of γ and p , resulting in Eqn. 7j. The vehicle dynamics in Eqn. 6j is converted to the affine representation in Eqn. 7k through the same substitution of γ and p .

The entire above formulation (Eqn. 7) is therefore in convex, second-order conic problem form. This can be verified by comparing the problem structure to the SOCP review paper in [32]. Therefore solvers capable of working with conic optimization problems can be used to find the solution.

D. Advanced Convex Controller Model

Although Eqn. 7 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.
- Safely driving a train that has another train in front of it along the same route, and ensuring safety and optimality throughout the entire trip.

To achieve the above, the convex optimization is extended to address the mentioned issues with additional objective terms and constraints.

E. Overspeed Compensation

Although controller performance was acceptable in closed 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 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, would force the optimal controller to react with an aggressive response. Furthermore, of more serious concern is that any violation of V_{max} would cause the onboard safety supervision system to deploy the emergency brake [6].

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. 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:

$$c_{os} = d_{os}^T w_{os} \max(v - (V_{max} - B_{os}), 0) \quad (8)$$

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:

$$c_{os} = w_{os} d_{os}^T b \quad (9)$$

with the additional convex inequalities:

$$b \geq 0 \quad (10a)$$

$$b_i \geq v_i - (V_{max}[i] - B_{os}) \quad i = 1, \dots, k-1 \quad (10b)$$

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

F. Reserve Braking

In normal train operation, the maximum braking is usually constrained to a fraction of the full emergency braking capabilities of vehicle [13]. Since Eqn. 7a does not penalize braking, although it does rigidly limit the maximum allowable braking, when a train approaches an edge where V_{max} decreases or is approaching a stop, the controller would hold off braking until the latest possible point; this is 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. Excessive braking can be minimized by adding additional soft constraints on braking and adding a weight c_{brk} to the cost as follows:

$$c_{brk} = d_{brk}^T w_{brk} \max(\gamma^+ / m - \gamma / m - B_{brk} \beta, 0) \quad (11)$$

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. Thus when the vehicle approaches a pooled brake region the controller can demand up to an additional 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 equivalently:

$$c_{brk} = d_h^T h \quad (12)$$

with the additional convex constraints on h of:

$$h \geq 0 \quad (13a)$$

$$h \geq \frac{\gamma^*}{m} - \frac{\gamma}{m} - B_{brk}\beta \quad (13b)$$

G. 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 in close proximity to one another [33]. 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.
- Leading trains may or may not be optimally controlled. Regardless, it is necessary for the following train to be able to predict the entire trajectory of its leading vehicle. Therefore, 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 such that following a leading train will not violate them.
- A required safety buffer distance (X_{sep}) is known.

Fig. 4: Leading and Following Train Model.

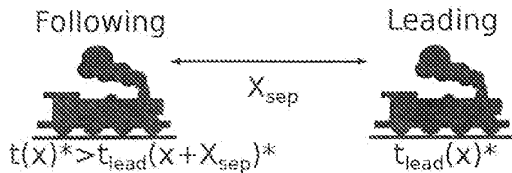


Fig. 4 demonstrates how a leading train can be used to constrain a following train in the position domain. Since the optimal time profile ($t_{lead}(x)^*$) is known, it can be shifted back by the required municipal, customer or safety separation distance (X_{sep}) and redescritized along the position 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.

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

$$t_i \geq T_{buff}[i] \quad i = 1, \dots, k-1 \quad (14)$$

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

H. Complete Advanced Closed Loop Optimization

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

$$\underset{\rho, \gamma, \gamma^*, b, h \in \mathbb{R}^{k-1}, v, t \in \mathbb{R}^k}{\text{minimize}} \quad f(z) = w_t t_b + d_z^T \left(w_e \frac{\gamma^*}{m} + w_{brk} h + w_{ov} b \right) \quad (15)$$

subject to: Eqn. 7b-k, 10, 13 and 14.

IV. IMPLEMENTATION AND RESULTS

ECOS (Embedded Conic Solver) is a small footprint Interior Point conic solver that uses a standard primal-dual Mehrotra predictor-corrector method with Nesterov-Todd scaling and self-dual embedding [30]. It is also fully open source (GNU GPLv3) and implements sparse matrix routines for accelerated computation.

The convex controller (Eqn. 15) was implemented in C with a direct interface to ECOS C API with a simple Lua wrapper to allow for easy access to the powerful torch7 computation environment for simulation and analytics. Since the convex controller formulation is highly sparse, all of the problem constraints were implemented as sparse triplet pair matrices using the CSPARSE sparse matrix data type. 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 optimality conditions hold and that the resultant solution is optimal [34], [30]. 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 [30], [32], [20]. Since the convex formulation in this paper has a strictly feasible constraints set (for feasible trip conditions) with strictly feasible equalities and relaxed

inequality variables, both the primal and dual are attainable and thus the optimal solution is globally optimal.

The entire formulation was then interfaced to a train model written in Lua and simulated as the ODE:

$$\frac{\partial x}{\partial t} = v \quad (16)$$

$$\frac{\partial v}{\partial t} = \frac{\min(\Gamma_1, R(v))}{m} + c_0 + c_{a1}v + c_{a2}v^2 + G(x) \quad (17)$$

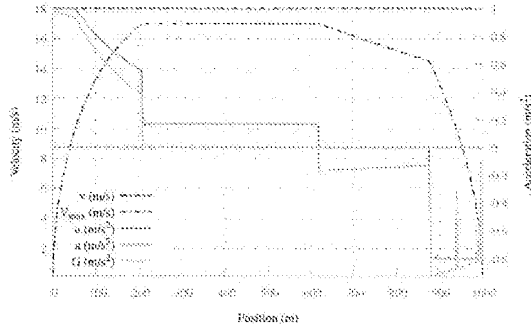
where the ODE was simulated with a high frequency Euler integration (1kHz) rate, assuring high modeling accuracy. Since the optimal controller discretizes the track in position to a substantially lower precision some significant modeling errors become inherently present.

All of the tests reported in the paper were conducted with $k = 500$ 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.

A. Simple Closed-Loop Case

Under a zero grade trip, with a slack T_{max} and a constant V_{max} profile, Howlett and Pudney theoretically proved that the energy optimal control strategy would be that of the form of MOTORING-SPEEDHOLD-COAST-BRAKE [16], [21]. The convex closed-loop controller also behaves as such given these same conditions as is apparent in Fig. 5. Thus, it can be claimed that similar performance as seen by [16], and [21] is observed without needing to structure a control problem to explicitly solve for optimal switching points [16], [21].

Fig. 5: Mixed Objective Trip on a Flat Grade



$$w_t = 1, w_e = 1, w_{as} = 0.02, w_{brk} = 1$$

$$T_f = 77.99s, E_f = 236.72J/kg, X_s = 1000.11m.$$

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.

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_{as} to the speed limit.

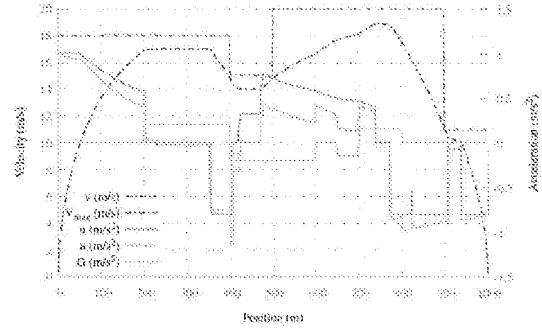
B. Advanced Closed-Loop Case

Since Fig. 5 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.

On an advanced track with non-constant V_{max} profile, the MOTORING-SPEEDHOLD-COAST-BRAKE is no longer optimal as demonstrated by the results in Fig. 6, Fig. 7, Fig. 8 and Fig. 9. 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. 6 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 ($B_{as} = 1m/s$) as much as possible and reserving 20% of braking for emergency situations.

Fig. 6: Time Efficient Control for an Advanced Trip.



$$w_t = 10, w_e = 1, w_{as} = 0.02, w_{brk} = 1$$

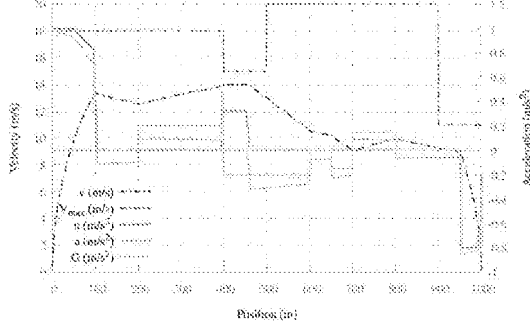
$$T_f = 79.31s, E_f = 331.52J/kg, X_s = 999.94m.$$

Fig. 7, 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. 6 (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 the time efficient operating mode.

Fig. 8 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 those noted in Fig. 6 and Fig. 7.

The trains in all three of the above test cases achieved very good stopping accuracy and adhered to all the trip constraints.

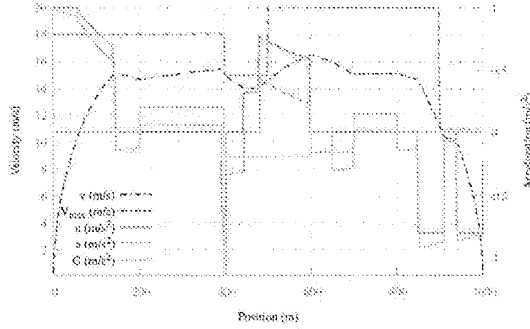
Fig. 7: Energy Efficient Control for an Advanced Trip.



$$w_t = 1, w_e = 10, w_{os} = 0.02, w_{brk} = 1$$

$$T_f = 100.87s, E_f = 115.34J/kg, X_s = 999.97m.$$

Fig. 8: Mixed Objective Control for an Advanced Trip.



$$w_t = 1, w_e = 1, w_{os} = 0.02, w_{brk} = 1$$

$$T_f = 84.92s, E_f = 222.12J/kg, X_s = 1000.02m.$$

C. Several Trains on a Single Guideway

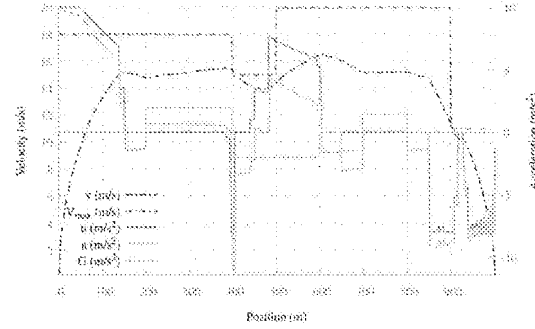
To demonstrate the effectiveness of the controller in driving safely behind a leading train, it is first necessary to launch a leading train running the optimal control system and then spawn a following train behind it. In order for the leading train to significantly impede the following, thus showing the full scale of the controller's ability to 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. 8. 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 configured with the same objective as those in Fig. 6.

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. 10 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 it can set off as it must wait until at least $T_{lead}[1]$ seconds. Since only the arrival time t_k is being optimized for due to the objective in Eqn. 15 and not times at any other $X[i]$ for $i = 1 \dots k-1$ there is no incentive for the following train to ride close to the buffer until at least 900m. Thus, since it had a mixed objective cost the controller minimizes the energy cost (w_e) since it cannot bring the arrival time down any more than $T_{lead}[k]$. This behavior can also be seen in Fig. 9 as, despite having a larger time weight (w_t) than the leading, the following's resultant control profile looks like the leading's and does not expend energy unnecessarily. Both leading and following vehicles drive optimally given their trip designs, and the following train successfully uses knowledge of the leading vehicle's expected behavior to optimally plan and implement a safe control strategy.

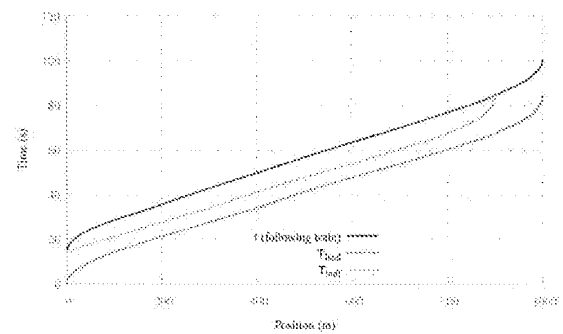
Fig. 9: Time Efficient Following Train.



$$w_t = 10, w_e = 1, w_{os} = 0.02, w_{brk} = 1$$

$$T_f = 100.87s, E_f = 199.95J/kg, X_s = 999.97m.$$

Fig. 10: Both Leading and Following Trains on the Same Track.



