Verified Train Controllers for the Federal Railroad Administration Train Kinematics Model: Balancing Competing Brake and Track Forces

Aditi Kabra Stefan Mitsch and André Platzer
Computer Science Department, Carnegie Mellon University, Pittsburgh, PA, USA
Email: {akabra | smitsch | aplatzer}@cs.cmu.edu

***Abstract*—Automated train control improves railroad operation by safeguarding the motion of trains while increasing efficiency by enabling motion within a safe envelope. Train controllers decide when to slow trains down to avoid collisions with other trains on the track, stay inside movement authorities, and navigate slopes, curves and tunnels safely. These systems must base their decisions on detailed motion models to guarantee the absence of overshoot of the movement authority (safety) and limit undershoot (efficiency). This paper is the first to formally verify the safety of the Federal Railroad Administration freight train kinematics model with all its relevant forces and parameters, including track slope and curvature, air brake propagation, and resistive forces as computed by the Davis equation. Due to the significant competing influence of these parameters on train stopping distances, even designing train controllers is a nontrivial control challenge, which we solve using formal verification. For increased generality at reduced verification effort, we verify symbolic mathematical generalizations of the train control models and subsequently apply efficient uniform substitutions to obtain verification results for physical train control models.**

***Index Terms*—train control, formal verification, hybrid systems, differential dynamic logic**

I. Introduction

EMBEDDED software for many complex cyber-physical systems like trains, planes, and self-driving cars is safety-critical: errors can have disastrous consequences. In order to ensure the safety of controllers, formal verification with computer-checked, repeatable mathematical proofs presents a particularly trustworthy method for controller design. This paper uses formal verification as a tool to design and verify a train controller, which is a practically important, representative problem with challenges common in other safety-critical embedded systems: complex dynamics with transcendental arithmetic, competing forces with subtle interaction, and effects whose exact magnitude is unknown at proof time.

Train controllers decide when to enforce braking to prevent movement authority violation and collisions. They must account for all the competing influences that govern train motion. Uphill slopes decrease velocity, for example, which decreases resistance, which permits a more rapid increase in velocity, slope and curve effect, all while the train's brake force builds gradually until saturation as air pressure propagates along brake pipes. These complex interactions make it hard to design a safe and efficient train controller, and even harder to ensure it is always safe. This paper designs and verifies train controllers for the Federal Railroad Administration (FRA) freight train kinematics model [1], [2] (henceforth FRA model), contributing generalizable verified controller design techniques.

Existing studies of formally verified train motion [3]–[5] do not account for at least two effects amongst track grade, track curvature, resistance, and air brake propagation time, rendering their results inapplicable to most real-world scenarios. We surmount the challenges of verification against the full dynamics of the FRA model, in which these effects interact subtly with each other. Our verification results are significant, because these parameters influence the motion of the train in safety-critical and/or performance-critical ways. Neglecting track slope profile and the gradual propagation of air pressure braking, in particular, can render otherwise verifiably safe train controllers unsafe, since their influence may diminish the train's ability to decelerate, causing collisions. Our verification is valid for realistic FRA models [1], [2].

Before verifying controller safety, we first design the controllers, balancing efficiency with provable safety. Conservative controllers are mathematically more simplistic, and easier to design and verify, but make railway operations inefficient, violating performance objectives. We start by presenting a conservative safe controller and then demonstrate how to iteratively make it more efficient by exploiting characteristics of the physical train dynamics for better but safe control.

Train controllers are assessed relative to a (changing) destination stopping point—called *end of movement authority*: overshoot of the end of movement authority is a safety violation, because that risks collision with other trains; efficiency is measured in terms of end of movement authority undershoot. We prove absence of end of movement authority overshoot when using our controllers in the FRA model by verification and demonstrate efficiency by simulation.

To keep our proofs as general and widely applicable as possible, we leverage nondeterministic controllers and a paradigm of mathematical abstraction. Each controller is intentionally built to be set-valued such that *all* of its control choices are simultaneously proved safe under all circumstances in the FRA

Manuscript received April 07, 2022; revised June 11, 2022; accepted July 05, 2022. This article was presented at the International Conference on Embedded Software (EMSOFT) 2022 and appeared as part of the ESWEEK-TCAD special issue. The work described in this paper was funded by the Federal Railroad Administration Office of Research, Development and Technology under contract number 693JJ620C000025.
Digital Object Identifier 10.1109/TCAD.2022.3197690

model. The safety of these controllers implies the safety of *all* their specializations [6], giving railroads significant freedom in how to adapt the verified controllers for their purposes. Controller verification follows a two stage process: we first prove mathematical models of abstract train control motion, and then obtain proofs of the actual physical models of train control by *uniform substitution* [7] to replace the abstract function symbols of the mathematical models with physical terms specific to the FRA model or even specific railroads.

Crucially, our approach uses three different types of models: (i) the high-fidelity physics model describing the kinematic motion of trains along the track, (ii) our generalized mathematical abstractions of the physics model, and (iii) the simplified but *computable* approximations of motion models used by the respective train controllers. Our verification results prove that the safety of (i) derives from the safety of (ii) and that all control decisions following (iii) are safe in (ii).

Our proof is written in differential dynamic logic [7], [8], and performed using the hybrid systems theorem prover KeYmaera X [9]. We compare the efficiency with concrete control algorithms [2] for a number of train consists (arrangement of locomotives and cars) and scenarios [2]. Our verified models act as safety envelopes for unverified controllers via runtime enforcement checks using ModelPlex [10]. We illustrate their behavior in simulation. The proved models are permissive and only interfere in train control operation when acting otherwise risks movement authority violation.

Contributions: The primary contribution of this paper is the formal verification results justifying the safety of train control in the FRA model. The secondary contribution is the design of new, safe train controllers and their permissive set-valued safe control envelopes. Technical contributions are the ideas of the formal proofs, and of substituting a specialized ground physics model into controllers designed with an abstract mathematical model. Finally, our simulations demonstrate the impact of verified train controllers in different terrain. Proofs are available at `https://doi.org/10.1184/R1/19542610` with ideas sketched inline.

II. Related Work

Due to their significance as safety-critical transportation systems, there have been many efforts to verify the safety of train control systems [11]. One approach is extensive simulation of train braking models [12], [13]. However, simulation can only show safety in a limited number of cases and is less appropriate when free acceleration is interspersed with braking. Similar limitations apply to test-based safety assurance of train control [14]. Our work uses differential dynamic logic (dL), a logic with a deductive proof system for hybrid systems [7], [8]. We write mathematical proofs that guarantee safety over the infinite state space in a model of physical motion.

In the realm of formal verification, there have been many studies of railway systems [15]–[17]. Discrete aspects of train control have been verified at industrial scale [18], [19]. Many studies [20]–[24] focus on scheduling trains to avoid route intersections. Train communication systems have been formally verified [16], [25]. Such studies are complementary to our work, which focuses on the motion of the train as it interacts with the environment. Results on the correctness of the motion of trains permit correct interaction with scheduling.

Studies of train motion have verified the European train control system with moving blocks [3], and the Chinese train control system [4], while ignoring the effect of resistance, air brakes, track grade and curvature. The FRA model has been verified while ignoring grade and curve [5]. Our work differs by accounting for *all* forces in the FRA model, creating a controller designed and verified against realistic physics.

III. Background

A. Differential Dynamic Logic

Differential dynamic logic dL is a logic with a deductive proof system for hybrid systems [7], [8]. We give a short overview.

Differential dynamic logic extends first-order logic with the notion of *hybrid programs*. A hybrid program runs according to a binary relation between states, mapping start states to end states that a program could reach. The program constructs include assignment, for example, $x := e_1$ which instantaneously assigns expression e_1 to variable x. In the special case of nondeterministic assignment, $x := *$, the transition relation accounts for any possible real value being assigned to x.

The test operator, as in $?F$, aborts the current run if formula F is false. The continuous evolution operator, $\{x' = f(x)\&Q\}$ follows the ordinary differential equation (ODE) $x' = f(x)$ for some nondeterministic amount of time, with evolution domain constraint Q being true throughout the evolution. Sequential composition, $\alpha;\beta$, runs program α followed by program β, for example, the discrete train controller α followed by the train's ODE β. The nondeterministic choice operator $\alpha\cup\beta$ runs either program α or β, for example, either accelerate the train with α or brake with β. The loop operator α^* runs hybrid program α any nondeterministically chosen $n \geq 0$ times. It is important for running a train control loop indefinitely. To express safety properties about hybrid programs, we use the box modality $[\alpha]F$, which is true in any state from which all runs of hybrid program α end in states in which the formula F is true.

For proofs of dL formulas, we use dL inference rules [26], [27]. The rules most relevant to the present work are *loop* (loop), *differential invariant* (dI), and *differential cut* (dC).

$$\text{(loop)}\ \frac{\Gamma \vdash J \qquad J \vdash [\alpha]J \qquad J \vdash P}{\Gamma \vdash [\alpha^*]P}$$

$$\text{(dI)}\ \frac{\Gamma \vdash e_1 \leq e_2 \quad Q \vdash [x' := f(x)]e_1' \leq e_2'}{\Gamma \vdash [x' = f(x)\&Q]e_1 \leq e_2}$$

$$\text{(dC)}\ \frac{\Gamma \vdash [x' = f(x)\&Q \wedge C]P \quad \Gamma \vdash [x' = f(x)\&Q]C}{\Gamma \vdash [x' = f(x)\&Q]P}$$

As usual, the loop rule uses an invariant J that holds initially, inductively after each step, and implies the post condition that we seek to prove. A differential invariant preserves properties along the flow of a differential equation: if $e_1 \leq e_2$ initially and e_1 grows slower than e_2, so $e_1' \leq e_2'$ (where e_1' and e_2' are evaluated after substituting in the assignment $x' := f(x)$ from the differential equation), then it remains true that $e_1 \leq e_2$. A more general rule form and its explanation can be found in the literature [7], [27]. The idea behind a

differential cut is that if formula C holds true at the end of every possible run of differential equation $x' = f(x)$, then C must hold true throughout its evolution. Differential cuts can be used to accumulate knowledge about a differential equation.