D. Real-time Computation Performance

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 evident from the results in Fig. 11 and Fig. 12.



Fig. 11: Cpu Performance - Intel i7 6700.

1) *Main-stream Desktop Processor:* On a mid-range Intel i7-6700 (Fig. 11), the worst case closed-loop optimization time is less than 0.2s for control horizons of $k = 1000$. This large k is substantially larger than what is necessary to maintain robust control of a passenger locomotive. The optimization time decreases as the train approaches the target station due to the receding control horizon and the shrinking value of k . The noise in the plots can be attributed to the lack of any hard or soft real-time enforcement on the testing hardware.

2) *Embedded Application:* It can be shown that a full desktop i686 or amd64 architecture machine is not even required for robust control. Therefore, 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. 12, the controller is fully capable of real-time execution on a low power system (sub 15w [35]) for realistic control horizon dimensions (k).

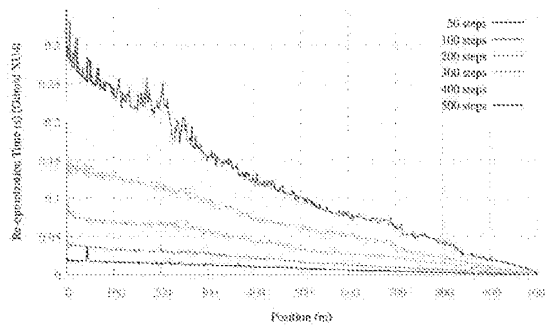


Fig. 12: Cpu Performance - Hardkernel Odroid XU4.

V. CONCLUSION

With the increasing computational capabilities of modern computer hardware, optimal train 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 research 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. In conclusion, this research represents the viability of advanced optimal control for real-time control applications through convex numerical optimization strategies.

Future work is to focus on more optimal methods of discretization of the control horizon such that equivalent controller performance can be gained with a smaller k , thus further reducing computation time for very long trips. Inclusion of propulsion and braking delays in the control formulation is another potential direction for future research. Additionally of interest are real-time capable optimal parameter estimation strategies such that the presented optimal control algorithm can be deployed on any rail vehicle with little to no required modeling knowledge of the vehicle in question. In this fashion a parameter estimator can be run in closed loop with the controller, where the optimal control solution becomes truly optimal for the vehicle once all parameters are quickly and optimally determined.

Appendix 2

Convex Real-Time Control of a Commuter Train

Abstract—Patent details for a convex real-time capable optimal controller for a commuter train. This document outlines the mathematical formulation and required computation algorithms for robust optimal train control.

Index Terms—Thales Canada, Optimal Control, Automatic Speed Control, Real-time, Second-Order-Cone-Programming

I. NOTATION AND ABBREVIATIONS

SOCP	Second Order Cone Program
ATO	Automatic Train Operation
ATS	Automatic Train Supervision
ATC	Automatic Train Control
DAS	Driver Assistant System
ECOS	Embedded Cone Solver
SLQP	Sequential Least Squares Programming
IPDP	Integer Path Optimizer
MOG	Modified Gradient Method
KF	Kalman Filter
API	Application Program Interface
IP	Interior Point
PID	Proportional Integral Derivative controller
u^*	The optimal value of the associated variable and optimal objective for u

TABLE I: 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 (9m/s for station 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
$N_{sep} \in \mathbb{R}$	Minimum separation distance between trains (m)
$\theta(x)$	Track inclination (rad)
f_{cl}	Closed loop control frequency (Hz)
$\epsilon \in \mathbb{R} > 0$	Very small positive number
$P^* \in \mathbb{R}$	Optimal primal solution ($P^* = f(x^*)$)
$D^* \in \mathbb{R}$	Optimal dual solution
$\mathbb{K}$	Proper cone
$\mathbb{K}_{++}$	Non-negative orthant cone
$\mathbb{K}_{so}$	Second order cone
$\mathbb{K}_{exp}$	Exponential cone

II. 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.

TABLE II: Train Parameters and Optimization Variables

$A(v, \Gamma)$	Traction and braking function (m/s ²)
$C(v)$	Motion resistance function (m/s ²)
$F_{max} \in \mathbb{R} \leq 0$	Maximum braking force
$F_{max,v} \in \mathbb{R} \geq 0$	Maximum propulsion force
$c_{01}, c_{02}, c_{03} \in \mathbb{R}$	Davis equation coefficients
$\gamma_0, \gamma_1, \gamma_2 \in \mathbb{R}$	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)
$\gamma, \gamma + \epsilon \in \mathbb{R}^{K-1}$	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
$f(x) \in \mathbb{R}$	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
$s \in \mathbb{R}^{7K-6}$	Optimization variable vector
$w \in \mathbb{R}^{7K-6}$	Standard form objective weight vector
$U \in \mathbb{R}^{2(K-1)+2, 7K-6}$	Standard form equality constraint matrix
$y \in \mathbb{R}^{2K+2}$	Standard form equality vector
$Q \in \mathbb{R}^{12(K-1)+2(K-1), 7K-6}$	Standard form inequality matrix
$Q_{++} \in \mathbb{R}^{12(K-1)+2, 7K-6}$	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 III: Uncertainty and Parameter Estimation

$\gamma \in \mathbb{R}$	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
$E \in \mathbb{R}^n$	Prior and posterior 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)
$\hat{t} \in \mathbb{R}^n$	Measured traction acceleration (KF)
$\hat{v} \in \mathbb{R}^n$	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)
$\hat{x}$	Measured vehicle position (m)
$\hat{v}$	Measured vehicle velocity (from speedometer) (m/s)
$\hat{a}$	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_u^2	Acceleration modelling variance (m/s ²)

A. 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/s²]) states at discrete points in time, t_i [1], [2]. 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 [1], [2], [3], [4]. The model can be described by the ordinary differential Newtonian equations of motion numerically integrated via the Euler method as below:

$$x_{i+1} = x_i + v_i \Delta t_i \quad (1a)$$

$$v_{i+1} = v_i + a_i \Delta t_i \quad (1b)$$

$$a_i = A(v_i, \Gamma_i) + C(v_i) + G(x_i) \quad (1c)$$

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

$$G(x_i) = g \sin(\theta(x_i)) \quad (1d)$$

For convenience, this document 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.

B. Davis Equation

The motion resistance model $C(v_i)$ can be expressed as the Davis equation with constant coefficients c_0, c_1, c_2 [3]. 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_1, c_2 [3], either through the optimal parameter estimation of vehicle test data or semi-empirically through equations such as the Armstrong and Swift resistance model [4]. The model is widely represented in modern train literature as:

$$C(v_i) = c_0 + c_1 v_i + c_2 v_i^2 \quad (2)$$

C. 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 [5], [6].

It is common for rail electric motors to exhibit tractive effort loss behavior and have maximum propulsion limits of the form of Fig. 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 $R(v)$ [6]. 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:

$$A(v_i, \Gamma_i) = \frac{\min(\Gamma_i, R(v_i))}{m} \quad \text{for} \quad F_{min} \leq \Gamma \leq F_{max} \quad (3)$$

On a flat grade profile, the hypothetical maximum speed limit occurs along the intersection of $A(v_i, \Gamma_i)$ and the resistance curve $C(v_i)$ [6].

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:

$$R(v_i) = r_0 + r_1 v_i + r_2 v_i^2 \quad (4)$$

where r_0, r_1, r_2 are constant polynomial coefficients provided by a vehicle manufacturer or through parameter estimation (least-squares) of provided motoring force curves.

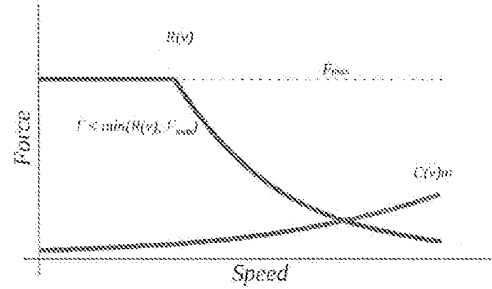


Fig. 1: Traction Loss and Vehicle Resistance

D. Position Domain

In the time domain, the train control optimization horizon is the discretized time vector t for $t = 0 \dots t_F$ with a fixed sampling interval of Δt [5]. 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 [7] that the advanced distributed vehicle model for a multicar train can be in total represented as a point-mass model in the position domain [7]. 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 [8]. 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) α (1/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 [8], [5], [1]. The dynamics are transformed to the following:

$$t_{i+1} = t_i + \frac{\Delta_x}{v_i} \quad (5a)$$

$$v_{i+1} = v_i + \alpha_i \Delta_x \quad (5b)$$

$$\alpha_i = \frac{\partial v_i}{\partial x} = \frac{a_i}{v_i} = \frac{A(v_i, \Gamma_i) + C(v_i) + G[i]}{v_i} \quad (5c)$$

These equations are equivalent to the time domain representation of the vehicle motion (1).

E. 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:

$$v \geq \epsilon \quad (6)$$

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

F. 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 modelling accuracy, thus resulting in a more computationally efficient optimal control solution.

In practice, a trapezoidal Δ_x profile — as demonstrated in Fig. 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.

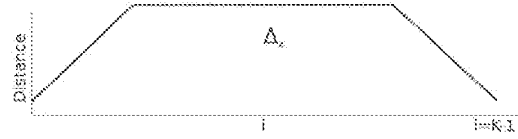


Fig. 2: Trapezoidal Δ_x Profile

III. NON-CONVEX POSITION DOMAIN OPTIMIZATION FORMULATION

A. 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 document 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:

$$\begin{aligned} & \underset{z \in \mathbb{R}^N}{\text{minimize}} \quad f(z) = w_1 t_K + w_2 \frac{\Delta_x(\Gamma^+)}{m} \quad (7a) \\ & \text{subject to:} \quad \Gamma, \Gamma^+ \in \mathbb{R}^{N-1} \end{aligned}$$

$$0 \leq \Gamma^+ \quad \Gamma \leq \Gamma^+ \quad (7b)$$

$$\epsilon \leq v_i \leq V_{max}[i] \quad i = 2, \dots, K-1 \quad (7c)$$

$$v_1 = V_0 \quad v_K = V_f = 0 \quad (7d)$$

$$t_{i+1} = t_i + \frac{\Delta_x[i]}{v_i} \quad i = 1, \dots, K-1 \quad (7e)$$

$$t_1 = T_{start} \quad (7f)$$

$$T_{min} \leq t_K \leq T_{max} \quad (7g)$$

$$F_{min} \leq \Gamma \leq F_{max} \quad (7h)$$

$$\Gamma_i \leq r_0 + r_v v_i + r_{v2} v_i^2 \quad i = 1, \dots, K-1 \quad (7i)$$

$$v_{i+1} = v_i + (c_0 + G[i] + \frac{\Gamma_i}{m} + c_v v_i + c_{v2} v_i^2) \frac{\Delta_x[i]}{v_i} \quad \text{for } i = 1, \dots, K-1 \quad (7j)$$

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 $(\Delta_e^T \Gamma^+)/m$ [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 $(\Delta_e^T \max(\Gamma, 0))/m$ 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:

$$\underset{z \in \mathbb{R}^K}{\text{minimize}} \quad f(z) = w_t t_K + w_e \frac{\Delta_e^T \Gamma^+}{m}$$

where Γ^+ is defined as the following epigraph transformation and is equivalent to the motoring input applied on the vehicle:

$$0 \leq \Gamma^+ \quad \Gamma \leq \Gamma^+$$

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:

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

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:

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

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:

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

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:

$$t_1 = T_{start}$$

Thus, the arrival time at $X[K]$ can be constrained from above and below by predefined minimum and maximum allowable trip traversal time limits:

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

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

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

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

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

Finally, the dynamic model for the vehicle velocity is constrained, resulting in the recursive equality equation:

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

B. The Closed Loop Problem

The full control problem can be defined as a closed-loop version of (7). 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 (7) for the optimal Γ^* that satisfies all of the constraints and results in the optimal cost $f(x^*)$. 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:

$$X[j] \leq X_0 \quad X[j+1] \geq X_0 \quad (8a)$$

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 these vectors.

$$V_{max} = [V_{max}[j], V_{max}[j+1], \dots, V_{max}[K]] \quad (8b)$$

$$G = [G[j], G[j+1], \dots, G[K-1]] \quad (8c)$$

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 .

$$X = [X_0, X[j+1], \dots, X[K]] \quad (8d)$$

$$\Delta_x = [X[2] - X[1], \Delta_x[j+1], \dots, \Delta_x[K-1]] \quad (8e)$$

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. 3. 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, (7) is resolved at resolved each $\frac{1}{f_{ctrl}}$ 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.