B. FRA Model of Train Kinematics

This section introduces the FRA model, which provides the forces acting on a train [1, Eq. (1)]. After the net force on the train has been identified, Newton's second law, using train mass, determines the acceleration that the train experiences, which in turn determines change in velocity and change in position for train control design. The forces are

$$\sum F = -F_G - F_C - F_B + F_L - F_R = m_T a \qquad (1)$$

where F_G denotes force due to track grade, F_C resistance due to track curvature, F_B force from the brakes, F_L force from the locomotive engine (tractive effort), F_R resistive forces, and m_T, train mass. Newton's second law, $\sum F = m_T a$, determines train acceleration a. Resistive force F_R follows the modified Davis equation [1, Eq. (18)]:

$$F_R = A_R w + B_R n + C_R w v + D_R v^2 \ ,$$

where A_R, B_R, C_R, D_R are experimentally determined positive constants whose numerical value given a choice of units can be found in other sources [2], [28], [29]. Further, n is the number of axles, and w is the weight of the train.

Grade and curve forces depend on the train position p on the track. Grade force [1, Eq. (19)] is proportional to train weight w and average track grade $\mathit{grade}(p)$ underneath the train:

$$F_G = A_G w \cdot \mathit{grade}(p) \ .$$

Similarly, curve force [1, Eq. (21)] is a function of average track curvature $\mathit{curve}(p)$ along the train and train weight w:

$$F_C = A_C w \cdot \mathit{curve}(p) \ .$$

where A_C and A_G are positive multiplicative constants. Braking force can be modeled as the minimum of two linear functions to capture the effect of air pressure brake force buildup [1, Fig. 17] and stabilization. Let F_{B0} be the force acting immediately on brake application, f_p, the slope with which brake force increases, t, the elapsed time, and $F_{B\max}$, the force after air pressure in the air pressure brakes saturates.

$$F_B = min(F_{B0} + f_p t, F_{B\max}) \ .$$

Brake enforcement and train protection algorithms [1], [2] approximately solve a differential equation derived from Eq. (1) to estimate the velocity and position of the train at future times.

C. Mathematical Model Abstraction

In order to maximize generality of the embedded software design, minimize verification effort, and simplify future proof maintenance, we present a mathematical abstraction of the FRA model [1], [2]. Concrete verified controllers and their safety proofs for the fully expanded model can be obtained automatically from the verified abstract model by uniform substitution [7].

Our abstract train kinematic model in (2) is an ODE in time. The rate of change of position is velocity, and the rate of change of velocity is acceleration. The variables and constants involved, along with their signs, when relevant, are (i) Train position p, (ii) velocity v, (iii) velocity and position-independent component of acceleration a_l ranging from immediate braking ability $-b_{\max} < 0$ to maximum train engine acceleration $a_{\max} > 0$, (iv) acceleration due to air brakes a_a in range $a_{b\max} < 0$ to 0, (v) rate of change m_b of air brake acceleration, which is $m_p < 0$ when brakes are ramping up and 0 otherwise, (vi) map a_s from position to acceleration due to grade, (vii) map a_c from position to acceleration due to curvature, and (viii) velocity-dependent resistance a_r. In the chosen sign convention, resistive acceleration is negative.

$$p' = v, v' = a_l + a_a + a_s(p) + a_r(v) + a_c(p), a_b' = m_b \qquad (2)$$
$$\text{with } a_l \in [-b_{\max}, a_{\max}], a_a = \max(a_b, a_{b\max}), m_b \in \{0, m_p\}$$

The Davis equation resistance $a_r(v) = -\frac{C_R w v + D_R v^2}{m_T}$ has the shape $a_r = a_1 v + a_2 v^2$ when $a_1 = -C_r g$ with gravity g summarizes the linear coefficient of velocity, and $a_2 = -\frac{D_R}{m_T}$ summarizes the quadratic coefficient. Grade and curvature are represented by unspecified but bounded functions a_s and a_c that map train positions to a numeric value for acceleration due to slope and average curvature, respectively. The quantity a_l summarizes locomotive tractive effort ($a_l \geq 0$) and train deceleration ($a_l < 0$) as commanded by the train controller, with adjustment for the velocity-independent resistance.

We later instantiate the proved abstract kinematic train model by dL's uniform substitution [7] to easily get proofs for specific physical train models such as the FRA model. Similarly, proofs for specific train configurations result from also substituting values for coefficients, or even for a specific train state when additionally substituting speed and position.

IV. Model Structure

We develop a conservative train controller in dL based on the abstract train kinematic model Eq. (2) of Section III-B, presenting it modularly and introducing conceptually important model components and functions along the way. We address the challenge of representing track grade and curve, which are unknown at proof time, using unspecified maps. In order to reason about them, we bound the maps with assumptions quantifying over all arguments. Our solution permits us to capture the full Eq. (2) without conservatively neglecting $a_s(p)$ and $a_c(p)$ when reasoning about it during verification. It generalizes to other embedded software that must reason about unspecified, bounded functions, such as noise or potential fields (e.g. electro-magnetic or gravitational effect).

We prove the controller safe: relative to Eq. (2), the controller will provably never permit the train's position to exceed the end of movement authority e, though it might be inefficient, braking unnecessarily early. Later, by revising modular components and functions to be more arithmetically sophisticated, Section V will retain provable safety but make the controller more efficient. Fig. 1 shows the relationship between the resulting controller models.

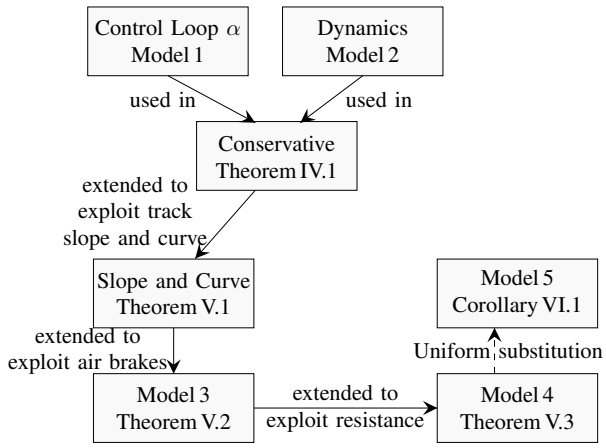


Fig. 1. Relationship between train models in this paper.

A. Model Description

The train controller consists of a time control loop. The control has a latency of time $T > 0$—the controller has to wait at most this long before being able to change the throttle position. In practice, reaction time T is typically to the order of 1 second, but train controllers often keep decisions in effect for a 10 second period without revising them [29]. Every control cycle, the controller computes an overapproximation $\texttt{stopDist}(p, v, a_b)$ of the *stopping distance*, the distance that the train will travel before stopping if it were to accelerate during the current control cycle, but then brake continuously starting at the next control cycle, until it comes to a halt. If the distance left to the end of movement authority exceeds $\texttt{stopDist}(p, v, a_b)$, the controller continues free driving (with any acceleration or deceleration choice within the physical limits of the train), otherwise it brakes. This control cycle, $\alpha(\texttt{stopDist})$, is parametric in stopping distance `stopDist`, and expressed as a hybrid program in Model 1.

Model 1 Train control $\alpha(\texttt{stopDist})$.

Reset Timer	1	$t := 0;$
	2	$\big($ $(?(e - p > \texttt{stopDist}(p, v, a_b)));$
Free Driving	3	$a_l := *;\ ? - b_{\max} \leq a_l < a_{\max};$
	4	$a_b := 0;\ m_b := 0)$
Braking	5	$\cup\ a_l := -b_{\max};\ m_b := m_p\big)$

a) Train Dynamics: Model 2 describes the physical motion of the train according to the abstract mathematical model of Eq. (2) augmented with a clock $t' = 1$ to switch back to control after at most time T. *We leave all constants symbolic and prove safety for all values*, so that individual railroads have the flexibility to instantiate them with the values that apply to their system and inherit our safety results. This generality in controller design comes at the cost of higher proof complexity, but compared to the alternative where we would initialize these constants with a conservative value, this allows for more efficient, tailored controllers adapted to the specific rail operation.

Model 2 Train dynamics.

Dyn.	1	$p' = v, a_b' = m_b, t' = 1,$
	2	$v' = a_l + \max(a_b, a_{b\max}) + a_s(p) + a_r(v) + a_c(p)$
	3	$\quad \& \ t \leq T \wedge v \geq 0$

b) Stopping Distance: In order to decide between free driving and braking, the controller computes the upper bound $\texttt{stopDist}(p, v, a_b)$ on the distance covered over one time period of acceleration and subsequently braking to a stop.

Thus, our models provide two distinct specifications of the distance that the train will take to stop. The first, indirect specification is through the differential equation in Model 2 that implicitly describes the physical motion of the train. The second, approximate specification is `stopDist` in Line 2 of Model 1, an explicit arithmetic expression that the controller can evaluate efficiently to make decisions at runtime.