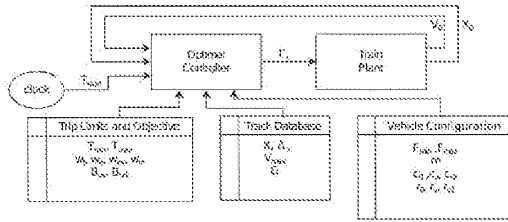


Fig. 3: Controller Architecture

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 [9]. 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 [9]. 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 than 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.

IV. CONVEX OPTIMIZATION FORMULATION

A. Convex SOCP

(7) 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 [10]. 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 [11]. 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 [10].

Convex problems largely benefit from their highly scalable nature [12], [13], 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.

(7) 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:

$$\underset{\rho, \gamma, \gamma^+ \in \mathbb{R}^{K-1}}{\text{minimize}} \quad f(z) = c_t + c_e = w_t t_K + \frac{\Delta_x^T(w_c \gamma^+)}{m} \quad (9a)$$

subject to:

$$\epsilon \leq v_i \leq V_{max}[i] \quad i = 2, \dots, K-1 \quad (9b)$$

$$v_1 = V_0 \quad v_K = V_f = 0 \quad (9c)$$

$$t_1 = T_{start} \quad (9d)$$

$$T_{min} \leq t_K \leq T_{max} \quad (9e)$$

$$t_{i+1} = t_i + \rho_i \Delta_x[i] \quad i = 1, \dots, K-1 \quad (9f)$$

$$\left\| \frac{2}{v_i - \rho_i} \right\|_2 \leq v_i + \rho_i \quad i = 1, \dots, K-1 \quad (9g)$$

$$E_{min} \rho \leq \gamma \leq E_{max} \rho \quad (9h)$$

$$0 \leq \gamma^+ \quad \gamma \leq \gamma^+ \quad (9i)$$

$$\gamma_i \leq r_0 \rho_i + r_0 + r_{c2} v_i \quad i = 1, \dots, K-1 \quad (9j)$$

$$(9k)$$

$$v_{i+1} = v_i + ((c_0 + G[i])\rho_i + \frac{\gamma_i}{m} + c_e + c_{v2} v_i) \Delta_x[i] \quad \text{for } i = 1, \dots, K-1 \quad (9l)$$

The objective function remains unchanged from (7) for the minimum time cost. Since Γ is no longer an optimization variable in the convex reformulation, an energy efficient similar objective of $c_e = (\Delta_x^T w_c \gamma^+)/m$ is presented instead, where γ^+ is the non-negative of γ ($\gamma^+ = \max(0, \gamma)$). Physically, this new objective 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:

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

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. 4 compares the optimal solutions for a mixed objective given trip with $w_e/w_t = 0.01 \dots 100$ 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. 4 the convex problem is more optimal than the non-convex objective.

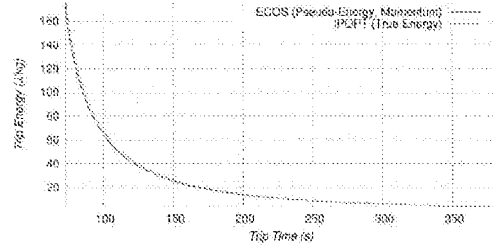


Fig. 4: Optimal Solutions with Convex and Non-Convex Objectives

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. 5, and 6.

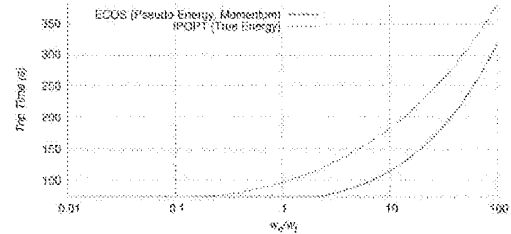


Fig. 5: Optimal Solutions with Convex and Non-Convex Objectives: Arrival Time

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.

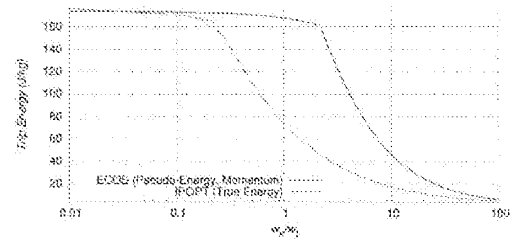


Fig. 6: Optimal Solutions with Convex and Non-Convex Objectives: Trip Energy

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 (7):

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

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

$$t_1 = T_{start}$$

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

ρ is substituted into the recursive time equation (7)e to result in:

$$t_{i+1} = t_i + \rho_i \Delta x[i] \quad i = 1, \dots, K-1$$

$\rho = 1/v$ 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:

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

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 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 (9)h, 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 $\gamma = \Gamma/v$, (7)h can trivially be converted to the equivalent set of inequalities:

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

Since the non-negative control input is required for the energy efficient objective function it must be isolated. The following is equivalent to (7)b due to the fact that n is strictly positive as a result of (9)b:

$$0 \leq \gamma^+ \quad \gamma \leq \gamma^+$$

The non-linear traction loss region model (7)i is converted to convex form through the substitution of γ and ρ :

$$\gamma_i \leq r_0 \rho_i + r_n + f_{a2} v_i \quad i = 1, \dots, K-1$$

The vehicle dynamics (7)j is converted to the affine representation below through the same substitution of γ and ρ :

$$v_{i+1} = v_i + ((c_0 + G[i])\rho_i + \frac{\gamma_i}{m} + c_b + c_{a2}v_i)\Delta x[i] \quad i = 1, \dots, K-1$$

The entire above formulation (9) is therefore in convex second-order conic problem form. This can be verified by comparing the problem structure to the SOCP review paper by [14]. Therefore this problem can be numerically optimized by solvers capable of working with conic optimization problems.

B. Advanced Convex Controller Model

Although (9) 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.

C. 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 SPEED-HOLD 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 aggressive 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_{ss}), 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:

$$c_{os} = \Delta_x^T w_{os} \max(v - (V_{max} - B_{os}), 0) \quad (11)$$

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:

$$c_{os} = w_{os} \Delta_x^T b \quad (12)$$

with the additional convex inequalities:

$$b \geq 0 \quad (13a)$$

$$b_i \geq v_i - (V_{max}[i] - B_{os}) \quad i = 1, \dots, K - 1 \quad (13b)$$

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.

D. Reserve Braking

In normal train operation, the maximum braking is usually constrained to a fraction of the full emergency braking capabilities of vehicle [15]. Since (9)a 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:

$$c_{brk} = \Delta_x^T w_{brk} \max(\gamma^+ / m - \gamma / m - B_{brk} \rho, 0) \quad (14)$$

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:

$$c_{brk} = \Delta_x^T h \quad (15)$$

with the additional convex constraints on h of:

$$h \geq 0 \quad (16a)$$

$$h \geq \frac{\gamma^+}{m} - \frac{\gamma}{m} - B_{brk} \rho \quad (16b)$$

E. 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 [16]. 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. 7 and Fig. 8.
- 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 ($X_{sep}(x)$).

Fig. 7 and Fig. 8 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 resampled along the position vector X of the following train such that: $t \geq T_{min}$ for $T_{lead} = t_{lead}(x + x_{sep})$. 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

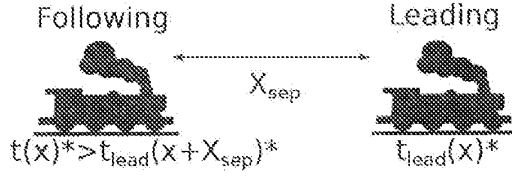


Fig. 7: Leading and Following Train Model

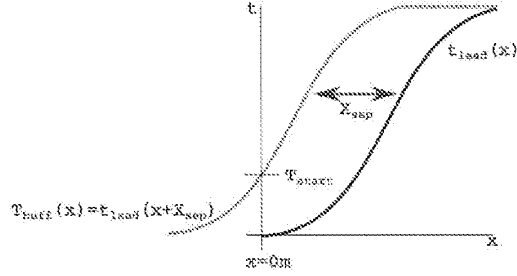


Fig. 8: Following Train Time Constraint

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:

$$t_i \geq T_{\text{min}}[i] \quad i = 1, \dots, K-1 \quad (17)$$

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

F. 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:

$$\begin{aligned} \text{minimize} \quad & f(x) = w_t t_K \\ & + \Delta_x^T (w_e \frac{\gamma^+}{m} + w_{br} h + w_{os} b) \end{aligned} \quad (18)$$

subject to: Equations (9)b-k, (13), (16) and (17)

(18) is henceforth referred to as the convex optimization formulation and the convex controller interchangeably and will be thoroughly tested and implemented in the subsequent sections of this document.

V. IMPLEMENTATION OF OPTIMAL CONTROL

A. SOCP In Standard form

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, starting with the fact that it is open-source (GNU GPL v3 licensed) whereas MOSEK is a closed source commercial product [13]. 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 [12]. 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 [12], [13]. 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 [12].

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 ((19)) 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) [12], [14]. 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:

$$\begin{aligned} \text{minimize} \quad & w^T z \\ \text{subject to} \quad & Uz = y \\ & Qz + s = r, \quad s \in \mathbb{K} \end{aligned} \quad (19)$$

Where:

$$\begin{aligned} z = [& v_1, t_1, p_1, \gamma_1^+, b_1, h_1, \dots, \\ & v_{K-1}, t_{K-1}, p_{K-1}, \gamma_{K-1}^+, b_{K-1}, h_{K-1}, \dots, v_K, t_K] \end{aligned} \quad (20)$$

$$\begin{aligned} w = [& 0, 0, 0, 0, \Delta_x[1]w_e, \Delta_x[1]w_{br}, \Delta_x[1]w_{os}, \dots, \\ & 0, 0, 0, 0, \Delta_x[K-1]w_e, \Delta_x[K-1]w_{br}, \Delta_x[K-1]w_{os}, 0, w_t] \end{aligned} \quad (21)$$

$$Q = \begin{bmatrix} Q_{++} \\ Q_{so} \end{bmatrix} \quad r = \begin{bmatrix} r_{++} \\ r_{so} \end{bmatrix} \quad (22)$$

Thus U and y represent the equalities (9)c, d, f and k; and Q and r represent the inequality (9)b,e,g,h,i and j, (13)a-b, (16)a-b and (17). z denotes the primal variables and s denotes the conic slack variables [12]. Since all of the mentioned equality and inequality constraints are mostly vectorized along X at each i for $i = 0 \dots K-1$ 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.

B. Numerically Solving with ECOS

ECOS is configured to terminate an optimization once a sufficiently small duality gap is achieved ($|P^* - D^*| \leq 1 \times 10^{-6}$), 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 optimality conditions hold and that the resultant solution is optimal [17], [12]. The dual problem to (19) is the following:

$$\begin{aligned} & \underset{\lambda, \mu}{\text{maximize}} && -y^T \lambda - r^T \mu \\ & \text{subject to} && Q^T \mu + U^T \lambda + w = 0 \\ & && \mu \in \mathbb{K} \end{aligned} \quad (23)$$

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 [12], [14], [18]. Since the convex formulation in this document 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.

C. 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. 3. As was noted previously, U and Q are very sparse and thus take advantage of ECOS's sparse computation through the efficient CSPARSE routines [12]. 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 (8).

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 [19]. 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 [20], [13].

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:

$$\Gamma_1 = \gamma_1^* v_1^* = z_1^* z_4^* \quad (24)$$

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. ?? 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 [21]. 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 (-O3), naturally eliminating any language implementation slowdowns that were 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 initialization 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.

VI. 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 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 [22].

A Kalman filter is introduced to estimate the mass every $1/f_{ctrl}$ seconds (every control cycle). The entire closed-loop

Algorithm 1 Optimal Controller

```

{Problem Initialization}
 $X[i] = \sum_{j=1}^i \Delta x[j] \quad \text{for } i = 1 \dots K$ 
 $V_{max} \leftarrow [V_{max}(X[1]), V_{max}(X[2]), \dots, V_{max}(X[K])]$ 
 $G \leftarrow [G(X[1]), G(X[2]), \dots, G(X[K-1])]$ 
 $\Delta_i = 0.01$ 
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}
   $*X[0] = *X[i]; \quad X[0] = X_0$ 
   $*\Delta_x[0] = *\Delta_x[i]; \quad \Delta_x[0] = X[1] - X[0]$ 
   $*V_{max}[0] = *V_{max}[i]$ 
   $*G[0] = *G[i]$ 
   $T_{start} = T$ 
  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  $\Gamma_1$  to vehicle
   $T+ = \Delta_i$ 
end while

```

simulation interface is altered to that in Fig. 9, 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 estimate 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.