Efficiency concerns demand that $\texttt{stopDist}(p, v, a_b)$ be as tight as possible; if the bound is too large, the controller would enforce braking unnecessarily. But verifiable safety requires $\texttt{stopDist}(p, v, a_b)$ to provably be an upper bound on the distance that the train covers (as determined by the dynamics). The tightest possible bound is the exact solution of the differential equation. However, even ignoring the effect of air brakes, the differential equation requires trigonometric solutions. Transcendental function arithmetic is undecidable [30]. To ensure mathematical provability[1], we develop polynomial approximations, which is a delicate design task because automated decision procedures for polynomial real arithmetic validity are computationally expensive [31], [32]. This constrains the complexity of the polynomial approximations that can be used as upper bounds. We therefore strike a balance between conflicting concerns: striving for efficiency while satisfying mathematical provability.

To illustrate this approach, we start with a simple conservative expression for `stopDist`. This expression is similar in complexity to previous work [3], [5], but is now proved safe for the full model including slope, curve friction, air brake propagation, and aerodynamic drag. We later improve on this approximation, focusing on one contributing factor at a time.

B. Stopping Distance: Conservative

This section constructs a first, conservative controller by instantiating control loop α with an expression for `stopDist`, and proving it safe. Referring back to the train dynamics in Eq. (2), we first need an upper bound for v. Integrating this bound via $p' = v$ computes a stopping distance upper bound.

The first impediment to obtaining a provable upper bound for v is that grade and curvature maps a_s and a_c are arbitrary functions, constrained only by upper and lower bounds, and bounded gradients. At runtime, the train knows their exact values as the controller is instantiated with maps for the

[1]In KeYmaera X, even when manually simplifying differential and modal expressions, arithmetic subgoals are outsourced to arithmetic decision procedures, which are subject to limitations in proving real arithmetic.

railroad it runs on. However, these maps are unknown at proof time. And yet, the proof has to show safety of the train control ahead of time for *all* possible track maps in order to justify safety of the train controller. In order to obtain a provable upper bound on stopping distance, the proof therefore bases on the limited information that we do have about the maps: upper bounds on the potential values of a_s and a_c.

A naïve upper bound on a_s is the value of acceleration that the train experiences when it is on the steepest permissible downward slope, m_s. Our proof will show that the distance required to stop for any permissible grade map cannot exceed the distance computed with the steepest downward slope. It first shows that the true acceleration is bounded above by an acceleration that uses the highest permissible value of grade acceleration, then that actual velocity cannot exceed the velocity computed using the worst-case acceleration, and consequently, that traveled distance cannot exceed the stopping distance computed using the worst-case estimate of velocity.

Accounting for grade force is important. On a downward hill, for example, a train with a controller that ignores grade would roll forward even at the end of its movement authority which may cause accidents. In contrast, we can safely ignore curve resistance when approximating `stopDist`, since resistances shorten stopping distance (upper bound 0). These simplifications result in differential equation $v_2' = a_l + m_s + a_r(v)$, since $max(a_b, a_{b\max}) = 0$ while the train is accelerating, where v_2 is the upper bound on v that we integrate to compute `stopDist`. However, the solution $v_2(t)$ is still transcendental:

$$\frac{\left(\sqrt{4(a_l+m_s)a_2 - a_1^2}\right)\cdot \tan\left(t\frac{\sqrt{4(a_l+m_s)a_2-a_1^2}}{2} + \tan^{-1}\left(\frac{a_1+2a_2v_0}{\sqrt{4(a_l+m_s)a_2-a_1^2}}\right)\right) - a_1}{2a_2}$$

The culprits are the linear and quadratic terms in velocity from the Davis equation. With another simplification of 0 as an upper bound for a_r (resistance always works against the train's motion), we derive a polynomial expression for `stopDist`:

$$\begin{aligned} v_3' = a_l + m_s \quad &\Rightarrow \quad v_3(t) = v_0 + (a_l + m_s)\cdot t \\ p_3' = v \quad &\Rightarrow \quad p_3(t) = p_0 + (v_0 + \frac{a_l + m_s}{2}\cdot t)\cdot t \end{aligned} \tag{3}$$

where v_0 and p_0 are the initial values of speed and position. The solutions provide a conservative stopping distance bound.

$$\begin{aligned} \texttt{stopDist}_b(p, v, a_b) = \underbrace{vT + \frac{a_{\max} + m_s}{2}\cdot T^2}_{\text{distance while accelerating}} \\ + \underbrace{\frac{(v + (a_{\max} + m_s)T)^2}{2(b_{\max} - m_s)}}_{\text{stopping from increased speed}} \end{aligned} \tag{4}$$

The conservative stopping distance $\texttt{stopDist}_b$ ignores its arguments p and a_b, but later refinements of `stopDist` functions also depend on p and a_b, which is why they are passed in.

The first two terms are the distance covered by the train in one control cycle of acceleration, while the third term is the distance that the train needs to stop should it start braking right after, assuming the worst-case value of 0 for a_a. This conservative distance is similar to what has been used in the literature [3], but has been adjusted to account for grade force with its worst-case accelerating or decelerating effects. Substituting Eq. (4) into control cycle α results in the dL hybrid program of the conservative controller: $\alpha(\texttt{stopDist}_b)$.

A physically important quantity is *braking distance*, the distance that the train will travel before coming to a stop should it start braking right now. We derive an upper bound, $\texttt{brakeDist}_b(v, a_b)$, which will be crucial to our proofs and the initial assumptions, and is improved upon later.

$$\texttt{brakeDist}_b(v, a_b) = \frac{v^2}{2(b_{\max} - m_s)} \tag{5}$$

The last term of Eq. (4) is $\texttt{brakeDist}_b(v_3(T), 0)$ with v_3 according to Eq. (3) when $a_l = a_{\max}$.

Initial assumptions $\texttt{init}(\texttt{brakeDist})$ parametrized by `brakeDist`, and `initAirbrake` (assumptions on air brakes) are required to prove the conservative controller safe. Assumptions about unspecified functions are represented by universal quantification over their input. This representation permits derivation of a formula about the unspecified function at any point of the train's evolution by substituting the quantified input with current values.

$$\begin{aligned} \texttt{init}(\texttt{brakeDist}) = a_{\max} > 0 \land b_{\max} > 0 \land a_1 < 0 \land a_2 < 0 \\ \land e - p > \texttt{brakeDist}(v, 0) \land b_{\max} - m_s > 0 \land v \geq 0 \\ \land m_s \geq 0 \land T > 0 \land \forall x\,(|a_s(x)| \leq m_s) \land \forall x\,(a_c(x) \leq 0) \\ \texttt{initAirbrake} = m_p < 0 \land a_{b\max} < 0 \land m_b = 0 \land a_b = 0 \end{aligned} \tag{6}$$

Theorem IV.1 presents the dL formula representing the safety of the conservative controller.

Theorem IV.1. *The conservative braking controller guarantees that the train always remains within the end of the movement authority. The* dL *formula below is provable, where* α *is the control loop from Model 1 parameterized with Eq.* (4) *for* `stopDist`.

$$\begin{aligned} \texttt{init}(\texttt{brakeDist}_b) \land \texttt{initAirbrake} \\ \rightarrow [(\alpha(\texttt{stopDist}_b); \textit{Model 2})^*]\, e - p > 0 \end{aligned}$$

Proof. The proof has been done in the theorem prover KeYmaera X, but we present its central ideas here. We use loop invariant $e - p \geq \texttt{brakeDist}_b(v, 0) \land a_b \leq 0 \land v \geq 0$ and split into cases for free driving and braking corresponding to the nondeterministic choice in Lines 2–5 of Model 1. On braking, the invariant is maintained because the derivative of the distance that the train will take to come to a stop does not exceed the derivative of the distance to the end of movement authority, i.e., $(\texttt{stopDist}_b)' \leq (e-p)'$, by dI. On free driving for a control period T, we first restate that the train maintains a distance to the end of movement authority of at least `stopDist` adjusted for time t since the last control decision, i.e., $v(T-t) + \frac{a_{\max}+m_s}{2}(T-t)^2 + \frac{(v+(a_{\max}+m_s)(T-t))^2}{2(b_{\max}-m_s)}$ by dC and dI. The required inequality relation between the derivatives, $-v \geq -v + v'(T-t) - (a_{\max}+m_s)(T-t) + \frac{(v+(a_{\max}+m_s)(T-t))(v'-(a_{\max}+m_s))}{(b_{\max}-m_s)}$, holds because $v' - (a_{\max} + m_s) \leq 0$. □

V. Safe Efficiency Improvements

We improve on the overapproximation for stopping distance in order to make our controller more efficient. The FRA model presents two challenges common in embedded controllers: it uses functions whose exact values are unknown at proof time (slope and curve maps), and has many interacting forces. Our techniques address these problems with two general principles: using quantified worst case bounds on unknown functions, and separation of dependencies. The first technique relies on the observation that the track changes are gradual and predictable (the rate of change of unknown functions is bounded). It drastically improves bounds on the effect of grade and curve over one time period of acceleration, after resolving *circular dependencies* between the variables of motion. The second technique improves the estimate `brakeDist` by accounting for air brake dynamics. It demonstrates a handling of triple integration, using *mode-splitting* to deal with the non-analytical change of behavior when brakes saturate. The third technique uses *Taylor polynomials* to capture the effect of resistance, which would otherwise lead to transcendental arithmetic. The track environment discussion in Section V-A and Section V-B will become assumptions of our models, while the calculations in Section V-C and Section V-D are machine-checked to be correct approximations as part of the KeYmaera X proofs.

A. Bound on Gradient

The train controller knows the current slope $a_s(p)$ and vertical curves of the track, which determine transitions from one track grade to another. This knowledge results in a bound $h_{\max}$ on the difference in grade per unit length [28, p.616–619]:

$$\left|\frac{\partial a_s(x)}{\partial x}\right| \leq h_{\max} \quad \Rightarrow |a_s'(p)| \leq v h_{\max}$$

The second inequality follows from the first using the chain rule and $p' = v$. After time T, a_s could have increased by no more than $u h_{\max} T$, where u is some upper bound on v over the course of T time, which we derive later in this section.

B. Bound on Curve Resistance

Similar to bounding the gradient change, we compute an upper bound on the rate of change of curve resistance as a function of velocity using track geometry. Curve resistance depends on average curve *curve*(p).[2] Assume that the tightest permissible curvature for the railroad corresponds to radius r. The greatest change in average curvature occurs when a train goes from a track with the greatest permissible curvature to a straight track (or vice versa). Over a small period of time dt, the portion of the train transitioning from greatest curvature to 0 curvature is dv, where v is velocity. So the rate of change of *curve*(p) (taken in radians) with respect to time is $-\frac{v}{lr}$ where l is the length of the train. For a given train, a_c relates to *curve* with some constant multiplicative factor q. We use $a_c' = q\frac{v}{lr} = c_{cft} v$ with the constant factor $c_{cft} = \frac{q}{lr}$.

With this bound on the maximum rate of change of a_c, we now estimate the upper bound on curve resistance over time T, where p_0 is initial train position, to be

$$a_c(p_0) + u c_{cft} T$$

As before, u is an upper bound on velocity for duration T.

C. Tight Stopping Distance Approximation

The upper bounds $\overline{a}_s$ on gradient from Section V-A and $\overline{a}_c$ on curve resistance from Section V-B are summarized as:

$$a_s(p) \leq \overline{a}_s(p_0) = \min(m_s, a_s(p_0) + u h_{\max} T)$$
$$a_c(p) \leq \overline{a}_c(p_0) = \min(0, a_c(p_0) + u c_{cft} T)$$

This enables us to improve our estimation of stopping distance:

$$v_4' = a_l + \overline{a}_s(p_0) + \overline{a}_c(p_0)$$
$$\Rightarrow v_4(t) = v_0 + \big(a_l + \overline{a}_s(p_0) + \overline{a}_c(p_0)\big)T \quad (7)$$

Upper bound v_4 is tighter than v_3 of Eq. (3) and thus integrates to an improved stopping distance estimate. It depends (transitively through $\overline{a}_s$ and $\overline{a}_c$) on the unknown upper bound u on velocity, which we still need to estimate provably correctly.

Circular Dependencies: The upper bound on velocity, u, is undefined in expression (7) above. We cannot use the bound v_4 for u, since v_4 itself is phrased in terms of u. The problem is a circular dependency between a_s and v: the bound on slope acceleration a_s depends on speed v, while the upper bound on speed v, in turn, depends on slope acceleration a_s; likewise for a_c. Physically, this is because if the train is moving faster, we know less about the nature of the track—its curve and slope—after the passage of some time, as the train is farther from its previous position on the track. However, we need information about the grade curve in order to better estimate the velocity that the train is traveling at. In order to cut through these circular dependencies, we use the conservative estimations of these quantities from (3) as a base case to bootstrap incrementally finer computations, as presented below.

We first use the initial upper bounds m_s for a_s and 0 for a_c to get a conservative bound $v(t) \geq v_0 + (a_{\max} + m_s)t$, so that we can set $u = v_0 + (a_{\max} + m_s)T$. Since $(a_{\max} + m_s)$ is a positive upper bound on the train's acceleration, velocity could have increased no more than $(a_{\max} + m_s)T$. Hence u is indeed an upper bound on v through the T time interval. Substituting this u refines the gradient and curve resistance bounds.

$$\overline{a}_s(p_0) = \min(m_s, a_s(p_0) + \overbrace{(v_0 + (a_{\max} + m_s)T)}^{u} h_{\max} T)$$
$$\overline{a}_c(p_0) = \min(0, a_c(p_0) + \underbrace{(v_0 + (a_{\max} + m_s)T)}_{u} c_{cft} T)$$

These expressions give the chosen definitions of $\overline{a}_s$ and $\overline{a}_c$ by replacing placeholder velocity bound u.

We could in principle further improve this upper bound on speed by using v_4 to obtain an even better bound on

[2] In practice, degree of curvature is the angle subtended by a 100ft arc of the track, 100ft chord of the track, or 100.7ft arc in the US. In Europe, radius is generally used [28]. In our rate of change of curvature derivation, we will use the mathematical idealization for average degree of curvature: *curve*$(p) = \int_{p-l}^{p} \frac{1}{lr(x)} dx$ where l is the length of the train and $r(x)$ is the radius of curvature at track point x. This approximately relates to the US definition by a multiplicative factor, with a small error introduced by the granularity of the 100ft measuring arc. This multiplicative factor can be rolled into the multiplicative coefficient c_{cft}.

a_s and a_c, which could in turn yield an improved bound on v. However, extra levels of extrapolation increase proof cost and computation time when the controller is run. Each extra intermediate bound requires a constant number of extra proof steps, but provides diminishing efficiency gains in return. Intuitively, proof length is asymptotically linear in number of iterations because under optimal proof rule application ordering, each iteration induces one extra application of rule dC to introduce the intermediate bound into the proof tree, and rule dI to justify this intermediate bound.

The `stopDist` expression below uses v_4 with $u = v_0 + (a_{\max} + m_s)T$ to estimate stopping distance, which is sufficiently tight to make useful control decisions (see Section VII).

$$\texttt{stopDist}_s(p,v,a_b) = vT + \left(\frac{a_{\max} + \overline{a}_s(p) + \overline{a}_c(p)}{2}\right)T^2 - \frac{\Big(v + \big(a_{\max} + \overline{a}_s(p) + \overline{a}_c(p)\big)T\Big)^2}{2(b_{\max} - m_s)} \tag{8}$$

We need further initial assumptions to prove the improved *slope-exploiting controller* $\alpha(\texttt{stopDist}_s)$ safe. These assumptions represent the result of the track environment discussion of Section V-A and V-B used for the computer-checked proof.

$$\begin{aligned}\texttt{init}_s(\texttt{brakeDist}) \equiv\ & \texttt{init}(\texttt{brakeDist}) \\ & \wedge a_{\max} - m_s + m_c > 0 \wedge h_{\max} \geq 0 \\ & \wedge m_c \leq 0 \wedge c_{cft} \geq 0 \wedge \forall x\,(a_c(x) \geq m_c) \\ & \wedge \forall x' \forall x\,(|a_c'| \leq x' c_{cft}) \wedge \forall x' \forall x\,(|a_s'| \leq x' h_{\max})\end{aligned} \tag{9}$$

The technique of this section applies to *time-triggered controllers* (where a control loop runs with some known maximum latency and sensors measure current state every cycle) for physical systems with functions affecting the environment that are unknown except for bounds on their rate of change. The future value of the functions can be bounded in terms of their worst-case rate of change. Furthermore, these bounds can be used to compute bounds on other variables in the system, just as here a bound on velocity was used to bound slope and curve effect, which was again used to obtain a better bound on velocity. The situation arises frequently in practice: examples of unknown functions are a potential field, or a noise or error effect, which may have circular dependence with position.

Theorem V.1. *The slope-estimating controller guarantees that the train stays within its movement authority. The* dL *formula below is provable, where α is the control loop from Model 1 parameterized with Eq.* (8) *for* `stopDist`.

$$\texttt{init}_s(\texttt{brakeDist}_b) \wedge \texttt{initAirbrake} \rightarrow [(\alpha(\texttt{stopDist}_s); \textit{Model 2})^*]\, e - p > 0$$

Proof. By proof in KeYmaera X. The proof builds on the ideas from Theorem IV.1. We apply the loop rule with the same loop invariant as Theorem IV.1. If the train brakes, differential invariant rule dI again shows that the loop invariant holds throughout differential equation evolution.

If the train chooses to accelerate, then as before, we restate that the train maintains at least a distance of $\texttt{stopDist}_s$ adjusted for time t since the last control decision. Unlike before, $\overline{a}_s$ instead of m_s accounts for worst-case a_s, and $\overline{a}_c$ instead of 0 accounts for worst-case a_c. Again, dI proves that this adjusted inequality remains true. In order to prove the required inequality on the derivatives, we first use differential cut rule dC to show $\overline{a}_s(p_0) \geq a_s(p)$ throughout the control cycle, and $\overline{a}_c(p_0) \geq a_c(p)$. There are two branches for each cut corresponding to how the $\min$ in $\overline{a}_s$ and $\overline{a}_c$ resolve. For example, using initial position and velocity p_0 and v_0, for $\overline{a}_s$, we need to show that $a_s(p) \leq m_s$, and $a_s(p) \leq a_s(p_0) + u h_{\max} T$. While the former follows from the quantified assumption on $a_s(p)$, to prove the latter, we again adjust it for elapsed time t, to argue that $a_s(p) \leq a_s(p_0) + (v_0 + u h_{\max}(T - t))$, proved using dI. The required derivative inequality follows from instantiating the quantified assumption bounding the rate of change of $(a_s(p))' \leq u h_{\max}$ with the current position, and showing that u is an upper bound on v in the control loop. The argument for $\overline{a}_c$ is analogous. □

D. Effect of Air Pressure Brakes

The term $\texttt{brakeDist}_b$ conservatively neglects the *significant* effects of air brakes to avoid reasoning about their time dependence. This section derives a tighter $\texttt{brakeDist}_a$ that accounts completely for air brakes. It specifies a controller that simultaneously benefits from the slope and curve estimation of the previous section, and from air brake dynamics. The central insight required to prove the improved controller safe is how to compose reasoning about time-dependent air brake propagation and velocity-dependent slope and curve estimations from the previous section. We first show that $\texttt{brakeDist}_a$, the component of `stopDist` affected by air brakes, is the desired upper bound on distance to brake throughout the control loop. Then, holding $\texttt{brakeDist}_a$ constant, we perform the differential reasoning on slope and curve estimation described in the previous section. The two results together permit an overall proof of safety of the air brake-exploiting controller.