A. Mass Estimation Model

A simple linear model is proposed for estimating mass based on (1)c — the dynamic model of the train. It should be noted that the index j for $j = 0 \dots n$ corresponds to the optimization cycle count in the time domain, increasing at each successive re-optimization. Since (1)c is actually linear in $1/m$ and not m , a new variable e is introduced as:

$$e = \frac{1}{m} \quad (25)$$

This allows to treat the estimation problem linearly, and thus a linear KF can be used to estimate the inverse mass which can

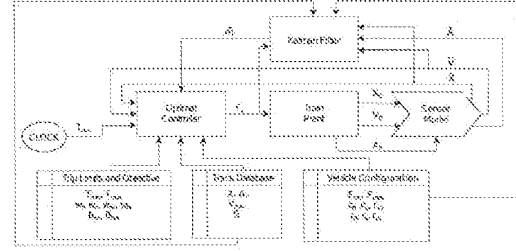


Fig. 9: Simulation Model with Sensor Noise and KF Mass Estimator

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:

$$e_j = M_A e_{j-1} + M_B d + w_R \quad (26a)$$

$$l_j = M_{Cj} e_j + w_H \quad (26b)$$

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:

$$M_A = 1 \quad M_B = 0 \quad (26c)$$

where M_{Cj} is:

$$M_{Cj} = \min(\Gamma_1^*, \tau_0 + r_v \bar{V} + r_{v2} \bar{V}^2) \quad (26d)$$

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:

$$l_j = \bar{A} - G(\bar{X}) - c_0 - c_v \bar{V} - c_{v2} \bar{V}^2 \quad (26e)$$

$\bar{X}, \bar{V}$ 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.

B. Scalar Kalman Filter Model

With the above mass estimation model, a scalar KF can be formulated with the estimation model equations [22]:

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

$$\hat{e}_{j|j-1} = M_A \hat{e}_{j-1|j-1} + M_B d_j \quad (27a)$$

$$E_{j|j-1} = M_A^T E_{j-1|j-1} + H \quad (27b)$$

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

$$\tilde{y}_j = l_j - \hat{M}_{Cj} \hat{e}_{j|j-1} \quad (27c)$$

and its residual Covariance:

$$S_j = \hat{M}_{Cj}^2 E_{j|j-1} + R \quad (27d)$$

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

$$\chi_j = \frac{E_{j|j-1} \hat{M}_{Cj}}{S_j} \quad (27e)$$

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:

$$\hat{e}_{j|j} = \hat{e}_{j|j-1} + \chi_j \tilde{y}_j \quad (27f)$$

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

$$\hat{m}_j = \frac{1}{\hat{e}_{j|j}} \quad (27g)$$

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

$$E_{j|j} = (1 - \chi_j \hat{M}_{Cj}) E_{j|j-1} \quad (27h)$$

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.

C. Kalman Filter Implementation

Substituting the inverse mass estimation model (26) into the above scalar kalman filter formulation (27) results in the following simplified equations: The priori covariance is evaluated as:

$$E_{j|j-1} = E_{j-1|j-1} + H \quad (28a)$$

The scalar Kalman gain (χ) evaluates to:

$$\chi_j = \frac{\min(\Gamma_1^*, r_0 + r_v \bar{V} + r_{v2} \bar{V}^2) E_{j|j-1}}{\min(\Gamma_1^*, r_0 + r_v \bar{V} + r_{v2} \bar{V}^2) E_{j|j-1} + R} \quad (28b)$$

This results in the posteriori estimate and covariance:

$$\begin{aligned} \hat{e}_{j|j} = \hat{e}_{j|j-1} + \chi_j ((\bar{A} - G(\bar{X}) - c_0 - c_v \bar{V} - c_{v2} \bar{V}^2) \\ - \min(\Gamma_1^*, r_0 + r_v \bar{V} + r_{v2} \bar{V}^2) \hat{e}_{j|j-1}) \end{aligned} \quad (28c)$$

$$E_{j|j} = E_{j|j-1} (1 - \min(\Gamma_1^*, r_0 + r_v \bar{V} + r_{v2} \bar{V}^2) \chi_j) \quad (28d)$$

And finally, inverting the estimate to resolve the mass:

$$\hat{m}_j = \frac{1}{\hat{e}_{j|j}} \quad (28e)$$

$E_{0|0}$, 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.

Appendix 3

Optimal Control of a Commuter Train with Traction and Braking Time Delays

Abstract—The traction and braking subsystems of commuter trains often exhibit significant time delays. Optimal control of a commuter train is presented under two cases: (i) equal, and (ii) unequal time delays in the train traction and braking commands. The solution framework involves formulating and solving numerical optimization problems to achieve minimum mixed energy-time optimal control in discretized spatial and time domains. The optimization problems are re-solved repeatedly along the track for the remainder of the trip, using latest sensor measurements. This would essentially establish a feedback mechanism in the control to improve robustness to modelling errors. The case of equal traction-braking delays is treated in discretized spatial domain, for which a convex approximation to the optimization problem is presented. The non-equal traction-braking scenario is formulated in the time domain, leading to a nonconvex optimization problem. A key feature of the proposed methods is that they explicitly incorporate model information, including the time delays, in control synthesis hence avoiding performance degradation and potential instability. Results of numerical simulations demonstrate the validity and effectiveness of the proposed controllers.

Index Terms—Control systems, Automatic control, Delay systems, Feedback, motion planning, optimal control, rail transportation

1. INTRODUCTION

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

Train control systems are concerned with determining the traction and braking commands that will drive the train from station A, the origin, to station B, the destination. Optimal train control systems determine the traction and braking commands that will drive the train from station A to station B while minimizing some cost function. Common examples of

such controllers are minimum-time, minimum-energy without regenerative braking, and minimum-energy with regenerative braking. This paper considers a mixture of time and energy as the cost function to minimize. While regenerative braking is not specifically addressed here, it is rather trivial to extend the proposed minimum input-energy with no regenerative braking formulation to the minimum input-energy with regenerative braking case.

There is extensive literature on the optimal control of a train, e.g. see [3]–[6], and in the book by Howlett and Pudney [7]. A majority of the existing literature is mainly concerned with analytical solutions to the optimal control problem using classical optimal control theory. Such approach involve formulating a system Hamiltonian and applying Pontryagin's Minimum principle. Numerical algorithms are sometimes given to solve the resulting equations. While they provide interesting theoretical insight, these approaches are limited in their ability to incorporate advanced system dynamics models, state constraints, input constraints, mixed state-input constraints, and complex objectives. This is due to the difficulty in obtaining analytical solutions when incorporating more complicated objectives and constraints.

Our recent work in [8] presents a novel convex approximation to the problem of optimal train control to which can be quickly and robustly solved using standard convex optimization solvers. The convex approximation is formulated in the discrete position domain. The main motivation for the present work is to include traction and braking delays in the problem formulation, which were not addressed in [8]. In this paper, the optimal control problem with input delays is transformed it into an NLP (nonlinear programming) problem which is solved numerically using optimization solvers.

The continuous-time optimal control problem can be viewed as an infinite-dimensional optimization problem, i.e. it has an infinite number of decision variables [9]. The direct transcription (simultaneous approach) method is chosen from the family of direct solution methods, described in [10], [11]. It involves discretization of both the states and controls, which then become the finite-dimensional set of decision variables in the NLP, and discretization of the differential equation constraints, which are converted into difference equation constraints. The approach described above is a model-based approach, and falls into the family of model predictive control strategies. This is because the model of the system, the system dynamics characterized by differential equations, are directly used to determine the control. The model is used to predict the response of the system to changes in the control over a window, and this is used to determine the optimal control

trajectory to be applied over the same window. The model predictive control framework has the advantage of explicitly considering constraints on the states, controls, and functions of the states and controls [12].

The goal of this paper is to present a formulation for optimal control of a train, considering delays in braking and propulsion systems. The case of equal delays in traction and braking is treated separately from the general case, leading to a convex constrained optimization problem. Actuator delays are common in many control systems, and if ignored, could degrade their performance and even cause instability [13]. The control of systems with delays pose considerable challenge and difficulty, and even more so in the case of optimal control. The existing literature on optimal control of trains does not consider delays in the actuation model. The delays in the traction and braking control inputs are due to the equipment and hardware used to implement the systems. The equipment and hardware suffers from dead-time, reconfiguration time, and warm-up time. Optimal controllers, designed without considering these delays, may not realize their full performance potential in actual implementation, specially in case of large delays.

A number of approaches have been proposed for the control of linear time-delay systems, including finite spectrum assignment [14], [15], reduction [16], modified Smith predictor [17], [18], and feedback stabilization [19]. Finite spectrum assignment is concerned with arbitrarily assigning a finite set of eigenvalues out of an infinite number of eigenvalues [14], [15]. The reduction method transforms an infinite-dimensional system to a finite-dimensional one, the system with delay is transformed to one without delay [16]. Feedback stabilization and the modified Smith predictor are both predictive approaches to dealing with delay [13], [19], [20]. All of these approaches transform the problem of control of a input-delay system to one without any input-delay. The predictor approach has been presented for nonlinear systems with the equal delays in input channels [13].

This paper presents a new optimal control formulation for moving a commuter train from Station A to Station B using a mixed time-energy objective, while considering time delays in traction and motoring commands, i.e. see Fig. 1. The control problem is solved repeatedly along the track using the latest sensor measurements, making it suitable for online control. The control approach is in contrast with other existing controllers which operate as merely driver-assist systems, e.g. see [3], [21]. The proposed model-based controller uses realistic models of the train and track and can impose user-specified constraints such as speed limits, maximum arrival time, and constraints on traction/braking commands.

The train optimal control problem is treated in discrete domains, allowing it to be formulated and solved as conventional numerical optimization problems. Two distinct cases are considered and treated separately, namely optimal control with equal time delays in traction and braking, and one with unequal delays. In the case of equal delays, a convex approximation of the constrained optimization problem is presented using discretization in the position-domain. Convex optimizers can effectively and quickly find a global optimal

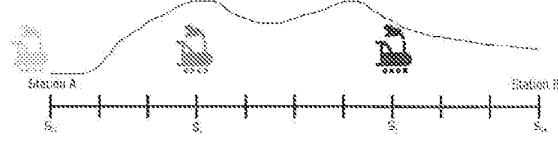


Fig. 1. The goal is to move the train from Station A to Station B using a mixed objective of time and energy. The position, velocity, acceleration, traction effort, and braking effort profiles for the remainder of the trip are determined by formulating and solving optimization problems repeatedly at discrete points along the track, e.g. s_0 (starting point), s_i , s_j , etc.

solution to such a problem. The case of unequal delays is treated in the discrete timedomain resulting in a nonlinear nonconvex optimization. It is noted that using the system model within a numerical optimization framework allows for the integration of a model-based predictor into the controller to effectively deal with the actuation delays.

The rest of the paper is organized as follows. In Section II, the optimal control formulation for the case of equal delays in the traction/braking commands is presented. The case of non-equal delays is studied in Section III. The controller implementation and the results of numerical simulations are discussed in Section IV. Finally, the paper is concluded in Section V.

II. OPTIMAL CONTROL FORMULATION WITH EQUAL DELAYS IN TRACTION/BRAKING COMMANDS

The case with equal delays in traction and braking commands is treated first. The system dynamics and optimization problem are formulated using position as the independent variable. The equations of motion using position as the independent variable are detailed in [3], [5], [6], [22] for a train, and for more general vehicles in [23]. In [24], it is shown that the equations of motion using a distributed-mass train model can be reduced to the equations of motion of a point-mass train model.

The train dynamics in the continuous position-domain are given by

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

$$mv(s) \frac{dv(s)}{ds} = -C(v(s)) + g(s) + u(s - S_D) \quad (2)$$

with the variables defined in Table I. The traction force is limited by

$$u_{\text{trk,max}} = \min \left\{ U_{\text{trk,max}}, \frac{P}{v}, \frac{k_P P}{v^2} \right\} \quad (3)$$

where P is the maximum power (constant), v is train velocity, $U_{\text{trk,max}}$ is an upper bound on tractive force (constant), and $k \geq 1$ is a constant. Note that the term $\frac{k_P P}{v^2}$ is an approximation used for the motor field weakening region [25], [26]. Moreover, S_D is the train position T_D seconds ago, i.e. $t(S_D) = t(s) - T_D$.