To derive the improved $\texttt{brakeDist}_a$, we first compute some intermediate functions from air brake dynamics. In Eq. (2) during brake rampup, with slope relaxed pessimistically to m_s, and curve and resistance to 0, $\max(a_b, a_{b\max})$ evaluates to a_b, and m_b to m_p. The solution for v in the resulting differential equation $v' = b_{\max} - m_s + a_b, a_b' = m_p$ is quadratic in t:

$$v = v_0 - (b_{\max} - m_s + a_b)t + \frac{1}{2}m_p t^2 \tag{10}$$

Function t_b below computes the time the train takes to achieve full braking by subtracting current brake buildup a_b from maximal air braking $a_{b\max}$, and dividing by the rate of increase in air brake force m_p. If the train comes to a stop before air brake saturation, it instead evaluates to the time until the train stops, as computed by solving Eq. (10) for $v = 0$.

$$t_b(v,a_b) = \min\Big((a_{b\max} - a_b)/m_p, \frac{(b_{\max} - m_s + a_b) - |(b_{\max} - m_s + a_b)^2 - 2m_p v|}{m_p}\Big) \tag{11}$$

The distance that the train travels before either stopping or reaching maximum air brake effect is $\int_0^{t_b(v,a_b)} p\,dt = v t_b(v,a_b) + \frac{1}{2}(b_{\max} - m_s + a_b)t_b(v,a_b)^2 + \frac{1}{6}(m_p)t_b(v,a_b)^3$. The

velocity of the train after this period of buildup, by Eq. (10), is $v_f = v - (b_{\max} - m_s + a_b)t_b(v,a_b) + \frac{1}{2}m_p t_b(v,a_b)^2$. So after the brakes finished ramping up, the distance traveled until the train comes to a halt is $\frac{v_f^2}{2(b_{\max} - m_s - a_{b\max})}$, using Newton's third equation of motion. If the train stops before finishing brake rampup, v_f evaluates to zero, as required. Adding the upper bounds on distance traveled before and after brake rampup results in $\texttt{brakeDist}_a(v,a_b)$ in Eq. (12), an upper bound on braking distance that accounts for the effect of air brakes. While this derivation for controller design is manual, its result will be verified by computer-checked proof in Theorem V.2.

$$\begin{aligned}\texttt{brakeDist}_a(v,a_b) &= vt_b(v,a_b)\\ &- \frac{1}{2}(b_{\max} - m_s + a_b)t_b(v,a_b)^2 + \frac{1}{6}(m_p)t_b(v,a_b)^3\\ &+ \frac{\big(v - (b_{\max} - m_s + a_b)t_b(v,a_b) + \frac{1}{2}m_p t_b(v,a_b)^2\big)^2}{2(b_{\max} - m_s - a_{b\max})}\\ \texttt{stopDist}_a(p,v,a_b) &= vT + \left(\frac{a_{\max} + \overline{a}_s(p) + \overline{a}_c(p)}{2}\right)T^2\\ &+ \texttt{brakeDist}_a\Big(v + \big(a_{\max} + \overline{a}_s(p) + \overline{a}_c(p)\big)T, 0\Big)\end{aligned} \tag{12}$$

In order to prove the invariant that after a control cycle of braking, $e - p > \texttt{brakeDist}_a(v,a_a)$, we must reason about the three dynamically distinct cases: (i) when $\max(a_b, a_{b\max})$ is $a_{b\max}$; (ii) when it is a_b, and $t_b(v,a_b)$ evaluates to $(a_{b\max} - a_b)/m_p$; and (iii) when it is a_b but $t_b(v,a_b)$ evaluates to $\frac{(b_{\max} - m_s + a_b) - |(b_{\max} - m_s + a_b)^2 - 2m_p v|}{m_p}$. We split the model dynamics into these three evolution domains, branching between the three possibilities in a loop to transition between modes freely (Model 3). This does not affect the semantics of evolution: per train dynamics, mode transition happens at most once (from (ii) to (i) or (ii) to (iii)). Unrolling the loop to two iterations, one per mode, suffices to model train behavior. This split formulation simplifies syntactic proofs.

Model 3 Mode-split train dynamics.

$$\text{Dyn.} \left| \begin{array}{ll} 1 & (\quad \{\text{Model 2 } \&\ a_{b\max} \geq a_b\} \\ 2 & \quad \cup \{\text{Model 2 } \&\ a_{b\max} \leq a_b \wedge t_a \geq t_p\} \\ 3 & \quad \cup \{\text{Model 2 } \&\ a_{b\max} \leq a_b \wedge t_a \leq t_p\})^* \end{array} \right.$$

where $t_a = (a_{b\max} - a_b)/m_p$,

$$t_p = \frac{(b_{\max} - m_s + a_b) - |(b_{\max} - m_s + a_b)^2 - 2m_p v|}{m_p}$$

Theorem V.2. *The air-brake-exploiting controller guarantees that the train stays within its movement authority. The* dL *formula below is provable, where α is the control loop from Model 1 parameterized with Eq.* (12) *for* stopDist.

$$\begin{aligned}&\texttt{init}_s(\texttt{brakeDist}_a) \wedge \texttt{initAirbrake}\\ &\qquad \rightarrow [(\alpha(\texttt{stopDist}_a); \textit{Model 3})^*]\, e - p > 0\end{aligned}$$

Proof. By proof in KeYmaera X. The high level idea is to use an outer loop invariant $e - p \geq \texttt{brakeDist}(v,a_a) \wedge a_b \leq 0 \wedge v \geq 0$ and again split into free driving and braking cases. On braking, we show that the outer loop invariant is maintained in each of the three dynamics modes using an inner loop invariant consisting of 4 formulas, the most important of which is $e - p > \texttt{brakeDist}(v, \max(a_b, a_{b\max}))$.

On free driving, we first show that $\texttt{brakeDist}_{a0} = \texttt{brakeDist}_a\big(\big(v + \big(a_{\max} + \overline{a}_s(p) + \overline{a}_c(p)\big)T\big), 0\big)$ is truly an upper bound on all reachable $\texttt{brakeDist}(v, \max(a_b, a_{b\max}))$ values in the control cycle. Then, holding $\texttt{brakeDist}_{a0}$ constant, we follow a proof similar to Theorem V.1 to show that, in every mode, the increase in p does not exceed the decrease in distance buffer $vT + \left(\frac{a_{\max} + \overline{a}_s(p) + \overline{a}_c(p)}{2}\right)T^2$. The free driving inner loop invariant consists of 14 formulas, most of which state that various upper bound expressions (such as on velocity, grade and curve) remain upper bounds over the course of the loop. By monotonicity, then, $e - p >$ distance buffer $+$ $\texttt{brakeDist}_{a0} \geq \texttt{brakeDist}(v, \max(a_b, a_{b\max}))$. □

E. Exploiting Resistance

Exactly accounting for the quadratic dependence of resistance on velocity, as discussed in Section IV-B, leads to an undecidable, transcendental exact solution for stopping distance. The controller must instead use an approximation. Since polynomial arithmetic is decidable, Taylor polynomials are a natural way to obtain decidable approximations. This section applies Taylor approximation to the FRA model, identifying techniques generalizable to verified control for other embedded systems with transcendental dynamics.

The Davis equation implies[3] $v' \geq (a_{\max} + \overline{a}_s(p) + \overline{a}_c(p)) + a_1 v + a_2 v^2$, where slope and curve bounds $\overline{a}_s$ and $\overline{a}_c$ are from Section V-C. The first-order Taylor polynomial of this expression for velocity is $v_0 + \Big((a_{\max} + \overline{a}_s(p) + \overline{a}_c(p)) + a_1 v_0 + a_2 v_0^2\Big)t$. Using this approximation at time T, with $a_l = a_{\max}$, as an upper bound for velocity after a time period of acceleration, we compute Eq. (13) for stopping distance that leverages resistance. While this derivation is manual, its result will be verified by computer-checked proof in Theorem V.3.

$$\begin{aligned}\overline{v}' &= (a_{\max} + \overline{a}_s(p) + \overline{a}_c(p)) + a_1 v + a_2 v^2\\ \texttt{stopDist}_t(v) &= vT + \left(\frac{a_{\max} + \overline{a}_s(p) + \overline{a}_c(p)}{2}\right)T^2 +\\ &\qquad \texttt{brakeDist}_a\Big(v + \big(\overline{v}'\big)T, 0\Big)\end{aligned} \tag{13}$$

Unlike previous stopping distance estimates, this expression is *not* always an upper bound. It uses resistance for the original velocity v_0, which is only a conservative bound when

[3]This is a lemma for the dL proof of Theorem V.3 justified as follows: consider two identical trains on tracks t_1 and t_2, starting with the same velocity. We want to bound the velocity v_1 of the train on t_1. Suppose t_2 is the track with worst case track and grade, and that a train on t_2 (the "ghost train", that we have constructed for the sake of our argument) always accelerates so that $v_2' = (a_{\max} + \overline{a}_s(p) + \overline{a}_c(p)) + a_1 v + a_2 v^2$. On the other hand, on track t_1, the real train that we require a proof about only obeys the restriction $|a_s| < m_s$. If $v_2 - v_1$ is to become negative, it must cross the boundary where its value is 0. However, whenever $v_1 = v_2$, necessarily, $v_2' > v_1'$. This ghost train argument serves a purpose similar to the "circular dependencies" argument of Section V-C: reasoning about mutually influencing factors one at a time. The ghost train permits us to represent and reason about a transcendental bound on velocity, v_2, derived using slope and curve estimates $\overline{a}_s$ and $\overline{a}_c$.

resistance is low enough to permit acceleration. This condition is captured by predicate vbound in Eq. (14).

$$\texttt{vbound}(v) \equiv (a_{\max} + \overline{a}_s(p) + \overline{a}_c(p)) + a_1 v + a_2 v^2 \geq 0 \quad (14)$$

For the Taylor approximation controller (Model 4), we define the $\texttt{stopDist}_{tp}$ predicate (15), that unlike previous expressions for stopping distance, returns a truth value. It uses previous definitions $\texttt{stopDist}_t$ from Eq. (13), vbound (14) to determine when $\texttt{stopDist}_t$ is applicable, and $\texttt{stopDist}_b$ from Eq. (8) is used as a fallback.

$$\begin{aligned}\texttt{stopDist}_{tp}(p, v, a_b) \equiv & e - p > \texttt{stopDist}_s(p, v, a_b) \vee \\ & \Big(\texttt{vbound}(v) \wedge e - p > \texttt{stopDist}_t(v)\Big)\end{aligned} \quad (15)$$

Higher order Taylor polynomials permit analogous reasoning. Theorem V.3 expresses that the *Taylor polynomial controller* in Model 4 is safe.

Model 4 Taylor polynomial controller and dynamics.

Reset Timer	1	$t := 0;$
	2	$\big((?\texttt{stopDist}_{tp}(p, v, 0);$
Free driving	3	$\quad a_l := *;\ ? - b_{\max} \leq a_l < a_{\max};$
	4	$\quad a_b := 0;\ m_b := 0)$
Braking	5	$\cup\ a_l := -b_{\max}; m_b := m_p\big);$
Dynamics	6	Model 3

Theorem V.3. *The Taylor polynomial controller guarantees that the train stays within its movement authority. The* dL *formula below is provable.*

$$\texttt{init}_s(\texttt{brakeDist}_a) \wedge \texttt{initAirbrake} \rightarrow [(\textit{Model 4})^*]\, e - p > 0$$

Proof. By proof in KeYmaera X. We start by showing that the loop invariant from Theorem V.2 is maintained (using dL rule loop). The case where the train is braking proceeds similar to Theorem V.2. When the train is accelerating, we need to show that the controller has insisted on a sufficient distance margin (stopping distance), so that even after a time period, the train has enough space to stop. As the Taylor polynomial computes the stopping distance, we must prove that it actually is an upper bound. We use a monotonicity argument by introducing an auxiliary variable that represents a "ghost" train, perpetually traveling down worst possible slope $\overline{a}_s$ and curve $\overline{a}_c$. This isolates slope and curve from the effect of resistance, breaking interdependence. We derive the Taylor polynomial result on the ghost train, show that it goes no slower than the real train, and that consequently the Taylor polynomial result must hold for the real train. Other elements of the proof remain similar to Theorem IV.1. □

VI. Kinematic Train Model Proofs

A proof for the FRA model [1] derives from the proof for our abstract mathematical models, e.g. Theorem V.3, by uniform substitution [7], which replaces abstract function symbols with specific terms using the correspondence in Section III-C. Model 5 lists the model resulting from substitution.

Model 5 Train controller in kinematic motion model.

Reset Timer	1	$t := 0;$
	2	$\big(\quad ?\texttt{stopDist}_{tp}(p, v, 0);$
Free Driving	3	$\qquad a_l := *;\ ? - b_{\max} \leq a_l < a_{\max};$
	4	$\qquad a_b := 0;\ m_b := 0$
Braking	5	$\quad \cup\ a_l := -b_{\max};\ m_b := m_p\big);$
	6	$(\quad \{\text{dyn } \&\ a_{b\max} \geq a_b\}$
	7	$\quad \cup \{\text{dyn } \&\ a_{b\max} \leq a_b$
Mode Split	8	$\qquad\qquad \wedge (a_{b\max} - a_b)/m_p \geq t_p\}$
	9	$\quad \cup \{\text{dyn } \&\ a_{b\max} \leq a_b$
	10	$\qquad\qquad \wedge (a_{b\max} - a_b)/m_p \leq t_p\})^*$

with dyn $\equiv \{p' = v, a_b' = m_b, t' = 1, v' = a_l +$

$$\max(a_b, a_{b\max}) - \frac{A_G w \cdot \textit{grade}(p)}{m_T} - \frac{A_C w \cdot \textit{curve}(p)}{m_T} - \frac{C_R w v + D_R v^2}{m_T}\ \&\ t \leq T \wedge v \geq 0\}$$

Corollary VI.1 (Kinematic train model is safe)**.** *The train controller for the FRA model never overshoots the end of movement authority, i.e., the following formula is provable with* $b_{max} = \frac{F_B}{m_T} + \frac{A_R w + B_R n}{m_T}$ *and* $a_{max} = \frac{F_L}{m_T} - \frac{A_R w + B_R n}{m_T}$*:*

$$\texttt{init}_s(\texttt{brakeDist}_a) \wedge \texttt{initAirbrake} \rightarrow [(\textit{Model 5})^*]\, e - p > 0\ .$$

Proof. By uniform substitution from Theorem V.3, using the substitutions σ below:

$$\sigma = \begin{cases} a_{\max} \mapsto \frac{F_L}{m_T} - \frac{A_R w + B_R n}{m_T} & b_{\max} \mapsto \frac{F_B}{m_T} + \frac{A_R w + B_R n}{m_T} \\ a_s(p) \mapsto -\frac{A_G w \cdot \textit{grade}(p)}{m_T} & a_c(p) \mapsto -\frac{A_C w \cdot \textit{curve}(p)}{m_T} \\ a_r(v) \mapsto -\frac{C_R w v + D_R v^2}{m_T} & \end{cases}$$

□

VII. Evaluation

For validation, we use ModelPlex [10] to derive a controller monitor from Model 5 that measures the safety margin in decisions of previous brake enforcement controllers [1], [2] and our verified control. That way, we measure if, and how well, our verified train controllers and existing controllers agree in order to assess the safety of those existing systems and the efficiency of our model. Existing brake enforcement controllers brake to a full stop once engaged.

The ModelPlex monitor computes a robustness measure indicating how close a decision is to losing the safety proof. When the robustness measure is positive, the decision is guaranteed to remain provably safe so that the system enjoys the safety proof of the verified model. When it is negative, emergency brakes should be applied for safety reasons. The ModelPlex controller monitor follows the structure of the verified model when it computes robustness. For example, a monitor for Lines 2–3 of Model 5 describes their effect with

TABLE I
FRA TRAIN PARAMETER INSTANTIATION EXAMPLE.

Param.	Value	Unit	Description
A_R	0.6	lbf/ton	(constant) weight coefficient
B_R	20	lbf/axle	(constant) train-size coefficient
C_R	0.01	$\text{lbf}/\text{ton}\cdot\text{mph}$	(linear) speed coefficient
D_R	0.07	lbf/mph^2	(quadratic) aerodynamic coefficient
	0.294		(for loaded cars)
A_G	20	lbf/ton	(constant) weight coefficient

the formula $\texttt{stopDist}_{tp}(p, v, 0) \wedge -b_{\max} \leq a_l < a_{\max}$ which translates to the robustness measure

$$\min\Big(\max\big(\min(e-p-\texttt{stopDist}_t(v),\\ (a_{\max}+\overline{a}_s(p)+\overline{a}_c(p))+a_1v+a_2v^2),\\ e-p-\texttt{stopDist}_a(p,v,0)\big),\\ a_l+b_{\max},\ a_{\max}-a_l\Big)\ .$$

The most important elements of the full Model 5 monitor are:

- in free driving (when $\texttt{stopDist}_{tp}$ is satisfied) it combines remaining position margin (the larger of Taylor margin $e-p-\texttt{stopDist}_t$ when $\texttt{vbound}(v)$, or fallback margin $e-p-\texttt{stopDist}_a$) with acceleration choice robustness ($\min(a_l+b_{\max}, a_{\max}-a_l)$ from control decision $a := *; ?-b_{\max} \leq a_l < a_{\max}$) and speed robustness ($v$ from evolution domain $\ldots \& v \geq 0$);
- during braking, which is always allowed, it measures speed robustness (v per evolution domain constraint).

Because ModelPlex's robustness measure combines multiple quantities of incompatible units, there is no direct interpretation of its magnitude, but only of its sign.

For validation, we implement the train model of Eq. (2) in Python by numerical integration, instantiating the model parameters per FRA model [1], [2]. These parameter values are estimated from train test runs and standards [2] and require careful consideration of their units (Table I).

Our evaluation compares start braking and stopping points of trains, highlighting braking performance in terms of overshoot (safety risk) and undershoot (performance objective of a maximum undershoot of 1000ft [1]) of the end of movement authority. We follow [2, p. 47] and implement the baseline controllers using numeric forward Euler integration to simulate the model in order to determine the stopping distance. Our verified controllers neither use numeric integration nor include the dynamic model, but instead decide based on the stopping distance overapproximation $\texttt{stopDist}_{tp}$ of Eq. (15).