The optimal control formulation, used in Algorithm 1, and based on the convex approximation in our recent work [8] is given below. A summary of the convex approximation is provided in Appendix A for ease of reference.

TABLE I
POSITION-DOMAIN NOTATION AND VARIABLES

Symbol	Quantity
s	Position
s_0	Initial position
s_f	Final position: position at destination
$t(k)$	Time
$v(s) = \frac{ds}{dt} = \left(\frac{dt(s)}{ds}\right)^{-1}$	Velocity
$u(s)$	Control acceleration: the control force normalized by mass
$g(s)$	Grade force: force of gravity tangential to the track. The grade force may be the most significant force experienced by the train other than traction and braking, due to the extremely large mass of the train. This force can be much greater in magnitude than the resistance forces experienced by the train.
$C(v(s))$	Resistance force. In [3], the modelling of the resistance force is briefly described. The resistance force is modelled by the Davis formula which takes the form of a second-order polynomial in $v(t)$ where all the coefficients are nonnegative. $C(v(t)) = C_0 + C_{v0}v(t) + C_{v2}v^2(t)$.

$$\min_{t[k], v[k], \rho[k], \gamma[k], \gamma^+[k], k \in \{q, \dots, N\}} \sum_{i=q}^{N-1} \{w_e \gamma^+[i] \delta s[i] + w_p \rho[i]\} \quad (4.1)$$

s.t.

$$t[q] = t_{\text{predicted}} \quad (4.2)$$

$$s[q] = s_{\text{predicted}} \quad (4.3)$$

$$v[q] = v_{\text{predicted}} \quad (4.4)$$

$$t_{\min} \leq t[N] \leq t_{\max} \quad (4.5)$$

$$s[N] = s_f \quad (4.6)$$

$$v[N] = 0 \text{ m/s} \quad (4.7)$$

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

$$t[k+1] = t[k] + \rho[k] \delta s[k] \quad (4.8)$$

$$v[k+1] = v[k] + \left\{ g[k] - \frac{C_0}{m} \rho[k] - \frac{C_{v0}}{m} v[k] - \frac{C_{v2}}{m} v^2[k] + \gamma[k] \right\} \delta s[k] \quad (4.9)$$

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

$$\left\| \begin{bmatrix} 2 \\ v[k] - \rho[k] \end{bmatrix} \right\|_2 \leq v[k] + \rho[k] \quad (4.10)$$

$$0 \leq \gamma^+[k] \quad (4.11)$$

$$\gamma[k] \leq \gamma^+[k] \quad (4.12)$$

$$0 \text{ m/s} < v[k] \leq v_{\max}[k] \quad (4.13)$$

$$0 \text{ s/m} < \rho[k] < \infty \text{ s/m} \quad (4.14)$$

$$(u_{\text{br,max}}) \rho[k] \leq \gamma[k] \quad (4.15)$$

$$\gamma[k] \leq (U_{\text{tr,max}}) \rho[k] \quad (4.16)$$

$$\gamma[k] \leq r_0 \rho[k] + r_v + r_{v2} v[k] \quad (4.17)$$

Equations (4.8) and (4.9) are state, time and velocity, evolution constraints based on forward Euler discretization of the differential equations. The speed limits are given in (4.13), the traction limits in (4.16) and (4.17), and the braking limits in (4.15). Equation (4.5) ensures that the train arrives within the scheduled arrival window. Equations (4.11) and (4.12) are used to isolate the tractive commands, which are then included in the objective to minimize energy. Equations (4.10) and (4.14) are part of the convex approximation as detailed in [8] and briefly explained in Appendix A. The objective, (4.1), includes a weighted sum of tractive force applied, for minimizing energy, and a weighted sum of $\rho[k]$, for minimizing time and more importantly to minimize error of $\left| \rho[k] - \frac{1}{v[k]} \right|$ so that the approximated dynamics for the convex approximation is equal to the actual dynamics. Minimizing only the sum of $\rho[k]$ corresponds to a minimum-time journey. A sufficiently large value for the weight on $\rho[k]$, w_p , is always required regardless of whether the journey is minimum-energy or minimum-time to ensure that the approximated dynamics in the convex formulation match the actual dynamics of the system.

Optimal control for the train system given by the dynamics in (2) is performed using Algorithm 1. Figure 2 shows the problem setup and a graphical overview of Algorithm 1. An overview of Algorithm 1 is provided as follows.

The position discretization nodes, denoted by s_i in Fig. 2, are selected once initially from the origin to the destination before beginning the trip. The discretization nodes are not modified again. The optimal control problem is successively solved at the pre-selected discretization nodes, i.e. see Fig. 2. The set of remaining position discretization nodes is prepended with an additional discretization node, $s[q]$, for the predicted state at the predicted position. This allows the initial conditions, time and velocity, and continuity of control, current control, to be fixed at the predicted position discretization node. This additional node is later discarded when the optimal solution is returned from the solver.

At each discretization node, the optimal control solution returns the control commands, and time and velocity states at the remaining discretization nodes. The time and control values are used in an interpolation scheme to generate control command values at regular control time steps of T_s , which are then issued to the train propulsion and braking subsystems. In solving the optimization problem, first the position and velocity of the train are predicted over a time horizon of T_D . This is accomplished by numerically integrating the train dynamics in the time-domain using the known control commands in the delay buffer. The predicted states serve as the initial state constraints in the optimal control formulation.

It is noted that due to the use of a model-based predictive control strategy the equal traction and braking delays are simply handled through model-based prediction. With the predicted states after of the application of the control commands in the delay buffer, the convex optimization formulation in our earlier work in [8] can be employed to solve the optimal control problem for the remainder of the trip. This approach would not be applicable in the case of non-equal delays in

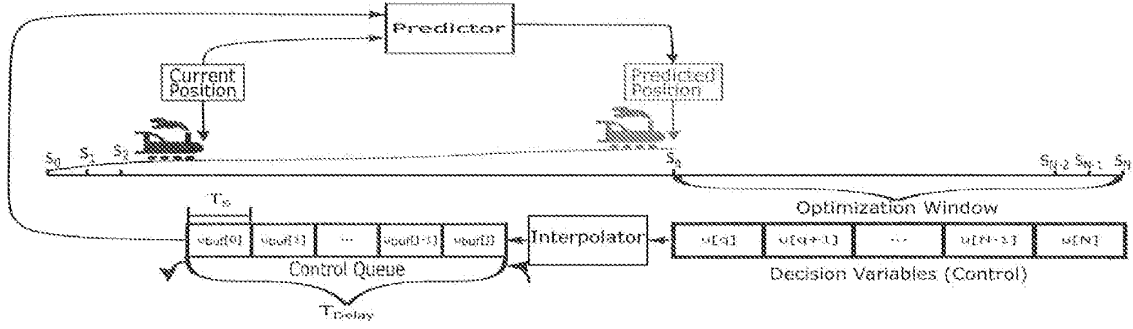


Fig. 2. Optimal control of a train with equal delays in traction/braking. Position, velocity, acceleration, traction effort, and braking effort profiles have to be determined for the entire trip from s_0 till s_N by solving an optimal control (optimization) problem.

TABLE II
TIME-DOMAIN: NOTATION AND VARIABLES

Symbol	Quantity
t	Time
t_0	Initial time
t_{qe}	Time at the end of the longest queue, equal to $t[m]$ in the optimization formulation
t_f	Final time
s_f	Final position: position at destination
$s(t)$	Position
$v(t) = \frac{ds(t)}{dt}$	Velocity
$u(t)$	Control acceleration
$g(s(t))$	Grade force
$C(v(t))$	Resistance force, $C(v(t)) = C_0 + C_1 v(t) + C_{p2} v^2(t)$, where all the coefficients are nonnegative [3].

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

```

1 begin
2   Initialize  $T_d$ 
3   Initialize  $s[k]$  for  $k \in \{0, \dots, N\}$ 
4    $t \leftarrow 0$ 
5    $t_{start} \leftarrow 0$ 
6    $s_{next} \leftarrow 0$ 
7   Measure  $t, s, v$ 
8   while  $s < s_f$  do
9     if  $s \geq s_{next}$  then
10      Function Prediction(Guideway Data,  $t, s, v$ )
11        { $u_{buf}[0], \dots, u_{buf}[N]$ }: is
12         $t_{start} \leftarrow t$ 
13         $t_{predicted} \leftarrow t$ 
14        while  $t_{predicted} - t_{start} < T_{delay}$  do
15          Integrate ODE using values from
            the control queue
16           $t_{predicted} \leftarrow t_{predicted} + dt$ 
17        return  $t_{predicted}, s_{predicted}, v_{predicted}$ 
18      Function Optimization(Guideway Data,
19         $t_{predicted}, s_{predicted}, v_{predicted}$ ,
20        { $s[q], \dots, s[N]$ }) is
21        Optimize
22        return ( { $t^*[q], \dots, t^*[N]$ },
23          { $u^*[q], \dots, u^*[N]$ },
24          { $u^*[q], \dots, u^*[N]$ } )
25       $s_{next} \leftarrow$  Find next discretization node
        greater than  $s_{next}$ 
26    if  $t - t_{start} \geq T_d$  then
27       $t_{start} \leftarrow t$ 
28       $u_{current} \leftarrow$  Interpolate ( $t + T_{delay}$ ,
29        { $u^*[q], \dots, u^*[N]$ }, { $t^*[q], \dots, t^*[N]$ })

```

traction and braking, which will be handled with a separate formulation in the next section. However in such case, it is also possible to artificially delay the input commands with shorter delays so that both channels would have equal delays, and then apply the control approach presented in this section. These two approaches to handling of non-equal delays will be compared in the results section.

III. OPTIMAL CONTROL WITH NON-EQUAL TRACTION AND BRAKING DELAYS

Some trains exhibit non-equal delays in the their traction and braking subsystems. This section presents a new formulation of the optimal control problem in the time-domain to accommodate such cases.

In this case, the continuous-time train dynamics are given by

$$\frac{ds(t)}{dt} = v(t) \quad (5)$$

$$m \frac{dv(t)}{dt} = -C(v(t)) + g(s(t)) + u_{tk}(t - T_{tk}) - u_{bk}(t - T_{bk}) \quad (6)$$

The control problem is formulated by assuming that the inputs are piecewise constant. A predictor compensates for the "common" portion of the delays by predicting the states

using the model of the train, and traction and braking control commands in the delay buffers. The optimal control problem is re-solved whenever the train passes pre-selected position markers along the track. The position markers are chosen once at the beginning of the trip and are not modified again for the remainder of the trip; these are not the same as the discretization nodes used in the optimization formulation.

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

There is a change of independent variable, as is detailed in [27], for the continuous-time differential equations from t to the variable τ , which takes on values in the interval $\tau \in [0, 1]$. The relationship between τ and t is given by:

$$\tau = \frac{t - t_{qe}}{t_f - t_{qe}} \quad (7)$$

where t_{qe} is defined as in Fig. 3.

The quantity $h = t_f - t_{qe}$ is a scalar that determines the trip time. The path of travel is known and fixed. The goal is to determine the velocity profile over the path that will minimize a weighted sum of energy and travel time. In this formulation, the velocity profile and the position profile that follows from it determine the magnitude of the single variable h , which in turn determines the trip time. The variable h is used in the dynamic equations that describe the behaviour of the train system dynamics.

As shown in Fig. 3, the discretization nodes in the optimization formulation include: (i) a fixed number of nodes separated by constant T_h intervals corresponding to a constraint on either the traction/braking control depending on which has the longer delay, and (ii) nodes derived from converting τ from the continuous-domain to the discrete-domain. The τ discretization is chosen to be uniform. The number of τ discretization nodes is equal to the number of remaining position markers from the current train position until the destination, i.e. once the train passes a position marker, the number of τ discretization nodes decreases by one. The set of τ discretization nodes changes as the train progresses along the track because the number of nodes decreases; however, τ is always in the interval $[0, 1]$.

Note that the optimization problem is being solved from the predicted time, $t_{\text{predicted}}$, till the end of the trip, t_f . The optimal control formulation, for a train at a position marker $s_{\text{Milepost}}[q]$, is given by:

$$\min_{s[k], v[k], u_{\text{trk}}[k], u_{\text{brk}}[k], h, k \in \{0, \dots, N+m\}} J \quad (8.1)$$

s.t.

$$t[0] := t_{\text{predicted}} \quad (8.2)$$

$$s[0] := s_{\text{predicted}} \quad (8.3)$$

$$v[0] := v_{\text{predicted}} \quad (8.4)$$

$$t_{\min} \leq t[N+m] \leq t_{\max} \quad (8.5)$$

$$s[N+m] := s_f \quad (8.6)$$

$$v[N+m] := 0 \text{ m/s} \quad (8.7)$$

for $k \in [0, \dots, N+m]$:

$$u_{\text{trk}}[k] u_{\text{brk}}[k] = 0 \quad (8.8)$$

$$s_{\min}[k] \leq s[k] \leq s_{\max}[k] \quad (8.9)$$

$$0 \text{ m/s} \leq v[k] \leq v_{\max}[s[k]] \quad (8.10)$$

$$0 \text{ kg}\cdot\text{m/s}^2 \leq u_{\text{trk}}[k] \leq u_{\text{trk},\max}[v[k]] \quad (8.11)$$

$$0 \text{ kg}\cdot\text{m/s}^2 \leq u_{\text{brk}}[k] \leq u_{\text{brk},\max} \quad (8.12)$$

for $k \in [1, \dots, m]$:

$$s[k] = f_1(s[0], \dots, s[m], v[0], \dots, v[m]) \quad (8.13)$$

$$v[k] = f_2(v[0], \dots, v[m],$$

$$u_{\text{trk}}[0], \dots, u_{\text{trk}}[m-1],$$

$$u_{\text{brk}}[0], \dots, u_{\text{brk}}[m-1]) \quad (8.14)$$

$$\delta u_{\text{trk},\min} \leq \frac{\Delta u_{\text{trk}}[k]}{\Delta t[k]} \leq \delta u_{\text{trk},\max} \quad (8.15)$$

$$\delta u_{\text{brk},\min} \leq \frac{\Delta u_{\text{brk}}[k]}{\Delta t[k]} \leq \delta u_{\text{brk},\max} \quad (8.16)$$

for $k \in [0, \dots, m-1]$:

$$u_{\text{trk}}[k] = \bar{u}_{\text{trk}}[k] \quad (8.17)$$

for $k \in [m+1, \dots, N+m]$:

$$s[k] = f_3(s[m], \dots, s[N+m],$$

$$v[m], \dots, v[N+m], h)$$

$$v[k] = f_4(v[m], \dots, v[N+m],$$

$$u_{\text{trk}}[m], \dots, u_{\text{trk}}[N+m],$$

$$u_{\text{brk}}[m], \dots, u_{\text{brk}}[N+m], h)$$

$$\delta u_{\text{trk},\min} \leq \frac{1}{h} \frac{\Delta u_{\text{trk}}[k]}{\Delta \tau[k-m]} \leq \delta u_{\text{trk},\max} \quad (8.20)$$

$$\delta u_{\text{brk},\min} \leq \frac{1}{h} \frac{\Delta u_{\text{brk}}[k]}{\Delta \tau[k-m]} \leq \delta u_{\text{brk},\max} \quad (8.21)$$

$$t[k] = t[m] + h \sum_{i=0}^{k-m-(m+1)} \delta \tau[i] \quad (8.22)$$

Note that $u_{\text{trk},\max}[v[k]] = \min \left\{ U_{\text{trk},\max}, \frac{P}{v[k]}, \frac{hP}{(v[k])^2} \right\}$, i.e. see (3).

Some important notes regarding the optimization problem in (8) follow next. The differences in the implementation of this case and the equal traction and braking delay case is that the optimal control solution returns at each discretization node: control, time state, position state, and velocity state. The position state is now a variable that has to be determined by the optimization solver. The position state is returned for each discretization node. The time state is only returned for the nodes in the τ -region.

Equations (8.13) to (8.17) are constraints in the "constraints on control" region in Fig. 3. Equations (8.18) to (8.21) are constraints after the "constraints on control" in the "optimization window" region in Fig. 3. Equations (8.8) to (8.12) apply for discretization nodes in both the time and τ -domain regions, i.e. "optimization window" region in Fig. 3.

Equations (8.13), (8.14), (8.18) and (8.19) are state evolution constraints and are left general as any discretization scheme can be used. Equation (8.17) serves as constraints on the control input with the longer delay, in this case braking, as shown in the "constraints on control" region in Fig. 3. Equation (8.8) ensures that both traction and braking are not active at the same time as it may result in equipment damage or increased wear and tear. Equation (8.9) ensures that the train departs from the origin and arrives at the destination without overshooting. The speed, traction, and braking limits are given in (8.10), (8.11), and (8.12), respectively. Equation (8.22) determines the time state at each discretization node in the τ -region, and (8.5) ensures that the train arrives within the scheduled arrival window. Finally, (8.15), (8.16), (8.20) and (8.21) are limits on the change in control due to the equipment limitations, equipment protection, and passenger comfort.

A. Objective Function

A sample objective function is presented in (8.23). Note that a zero-order hold is assumed on the control input for samples associated with T_s (time-domain), while a first-order hold is assumed on the control input for samples associated with $\delta\tau[k]$ (τ -domain). The objective is to minimize a weighted sum of the trip time, energy consumption, and change in control.

A penalty term $\sum_{k=m}^{k=N-1} w_{\Delta ctrl} |\pi_{\Delta ctrl}[k]|$ is added to smooth out the resulting optimal controls. This is done primarily for reasons of passenger comfort, to filter out solutions that consist of cycles of traction and braking.

$$J = w_t h + w_e \sum_{k=m}^{k=N-1} p[k] T_s + w_e \sum_{k=m}^{k=N-1} \frac{1}{2} (p[k] + p[k+1]) h \delta\tau[k] + w_{\Delta ctrl} \sum_{k=m}^{k=N-1} |\pi_{\Delta ctrl}[k]| \quad (8.23)$$

where

- * $p[k] = u_{trk}[k] \text{ or } [k]$
- * $\pi_{\Delta ctrl}[k] = \frac{u[k+1] - u[k]}{T_s}$, $k = 0, \dots, m-1$ for time-domain samples
- * $\pi_{\Delta ctrl}[k] = \frac{u[k+1] - u[k]}{h\delta\tau[k]}$, $k = m, \dots, N+m$ for τ -domain samples
- a) For minimum-time $w_t = 1$, $w_e = 0$, $w_{\Delta ctrl} = 0$
- b) For minimum-energy $w_t = 0$, $w_e = 1$, $w_{\Delta ctrl} = 0$

Algorithm 2 details how the optimal control is computed in the case of non-equal traction and braking delays. Also, Fig. 3, provides a graphical overview of the algorithm. In this figure, it is assumed that the braking delay is larger than the traction delay but the problem formulation can be easily revised for the opposite case. Separate buffers of different lengths are used to store the traction and braking commands. The lengths of the buffers are dictated by the corresponding delays in the

traction and braking subsystems and the control update rate, T_s in Algorithm 2 and Fig. 3. The samples enter the buffer from the right end and exit from the opposite end. The optimization window is from the end of the prediction window till the end of the trip.

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

```

1 begin
2   Initialize  $T_s$ 
3   Initialize  $S_{k|stepout}[k]$  for  $k \in \{0, \dots, L\}$ 
4    $T_{min\_Delay} \leftarrow \min(T_{brk}, T_{trk})$ 
5    $t \leftarrow 0$ 
6    $S_{next} \leftarrow 0$ 
7   Measure  $t, s, v$ 
8   while  $s < s_f$  do
9     if  $s \geq S_{next}$  then
10      Function Prediction(Guideway Data,  $t, s, v$ )
11        { $u_{trk,buf}[0], \dots, u_{trk,buf}[j-1]$ },
12        { $u_{brk,buf}[0], \dots, u_{brk,buf}[j-1]$ }: is
13         $t_{start} \leftarrow t$ 
14         $t_{predicted} \leftarrow t$ 
15        while  $t_{predicted} - t_{start} < T_{min\_Delay}$  do
16          Integrate ODE using values from
17            the traction/braking control queues
18           $t_{predicted} \leftarrow t_{predicted} + dt$ 
19        return  $t_{predicted}, S_{predicted}, v_{predicted}$ 
20      Function Optimization(Guideway Data,
21         $t_{predicted}, S_{predicted}, v_{predicted}$ )
22        { $u_{trk,buf}[0], \dots, u_{trk,buf}[j-1]$ },
23        { $u_{brk,buf}[0], \dots, u_{brk,buf}[j+m-1]$ }: is
24        Optimize
25        return ( $\{t^*[q], \dots, t^*[N]\}$ ,
26           $\{s^*[q], \dots, s^*[N]\}$ ,
27           $\{v^*[q], \dots, v^*[N]\}$ ,
28           $\{u_{trk}^*[q], \dots, u_{trk}^*[N]\}$ ,
29           $\{u_{brk}^*[q], \dots, u_{brk}^*[N]\}$ )
30       $S_{next} \leftarrow$  Find Next Discretization Node
31        Greater Than  $S_{next}$ 
32    if  $t - t_{start} \geq T_s$  then
33       $t_{start} \leftarrow t$ 
34       $u_{trk} \leftarrow$  Interpolate ( $t + T_{trk}$ ,
35        { $u_{trk}^*[q], \dots, u_{trk}^*[N]$ },
36        { $t^*[q], \dots, t^*[N]$ })
37       $u_{brk} \leftarrow$  Interpolate ( $t + T_{brk}$ ,
38        { $u_{brk}^*[q], \dots, u_{brk}^*[N]$ },
39        { $t^*[q], \dots, t^*[N]$ })

```

Note that the prediction window in Algorithm 2 is shown in Fig. 3. The length of the fixed time-steps in the prediction window are determined by the control command input rate. The optimal control problem is formulated over the end of the prediction, $t_{predicted}$, till the end of the trip, t_f . The predicted states constitute the initial constraints on the system state in the optimization problem. For the control buffer with the longer

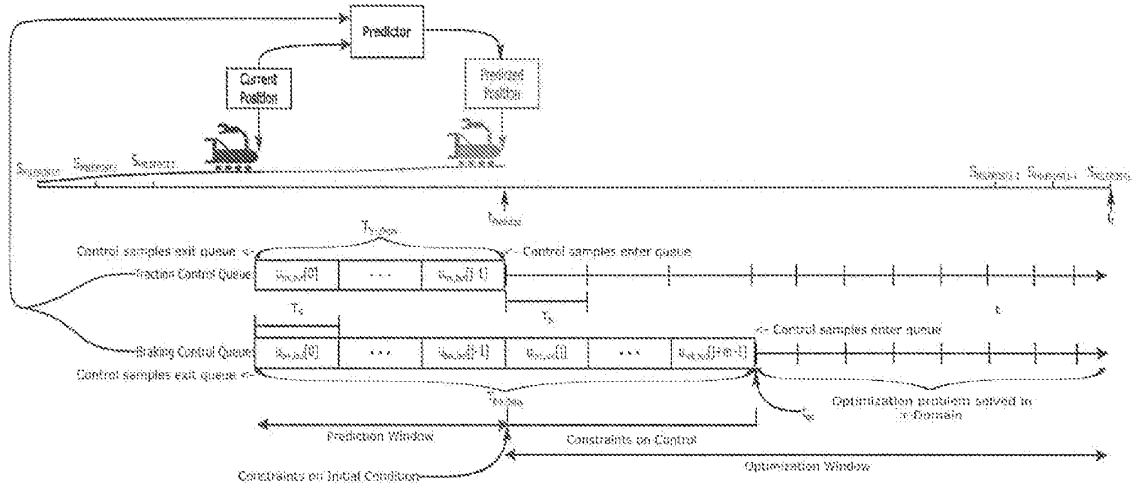


Fig. 3. Optimal control of train system with traction and braking as separate inputs, each with its own unique actuation delay. The optimization problem for the section “constraints on control” is in time-domain while the optimization problem from the end of that segment till the end of the trip is in τ -domain.

delay, the control samples from the end of the prediction till the end of the buffer constrain the corresponding control input in the optimization formulation. The fixed time-steps during this period are of the length T_b , in Fig. 3. The optimization problem will be solved in both the time-domain and the τ -domain. Part of the constraints will be in time-domain, and part of the constraints will be in τ -domain. This is shown in Fig. 3.