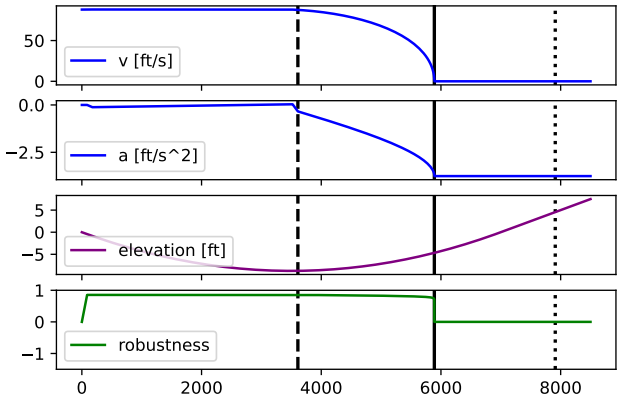
The most interesting train behavior arises from the subtle interplay between air pressure propagation, aerodynamic/roll resistance, and acceleration/deceleration due to slope. It peaks on crests that change gradient from uphill to downhill and in troughs that change gradient from downhill to uphill. When calculating stopping distance, numerical integration in the baseline enforcement algorithms discretizes train speed and position to calculate forces, which overestimates resistance while simultaneously underestimating available brake force. Acceleration/deceleration due to slope is even more subtle as it depends on the position of the train along the slope (e.g., on a crest changing from uphill to downhill, deceleration on the uphill segment is overestimated until the train passes the top, afterwards acceleration is underestimated). These effects do *not* balance out and thus make numerical integration errors unreliable and hard to predict. Moreover, changing the integration step size shifts how distance estimates are biased towards undershoot or overshoot (e.g., in typical configurations, brake rampup is the dominating influence on stopping distance, and so larger integration step sizes bias towards undershoot). As a result, for any given configuration of numerical integration in enforcement algorithms, we can construct scenarios where the numerical integration underestimates stopping distance and train enforcement exhibits unsafe behavior. Our formal models and proofs design *provably correct* stopping distance overapproximations instead of using numerical integration and are, therefore, not subject to these intricate safety tradeoffs.

A. Stopping Behavior in Troughs

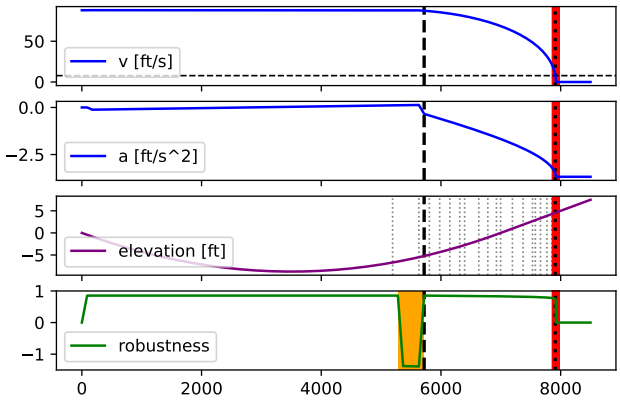
Our first validation in Fig. 2 follows [1, Fig. 10] with a train configuration of a medium (75 cars), loaded (car weight 286klbf), mixed freight train traveling at initial speed 60mph in a trough, with train starting position at 0ft. The trough is configured with uniform rate of change from 0.5% downhill to -0.5% uphill between positions 0ft and 7000ft. Locomotive tractive effort and locomotive braking compensates for speed loss and gain due to gradient. Air brakes engage only for the full braking maneuver when attempting to not exceed the movement authority. We configure our monitors and controllers to match the trough maximum uphill/downhill gradient of 0.5% for slope estimation. A sweep of movement authority endpoints in the range 4000ft–8000ft with 10ft steps identifies challenging configurations. The base enforcement algorithm [1] underestimates stopping distance for 19 (of the 400) endpoints. Fig. 2 compares the base enforcement algorithm to our verified controller on one of these challenging points at 7910ft, past the maximum uphill slope of the trough. The base enforcement algorithm in Fig. 2a uses numeric integration to determine the stopping distance, and adds a generous constant fudge factor plus speed-dependent safety offset that results in the train initiating braking at 3609ft. The maneuver finishes at 5889ft, stopping 2021ft short of the end of movement authority. In Fig. 2c, our verified controller does *not* use any such offsets. It initiates braking much later at position 5368ft and stops at 7578ft (significantly closer to the movement authority endpoint, limiting undershoot to 332ft).

Fig. 2b, removes the safety offset from the base enforcement algorithm in an experiment illustrating the subtle interplay between forces and their safety consequences. Even though the uphill segment of the trough helps reduce stopping distance, numeric integration overestimates this effect and initiates braking too late, which our monitor detects at 5368ft. Should the train ignore the warning, it overshoots by 9ft (highlighted in red[4]) with a remaining speed of $7.8\frac{\text{ft}}{\text{s}}$ when passing the desired

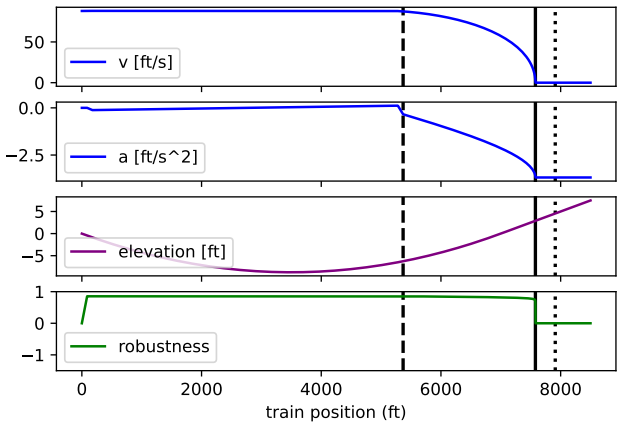
[4]The controller monitor subsequently no longer flags a violation, because the base enforcement algorithm then applies the correct decision of maximum braking (agreeing with the model). But it does so too late, as the earlier monitor warning was ignored.



(a) Base enforcement algorithm [1] initiates braking at 3609ft and stops at position 5889ft with an undershoot of 2021ft, which exceeds the 1000ft maximum undershoot performance objective [1].



(b) Base enforcement algorithm [1] without offset overshoots by 9ft ϟ. Light-gray dotted lines mark other challenging movement authority endpoints in the range 4000ft–8000ft that would also be violated.



(c) Verified controller (Model 5) limits undershoot to 332ft.

Fig. 2. Comparison of start braking (dashed vertical line) and stopping points (solid vertical lines) in a trough from 0.5% downhill at start to −0.5% uphill at 7000ft, with end of movement authority (dotted vertical line) at 7910ft.

stopping point. Initiating fallback control $a_l = -b_{\max}$ at the monitor violation would have kept the train from overshooting the movement authority.

B. Stopping Behavior on Crests

An assumption in the brake enforcement algorithms and thus an initial condition in our proof is that train locomotives are not underpowered: their tractive effort is enough to overcome maximum uphill slope and stay stopped on the maximum downhill slope. Fig. 3 illustrates how an underpowered, initially stopped train starts rolling downhill until the air brakes build enough deceleration to stop the train. This configuration, violating initial conditions, is unsafe to start in the first place.

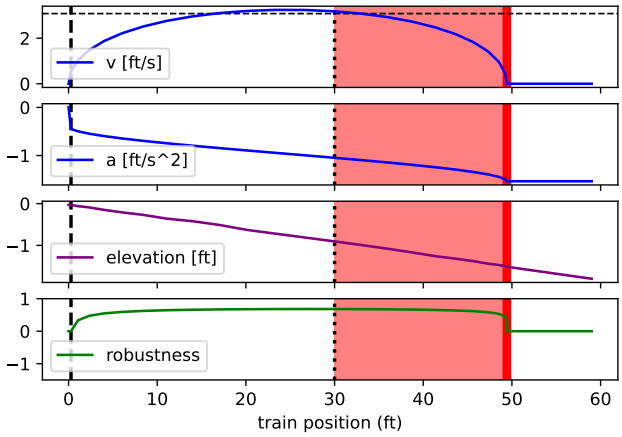


Fig. 3. Initial conditions of proof (Corollary VI.1) not satisfied ϟ: an underpowered train on a 3% downhill slope needs air brakes to stay stopped; the stopping point is closer than the rolling distance during brake rampup.

Underpowered locomotives are especially challenging on a crest where (full) tractive effort is needed to limit the speed loss on the uphill slope and regain desired speed on the downhill segment, but air brakes are needed to stay stopped. Fig. 4 compares the behavior of the base enforcement algorithms and our verified algorithm with underpowered locomotives on a crest with movement authority endpoint at 7920ft. In this experiment, we configure our monitors and controllers for a maximum uphill/downhill gradient of 3% for slope estimation.

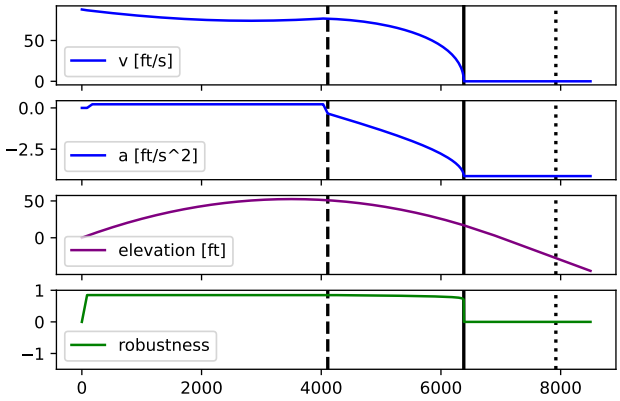
Numeric integration underestimates stopping distance in several configurations, whereas our verified controller correctly identifies the need to engage air brakes in time while simultaneously avoiding the inefficiencies of fudge factors.