IV. COMPUTER IMPLEMENTATION AND RESULTS

The optimization problem s in (4) is convex and can be solved for a global optimum with any nonlinear programming solver or any dedicated convex optimization solver. The optimization problem in (8) is nonconvex and nonlinear and can be solved for a local optimum using any nonlinear programming software. The nonlinear programming solver may solve for a global optimum; however, it is not guaranteed.

The computer implementation results for both the convex and the nonconvex problem are produced by using IPOPT [28] interfaced to MATLAB using the OPTI Toolbox [29]. IPOPT implements the primal-dual interior-point algorithm [28].

The IPOPT solver is provided with the values of the objective function and its gradient, constraints, Jacobian of the constraints and its sparsity pattern, and the Hessian of the Lagrangian and its sparsity pattern. The IPOPT default termination criteria are used [30].

The optimization problem is re-solved at predetermined positions along the track. The control commands found from the solution of the optimization problem are interpolated to generate the actual control commands at regular time intervals until the train hits the next discretized position at which point the optimization solver is recalled.

The simulations were carried out on an Intel(R) Core(TM) i7-6700HQ CPU 2.60GHz with 16.0 GB of RAM running on Microsoft Windows 10 operating system. Computation times

are provided for each closed-loop optimization run for the nonconvex problem. The numbers for the convex problem are already available in our work in [8]. For nonconvex optimization, the solution times can vary widely because of the nonconvexity. The starting point for the initial optimization is chosen at random, and for any subsequent optimization problems the solution from the previous step is used. The computation times are measured from the internal computer clock. The time shown is actual time rather than CPU time. There are multiple processes running at the same time, and it is at the sole discretion of the operating system which process to schedule on the CPU at what time. The solver is set to time out after 100 s for all optimizations except the initial optimization, which can be run off-line before the train departure.

A. Equal Delays in Traction and Braking using Convex Formulation

The convex optimization formulation presented in (4) is used here. The formulation parameters are:

- $N = 401$
- $T_{\text{trk}} = 2.8$ s
- $T_{\text{brk}} = 2.8$ s
- $m = 1$ kg
- $t[1] = 0$ s
- $s[1] = 0$ m
- $v[1] = 0$ m/s
- $t_{\text{max}} = 1000$ s
- $s[N] = 10,000$ m
- $v[N] = 0$ m/s
- $v_{\text{max}}[k] = 100$ m/s for
- $k \in \{1, \dots, N\}$
- $u_{\text{brk,max}} = -1.25$ m/s²
- $u_{\text{trk,max}} = 1.25$ m/s²
- $r_0 = 1.327e0$ kg•m/s²
- $r_v = -3.663e-2$ kg/s
- $r_{v^2} = 2.528e-4$ kg/m
- $c_0 = -0.1$ kg•m/s²
- $c_v = -0.01$ kg/s
- $c_{v^2} = -0.001$ kg/m
- $w_p = 1000$
- $w_n = 10,000$

The system in the convex case has equal traction and braking delays. While this case considers equal traction and braking delays of 2.8 s, its results can also be used as reference

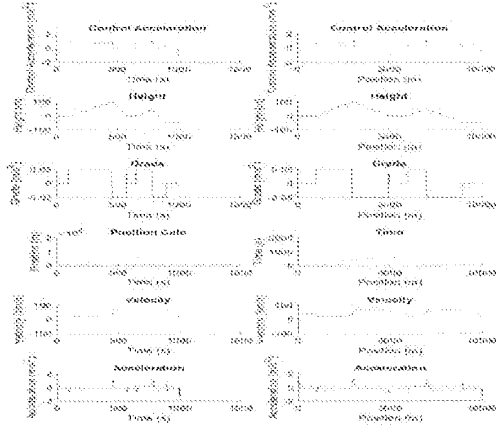


Fig. 4. Convex approximation: minimum-energy considering delays in the control input. Closed-loop trajectory. Final Position: 10000.73 m. Overshoot: 0.73 m. Trip Time: 1003.73 s. Trip Delay: 3.73 s. Energy Consumed: 3463.84 J.

for comparison with another case discussed later in which the traction delay of 1.4 s is shorter than the braking delay, 2.8 s; the nonconvex formulation will be used to solve that problem. Alternatively, addition of artificial delay of 1.4 s to the traction commands renders both delays equal, resulting in the case considered here with the convex formulation.

The results are shown in Fig. 4. A relatively large value of w_e would put more weight on minimizing energy, and a relatively large value of w_p would place more weight on minimizing travel time. A sufficiently large value of w_p is necessary for the formulation to work as explained in Section II. As can be seen from Fig. 4, the trip is very similar to the minimum-energy scenario for the nonconvex formulation in Fig. 9, which will be presented later. This is because of the relatively large value of the energy weight term, w_e , compared to the travel time weight term, w_p . A minimum-energy journey is characterized by more time spent coasting and less time spent braking (in the case of no regenerative braking). This is because braking results in energy loss. For example, a minimum-energy trip with no maximum arrival time is characterized by phases of extremal and off control: periods of traction and coasting. There is no braking because there is no time-limit and the energy of the system can be reduced to zero due to resistance losses alone.

The details of computation times in this case are available [8], which shows numbers in the range of milliseconds to hundreds of milliseconds with the solver always being able to find a globally optimal solution due to the convexity of the problem formulation.

B. Non-equal delays in Traction and Braking using Nonconvex Formulation

The nonconvex optimization formulation presented in (8) is used to solve the control problem for the case of non-equal

delays in traction and braking. The values of the parameters are:

- $N = 201$
- $m = 40$
- $T_{tk} = 1.4$ s
- $T_{brk} = 2.8$ s
- $T_s = 0.07$ s
- $m = 1$ kg
- $\ell[1] = 0$ s
- $s[1] = 0$ m
- $v[1] = 0$ m/s
- $t_{max} = 1000$ s
- $s[N] = 10,000$ m
- $v[N] = 0$ m/s
- $v_{max}[k] = 100$ m/s for
- $k \in \{1, \dots, N\}$
- $u_{brk,max} = -1.25$ m/s²
- $U_{max} = 1.25$ kg•m/s²
- $P = 12.5$ kg•m²/s³
- $k_p = 12.5$ m/s
- $\delta u_{trk,min} = -0.2$ kg•m/s³
- $\delta u_{trk,max} = 0.2$ kg•m/s³
- $\delta u_{trk,min} = -0.2$ kg•m/s³
- $\delta u_{brk,max} = 0.2$ kg•m/s³
- $c_0 = -0.1$ kg•m/s²
- $c_v = -0.01$ kg/s
- $c_{v^2} = -0.001$ kg/m

1) *Minimum-Time Optimal Control*: The parameters in the objective function are selected as $w_t = 1$, $w_{\Delta cm} = 0$, and $w_e = 0$, which correspond to minimum time optimal control. The control problem is solved with and without considering the input delays. The results are shown in Figs. 5 to 7. The computation times for the case of ignoring delays in traction/braking are similar to Fig. 7. As can be seen from these results, when ignoring the delays, the train significantly overshoots its destination by about 25.34 m, which is obviously unacceptable. When the delays are factored in the computation of the optimal traction/braking commands, the train slightly undershoots the destination by about 0.27 m. The energy usage is only slightly lower when considering delays. A minimum-time journey is characterized by intensive traction, minimum time spent coasting, and braking near the destination. For example, a minimum-time trip with no minimum arrival time is characterized by extremal, bang-bang, control: phases of maximum traction and braking.

2) *Minimum-Energy Optimal Control*: The gains in the objective function are $w_t = 0$, $w_{\Delta cm} = 1$, and $w_e = 1$ to move the train from its origin to destination using the minimum amount of energy subject to maximum arrival time of $t_{max} = 1000$ s.

The results of simulations are shown in Figs. 8 to 10. If the delays are ignored, the train would stop 5.89 m past its destination. Considering the delays in the control inputs results in a slightly higher energy usage ($\leq 0.1\%$) but reduces the overshoot to a mere 0.07 m. Minimizing energy would entail driving at lower speeds, thus increasing the travel time. This is because losses in energy occur due to friction and braking. Friction losses increase as the speed increases, thus increasing energy consumption.

Note that the problem considered here can be solved by adding an artificial delay to the traction commands and using the convex formulation of the optimal control. Comparing these results with those from Fig. 4, it turns out that the convex formulation with the additional delay results in a slightly larger energy usage ($\approx 0.60\%$) and longer trip time ($\approx 0.17\%$). Given that the convex formulation has a much shorter computation time and can be solved more robustly than the nonconvex formulation, the slight degradation in the

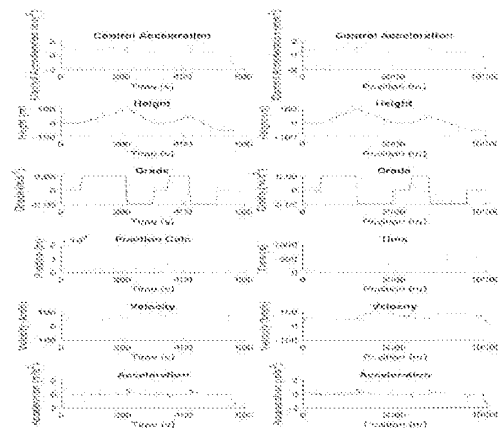


Fig. 5. Minimum-time optimal control by ignoring non-equal traction and braking delays; closed-loop responses. Final Position: 10025.34 m. Overshoot: 25.34 m. Trip Time: 567.0 s. Trip Delay: N/A. Energy Consumed: 6364.06 J.

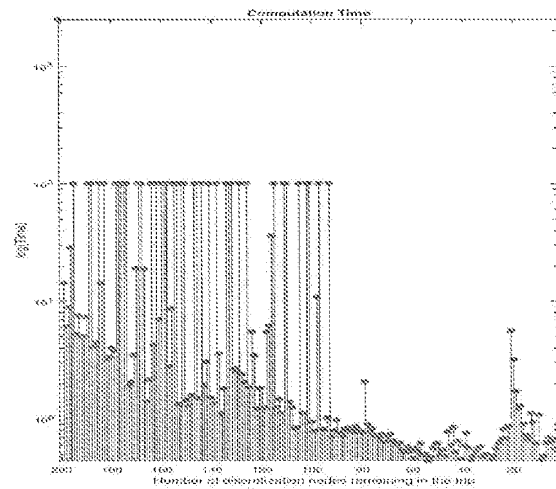


Fig. 7. Minimum-time optimal control by considering non-equal traction and braking delays; computation times. The computation times for the case of minimum-time optimal control ignoring non-equal traction and braking delays is found to be similar.

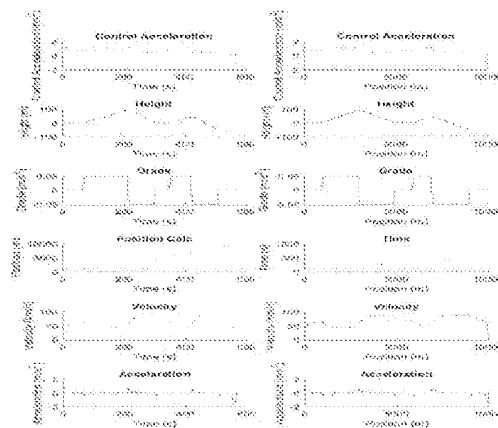


Fig. 6. Minimum-time optimal control by considering non-equal traction and braking delays; closed-loop trajectory. Final Position: 9999.73 m. Overshoot: -0.27 m. Trip Time: 568.54 s. Trip Delay: N/A. Energy Consumed: 6354.93 J.

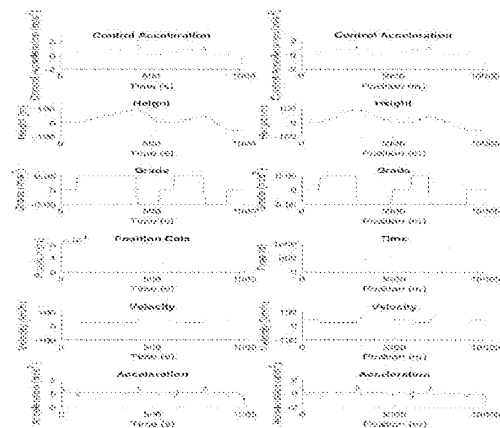


Fig. 8. Minimum-energy by ignoring non-equal traction and braking delays; closed-loop trajectory. Final Position: 10005.89 m. Overshoot: 5.89 m. Trip Time: 997.36 s. Trip Delay: N/A. Energy Consumed: 3423.78 J.

controller performance due to the extra delay may very well be acceptable in many scenarios.