VIII. Conclusion

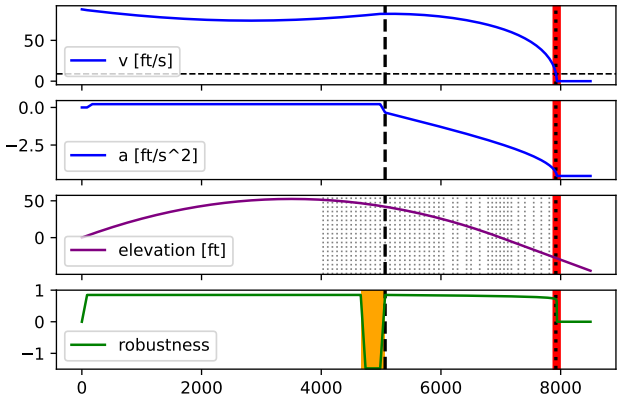
We created formally verified train controllers that account for the FRA model with all its competing influences of track grade, curve resistance, air brakes and Davis resistance. Techniques that generalize to resolve challenges in safety critical embedded software design improved controller efficiency. Validation in simulation shows significant improvement in undershoot over conservative controllers that use safety offsets, and improved safety compared to controllers without safety offsets. In future work, VeriPhy [33], a verified pipeline that automatically converts verified dL models to verified executables can bridge the gap between the real arithmetic verification of the paper and floating points used by software.

References

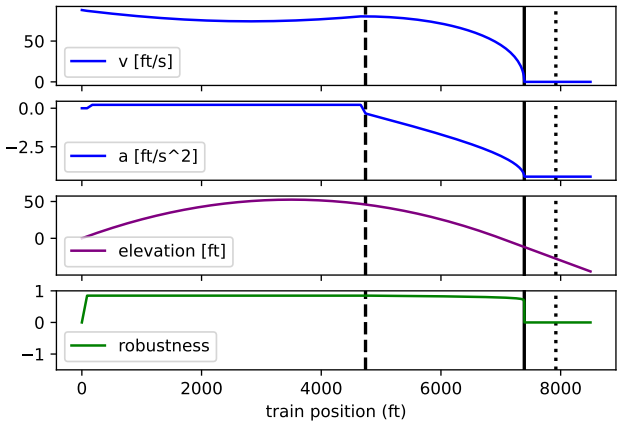
[1] J. Brosseau and B. M. Ede, "Development of an adaptive predictive braking enforcement algorithm," Federal Railroad Administration, Tech. Rep. FRA/DOT/ORD-9/13, 2009.



(a) Base enforcement algorithm [1] undershoots by 1539ft.



(b) Base enforcement algorithm [1] without offset overshoots by 14ft ϟ. Light-gray dotted lines mark other movement authority endpoints in the range 4000ft–8000ft that the algorithm would also overshoot.



(c) Verified controller (Model 5) limits undershoot to 528ft.

Fig. 4. Initial conditions of proof (Corollary VI.1) not satisfied ϟ: an underpowered train on a crest from 3% uphill to −3% downhill slows down uphill despite full tractive effort and regains speed on the downhill segment, but needs air brakes to stay stopped; in this scenario, the movement authority ends past the rolling distance during brake rampup.

[2] J. Brosseau, B. M. Ede, S. Pate, R. Wiley, and J. Drapa, "Development of an operationally efficient PTC braking enforcement algorithm for freight trains," Tech. Rep. DOT/FRA/ORD-13/34, 2013.

[3] A. Platzer and J.-D. Quesel, "European Train Control System: A case study in formal verification," in *ICFEM*, 2009, pp. 246–265.

[4] L. Zou, J. Lv, S. Wang, N. Zhan, T. Tang, L. Yuan, and Y. Liu, "Verifying chinese train control system under a combined scenario by theorem proving," in *Verified Software: Theories, Tools, Experiments*, 2014, pp. 262–280.

[5] S. Mitsch, M. Gario, C. J. Budnik, M. Golm, and A. Platzer, "Formal verification of train control with air pressure brakes," in *RSSRail*, 2017, pp. 173–191.

[6] S. M. Loos and A. Platzer, "Differential refinement logic," in *LICS*, 2016, pp. 505–514.

[7] A. Platzer, "A complete uniform substitution calculus for differential dynamic logic," *J. Autom. Reas.*, vol. 59, no. 2, pp. 219–265, 2017.

[8] ——, "Differential dynamic logic for hybrid systems." *J. Autom. Reas.*, vol. 41, no. 2, pp. 143–189, 2008.

[9] N. Fulton, S. Mitsch, J.-D. Quesel, M. Völp, and A. Platzer, "KeYmaera X: An axiomatic tactical theorem prover for hybrid systems," in *CADE*, 2015, pp. 527–538.

[10] S. Mitsch and A. Platzer, "ModelPlex: Verified runtime validation of verified cyber-physical system models," *Form. Methods Syst. Des.*, vol. 49, no. 1-2, pp. 33–74, 2016.

[11] A. Ferrari, M. H. ter Beek, F. Mazzanti, D. Basile, A. Fantechi, S. Gnesi, A. Piattino, and D. Trentini, "Survey on formal methods and tools in railways: The astrail approach," in *RSSRail*, 2019.

[12] J. Eckert, Í. Teodoro, L. da Silva Teixeira, T. Martins, P. Kurka, and A. Santos, "A fast simulation approach to assess draft gear loads for heavy haul trains during braking," *Mechanics Based Design of Structures and Machines*, 2021.

[13] Í. P. Teodoro, J. J. Eckert, P. F. Lopes, T. S. Martins, and A. A. Santos, "Parallel simulation of railway pneumatic brake using openMP," *International Journal of Rail Transportation*, vol. 8, no. 2, pp. 180–194, 04 2020.

[14] C. Budnik, M. Gario, G. Markov, and Z. Wang, "Guided test case generation through AI enabled output space exploration," 05 2018, pp. 53–56.

[15] U. Berger, P. James, A. Lawrence, M. Roggenbach, and M. Seisenberger, "Verification of the European Rail Traffic Management System in Real-Time Maude," *Science of Computer Programming*, vol. 154, pp. 61–88, 2018.

[16] A. Fantechi, "Connected or autonomous trains?" in *RSSRail*, 2019, pp. 3–19.

[17] M. D. Claudio, A. Fantechi, G. Martelli, S. Menabeni, and P. Nesi, "Model-based development of an automatic train operation component for communication based train control," in *ITSC*, 2014, pp. 1015–1020.

[18] S. L. Gerhart, D. Craigen, and T. Ralston, "Case study: Paris metro signaling system," *IEEE Software*, vol. 11, pp. 32–28, 1994.

[19] D. Essamé and D. Dollé, "B in large-scale projects: The canarsie line cbtc experience," 12 2006, pp. 252–254.

[20] L. H. Vu, A. E. Haxthausen, and J. Peleska, "Formal modelling and verification of interlocking systems featuring sequential release," *Science of Computer Programming*, vol. 133, pp. 91–115, 2017, formal Techniques for Safety-Critical Systems.

[21] A. Cimatti, R. Corvino, A. Lazzaro, I. Narasamdya, T. Rizzo, M. Roveri, A. Sanseviero, and A. Tchaltsev, "Formal verification and validation of ERTMS industrial railway train spacing system," in *CAV*, 2012, pp. 378–393.

[22] A. Bonacchi and A. Fantechi, "On the validation of an interlocking system by model-checking," 09 2014, pp. 94–108.

[23] E. Kamburjan and R. Hähnle, "Deductive verification of railway operations," in *RSSRail*, 2017, pp. 131–147.

[24] S. L. Karra, K. G. Larsen, F. Lorber, and J. Srba, "Safe and time-optimal control for railway games," in *RSSRail*, 2019, pp. 106–122.

[25] M. Wahl, *Survey of railway embedded network solutions - Towards the use of Industrial Ethernet technologies. In: Les Collections de l'Inrets, Synthése S61, 104 pages*, 2010.

[26] A. Platzer, *Logical Analysis of Hybrid Systems: Proving Theorems for Complex Dynamics*. Heidelberg: Springer, 2010.

[27] ——, *Logical Foundations of Cyber-Physical Systems*. Springer, 2018.

[28] W. W. Hay, *Railroad engineering*, 2nd ed. New York: Wiley, 1982.

[29] J. Dasher and K. Morrison, "Evaluation of ptc braking enforcement algorithms for passenger and commuter trains," Federal Railroad Administration, Tech. Rep., 2020.

[30] D. Richardson, "Some undecidable problems involving elementary functions of a real variable," *J. Symb. Log.*, vol. 33, no. 4, pp. 514–520, 1968.

[31] J. H. Davenport and J. Heintz, "Real quantifier elimination is doubly exponential." *J. Symb. Comput.*, vol. 5, no. 1/2, pp. 29–35, 1988.

[32] V. Weispfenning, "The complexity of linear problems in fields," *J. Symb. Comput.*, vol. 5, no. 1-2, pp. 3–27, 1988.

[33] B. Bohrer, Y. K. Tan, S. Mitsch, M. O. Myreen, and A. Platzer, "VeriPhy: Verified controller executables from verified cyber-physical system models," in *PLDI*, 2018, pp. 617–630.