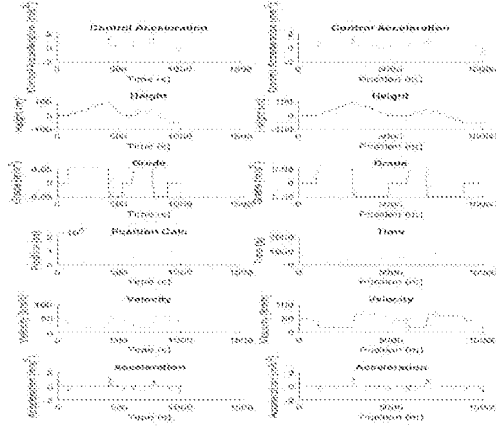


Fig. 9. Minimum-energy by considering non-equal traction and braking delays; closed-loop trajectory. Final Position: 10000.07 m. Overshoot: 0.07 m. Trip Time: 1002.03 s. Trip Delay: 2.05 s. Energy Consumed: 3443.09 J.

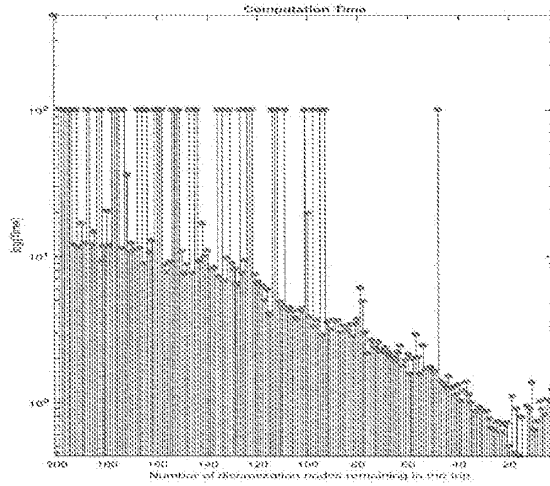


Fig. 10. Minimum-energy by considering non-equal traction and braking delays; computation times. The computation times for the case of minimum-energy optimal control ignoring non-equal traction and braking delays is found to be similar.

V. CONCLUSION

In this paper, optimal control control of a commuter train under the presence of traction and braking delays was studied. Solutions were proposed for the two cases of equal and non-equal delays. In the case of equal delays, a convex constrained optimization model was presented that employed a model of the system in the discrete position domain resulting in a fast global solution to the problem of mixed energy-time optimal control. For non-equal delays, a nonconvex optimization model using the model of the system in the discrete time domain was developed. Both of these controllers incorporate model-

based predictors to handle the actuator delays. The control formulations are quite general and provide a mechanism to achieve optimal mixed energy-time objectives under various user-specified operational constraints.

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

APPENDIX A CONVEX FORMULATION OF OPTIMAL MIXED ENERGY-TIME TRAIN CONTROL

This appendix provides an overview of the convex approximation detailed in [8]. The original train optimal control problem formulated as an optimization problem, i.e. before the convex approximation is applied, is as follows.

$$\min_{t[k], v[k], u[k], k \in \{q, \dots, N\}} w_c \left\{ \sum_{i=q}^{N-1} u^+[i] \delta s[i] \right\} + w_t t[N] \quad (9.1)$$

s.t.

$$t[q] = t_{\text{predicted}} \quad (9.2)$$

$$s[q] = s_{\text{predicted}} \quad (9.3)$$

$$v[q] = v_{\text{predicted}} \quad (9.4)$$

$$t_{\min} \leq t[N] \leq t_{\max} \quad (9.5)$$

$$s[N] = s_f \quad (9.6)$$

$$v[N] = 0 \text{ m/s} \quad (9.7)$$

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

$$t[k+1] = t[k] + \frac{1}{v[k]} \delta s[k] \quad (9.8)$$

$$v[k+1] = v[k] + \left\{ \frac{(q[k] - C_u) - C_v}{m v[k]} - \frac{C_w}{m} v[k] + \frac{u[k]}{v[k]} \right\} \delta s[k] \quad (9.9)$$

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

$$0 \leq u^+[k] \quad (9.10)$$

$$u[k] \leq u^+[k] \quad (9.11)$$

$$0 \text{ m/s} < v[k] \leq v_{\max}[k] \quad (9.12)$$

$$u_{\text{brk}, \max} \leq u[k] \leq \min \left\{ U_{\text{trk}, \max}, \frac{P}{v[k]}, \frac{kP}{(v[k])^2} \right\} \quad (9.13)$$

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

$$t[k+1] = t[k] + \rho[k] \delta s[k] \quad (10)$$

$$v[k+1] = v[k] + \left\{ \frac{(g[k] - C_0) \rho[k]}{m} - \frac{C_v}{m} - \frac{C_{v^2}}{m} v[k] + \gamma[k] \right\} \delta s[k] \quad (11)$$

along with the following algebraic equation

$$\frac{1}{v[k]} = \rho[k] \quad (12)$$

Note the following:

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

The optimal control problem is now transformed to the disciplined convex form [31], thus ensuring convexity. Equation (12) is relaxed from equality to the inequality constraint,

$$\frac{1}{v[k]} \leq \rho[k] \quad (13)$$

Equation (13) is also known as an hyperbolic constraint, and in its current form is nonconvex, but can be transformed into the following constraint that is convex [32],

$$\left\| \begin{bmatrix} 2 \\ v[k] - \rho[k] \end{bmatrix} \right\|_2 \leq v[k] + \rho[k] \quad (14)$$

To ensure that $\rho[k] \approx \frac{1}{v[k]}$, the term $\rho[k]$ is included in the objective function with a significant enough weight to ensure that the inequality is active.

The dynamics are now characterized by the difference equations (9.8) and (9.9), and the algebraic equation (14). Note, that this approximation only holds if (14) is active.

The constraints on the control, (9.13), have to be approximated by convex functions for the optimization problem to be convex, see [8].

WHAT IS CLAIMED IS:

1. A controller for a vehicle, the controller comprising:
a processor configured to repeatedly perform, at each control iteration among a plurality of control iterations along a path of the vehicle from a start location to a target location, operations including
based on a current state of the vehicle, generating a travel plan for a remainder of the path from a current position of the vehicle to the target location, by solving an optimization problem, and
controlling a motoring and braking system of the vehicle to execute the generated travel plan until a next control iteration among the plurality of control iterations.
2. The controller of claim 1, wherein
the processor is configured to solve the optimization problem based on
the current state of the vehicle,
at least one control objective, and
at least one constraint.
3. The controller of claim 1, wherein
the optimization problem includes a convex second order cone problem.
4. The controller of claim 1, wherein
the processor is configured to change, in real-time while the vehicle is travelling on the path, one or more of

the at least one control objective, and

the at least one constraint.

5. The controller of claim 1, wherein

the at least one constraint comprises a first brake constraint and a second brake constraint, the second brake constraint corresponding to a harder brake operation than the first brake constraint,

the processor is configured to

solve the optimization problem constrained by the first brake constraint,

in response to a determination that the optimization problem constrained by the first brake constraint is feasible, control the motoring and braking system of the vehicle based on a solution of the optimization problem,

in response to a determination that the optimization problem constrained by the first brake constraint is infeasible, solve the optimization problem constrained by the second brake constraint corresponding to the harder brake operation.

6. The controller of claim 5, wherein

the optimization problem includes a convex second order cone problem, and the processor is configured to determine whether the convex second order cone problem is feasible or not in a single control iteration.

7. The controller of claim 1, further comprising:

a communication circuit configured to perform communication with a further controller of a trailing vehicle that follows the vehicle with the controller on the path from the start location to the target location,

wherein the processor is further configured to repeatedly perform, at each control iteration among the plurality of control iterations along the path from the start location to the target location,

causing the communication circuit to transmit a driving profile corresponding to the travel plan generated in said control iteration to the further controller of the trailing vehicle to be used by the further controller as a constraint in generating a further travel plan for the trailing vehicle.

8. The controller of claim 1, further comprising:

a communication circuit configured to perform communication with a further controller of a leading vehicle that the vehicle with the controller follows on the path from the start location to the target location,

wherein the processor is further configured to repeatedly perform, at each control iteration among the plurality of control iterations along the path from the start location to the target location,

causing the communication circuit to receive, from the further controller, a further travel plan of the leading vehicle, and

using the received further travel plan of the leading vehicle as a constraint in said generating the travel plan for the vehicle with the controller.

9. The controller of claim 1, wherein
- the processor is further configured to repeatedly perform, at each control iteration among the plurality of control iterations along the path from the start location to the target location,
 - estimating a parameter of the vehicle, and
 - using the estimated parameter in said generating the travel plan for the vehicle.
10. The controller of claim 1, wherein
- the processor is further configured to perform said generating the travel plan for the remainder of the path from the current position of the vehicle to the target location by
 - determining, between the current position and the target location, a predicted position of the vehicle corresponding to a delay of the motoring and braking system,
 - solving the optimization problem to determine a first part of the travel plan between the predicted position and the target location, and
 - determining, based on the first part of the travel plan, a second part of the travel plan between the current position and the predicted position.
11. A system for controlling a plurality of vehicles along a common path, the system comprising:
- a first controller associated with a leading vehicle among the plurality of vehicles, the first controller configured to
 - generate a first driving profile, and
 - control movement of the leading vehicle along the path based on the generated first driving profile; and

a second controller associated with a trailing vehicle among the plurality of vehicles, the second controller configured to

receive the first driving profile generated by the first controller,
generate a second driving profile based on the received first driving profile, and
control movement of the trailing vehicle along the path and following the leading vehicle based on the generated second driving profile.

12. The system of claim 11, wherein

the first and second controllers further comprise first and second communication circuits, respectively, configured to perform one-way communication with each other to transmit the first driving profile from the first controller to the second controller.

13. The system of claim 11, wherein

each of the first and second controllers is configured to repeatedly perform, at each control iteration among a plurality of control iterations along the path from a start location to a target location, operations including

based on a current state of the corresponding leading or trailing vehicle,
generating a travel plan for the corresponding leading or trailing vehicle on a remainder of the path from a current position of the corresponding leading or trailing vehicle to the target location,
and

controlling a motoring and braking system of the corresponding leading or trailing vehicle to execute the generated travel plan until a next control iteration among the plurality of control iterations, and

the second controller is further configured to perform said generating the travel plan for the trailing vehicle based on the first driving profile received from the first controller, the first and second driving profiles corresponding to the travel plans generated by the first and second controllers, respectively, for the leading vehicle and the trailing vehicle, respectively.

14. The system of claim 11, wherein

the second controller is configured to generate the second driving profile by solving an optimization problem using the received first driving profile as a constraint.

15. The system of claim 14, wherein

the first controller is configured to generate the first driving profile by solving a further optimization problem, and

the optimization problem solved by the second controller and the further optimization problem solved by the first controller have at least one common control objective.

16. The system of claim 14, wherein

the first controller is configured to generate the first driving profile by solving a further optimization problem, and

the optimization problem solved by the second controller and the further optimization problem solved by the first controller have different control objectives.

17. The system of claim 11, further comprising:

a third controller associated with a further trailing vehicle among the plurality of vehicles, the further trailing vehicle following the trailing vehicle,

wherein the third controller is configured to

receive the second driving profile generated by the second controller,

generate a third driving profile based on the received second driving profile, and

control movement of the further trailing vehicle along the path and following the trailing vehicle based on the generated third driving profile.

18. A method of controlling a vehicle, the method comprising:

performing convex optimization to generate a travel plan for the vehicle along a path; and

controlling a motoring and braking system of the vehicle to execute the generated travel plan to move the vehicle along the path,

wherein the convex optimization comprises a second order cone problem.

19. The method of claim 18, further comprising:

defining a plurality of discretization nodes on the path between a start location and a target location; and

repeatedly performing, at each discretization node among the plurality of discretization nodes,

the convex optimization, based on a current state of the vehicle, to generate a control vector including control elements corresponding to discretization nodes remaining on the path from a current position of the vehicle to the target location, and

controlling the motoring and braking system of the vehicle to execute a first control element among the control elements of the generated control vector to move the vehicle to a next discretization node among the discretization nodes remaining on the path.

20. The method of claim 19, wherein
- saïd defining comprises distributing the plurality of discretization nodes
 - at a predetermined distance from each other, or
 - at a predetermined time interval from each other.