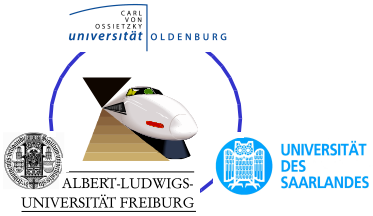


AVACS – Automatic Verification and Analysis of Complex Systems

REPORTS

of SFB/TR 14 AVACS

Editors: Board of SFB/TR 14 AVACS

European Train Control System: A Case Study in Formal Verification

by

André Platzer Jan-David Quesel

AVACS Technical Report No. 54
October 2009
ISSN: 1860-9821

Publisher: Sonderforschungsbereich/Transregio 14 AVACS
(Automatic Verification and Analysis of Complex Systems)
Editors: Bernd Becker, Werner Damm, Martin Fränzle, Ernst-Rüdiger Olderog,
Andreas Podelski, Reinhard Wilhelm
ATRs (AVACS Technical Reports) are freely downloadable from `www.avacs.org`

Author(s) contact: Jan-David Quesel (`Quesel@informatik.uni-oldenburg.de`).

European Train Control System: A Case Study in Formal Verification*

André Platzer[1] and Jan-David Quesel[2]

[1] Computer Science Department, Carnegie Mellon University, Pittsburgh, PA
aplatzer@cs.cmu.edu
[2] University of Oldenburg, Department of Computing Science, Germany
quesel@informatik.uni-oldenburg.de

Abstract. Complex physical systems have several degrees of freedom. They only work correctly when their control parameters obey corresponding constraints. Based on the informal specification of the *European Train Control System* (ETCS), we design a controller for its cooperation protocol. For its free parameters, we successively identify constraints that are required to ensure collision freedom. We formally prove the parameter constraints to be sharp by characterizing them equivalently in terms of reachability properties of the hybrid system dynamics. Using our deductive verification tool KeYmaera, we formally verify controllability, safety, liveness, and reactivity properties of the ETCS protocol that entail collision freedom. We prove that the ETCS protocol remains correct even in the presence of perturbation by disturbances in the dynamics. We verify that safety is preserved when a PI controlled speed supervision is used.

Keywords: formal verification of hybrid systems, train control, theorem proving, parameter constraint identification, disturbances

1 Introduction

Complex physical control systems often contain many degrees of freedom including how specific parameters are instantiated or adjusted [1–3]. Yet, virtually all of these systems are hybrid systems [4] and only work correctly under certain constraints on these parameters. The *European Train Control System* (ETCS) [5] has a wide range of different possible configurations of trains, track layouts, and different driving circumstances. It is only safe for certain conditions on external parameters, e.g., as long as each train is able to avoid collisions by braking with its specific braking power on the remaining distance to the rear end of the next train. Similarly, internal control design parameters for supervisory speed control and automatic braking triggers need to be adjusted in accordance with the underlying train dynamics. Moreover, parameters must be constrained such that

* *All propositions have been verified in KeYmaera!* This work was partially supported by the German Research Council (DFG) as part of the Transregional Collaborative Research Center "Automatic Verification and Analysis of Complex Systems" (SFB/TR 14 AVACS), and by the National Science Foundation (NSF) under grants no. CNS-0931985 and CCF-0926181.

the system remains correct when passing from continuous models with instant reactions to sampled data discrete time controllers of hardware implementations. Finally, parameter choices must preserve correctness robustly in the presence of disturbances caused by unforeseen external forces (wind, friction, ...) or internal modelling inaccuracies of ideal-world dynamics, e.g., when passing from ideal-world dynamics to *proportional-integral* (PI) controller implementations.[3] Yet, determining the range of external parameters and the choice of internal design parameters for which complex control systems like ETCS are safe, is not possible just by looking at the model, even less so in the presence of disturbance.

Likewise, it is difficult to read off the parameter constraints that are required for correctness from a failed verification attempt of model checkers [6–8], since concrete numeric values of a counterexample trace cannot simply be translated into a generic constraint on the free parameters of the system which would have prevented this kind of error. While approaches like counterexample-guided abstraction refinement [9, 8] are highly efficient in undoing automatic abstractions of an abstract hybrid system from spurious counterexamples, they stop when true counterexamples remain in the concrete system. For discovering constraints on free parameters, though, even concrete models will have counterexamples until all required parameter constraints have been identified.

Instead, we use our techniques based on symbolic decompositions [10] for systematically exploring the design space of a hybrid system and for discovering correctness constraints on free parameters. For a complex physical system, we show step by step how a control system can be developed that meets its control design goals and desired correctness properties. Starting from a coarse skeleton of the ETCS cooperation protocol obtained from its official specification [5], we systematically develop a safe controller and identify the parameter constraints that are required for collision freedom. Although these parameter constraints are safety-critical, they are not stated in the official specification [5]. Rather, they result from the system dynamics and objectives and need to be made explicit to find safe choices. The constraints are nontrivial especially those needed to ensure a safe interplay of physics and sampled control implementations. Using the parametric constraints so discovered, we verify correctness properties of the ETCS cooperation protocol that entail collision freedom. We verify rich properties, including safety, controllability, reactivity, and liveness, which are not uniformly expressible and verifiable in most other approaches. Moreover, we verify those correctness properties of the parametric ETCS case study almost fully automatically in our verification tool KeYmaera [11]. These results are a significant extension of our preliminary short report [12] where we have only shown one property. Now we verify 13 properties of ETCS and verify PI control and disturbance extensions.

Contributions We show how realistic fully parametric hybrid systems for traffic protocols can be designed and verified using a logic-based approach. For ETCS, we identify all relevant safety constraints on free parameters, including external

[3] PI is a standard control technique and also used for controlling trains [2].

system parameters and internal design parameters of controllers. Safe control choices will be important for more than two million passengers in Europe per day. Our first contribution is that we characterize safe parameter choices equivalently in terms of properties of the train dynamics and that we verify controllability, reactivity, safety, and liveness properties of ETCS. Our second contribution is that we show how to verify ETCS with a proportional-integral (PI) controller. In contrast to their routine use in control, giving formal proofs for the correct functioning of PIs has been an essentially unsolved problem. Other issues often arise from verification results for ideal-world dynamics that cease to hold for real-world dynamics. Our third contribution is to show how to extend ETCS verification to the presence of disturbances in the dynamics, which account for friction etc. Most notably, the ETCS model with its rich set of properties is out of scope for other approaches. ETCS further illustrates a more general phenomenon in hybrid systems: safely combining dynamics with control requires parameter constraints that are much more complicated than the original dynamics.

Related Work Model checkers for hybrid systems, for example HyTech [4] and PHAVer [8], verify by exploring the state space of the system as exhaustively as possible. In contrast to our approach they need concrete numbers for most parameters and cannot verify liveness or existential properties, e.g., whether and how a control parameter can be instantiated so that the system is always safe.

Other approaches [13, 7] use quantifier elimination for model checking hybrid automata. We use the same decision procedure for handling arithmetics and generalize these results adding free parameter constraint handling and discovery. In [7] real quantifier elimination is used for model checking TCTL properties of semi-algebraic hybrid systems. However, they use approximated solutions of the differential equations, whereas our approach avoids those approximation errors.

Batt et al. [3] give heuristics for splitting regions by linear constraints that can be used to determine parameter constraints. Frehse et al. [14] synthesize parameters for linear hybrid automata. However, realistic systems like ETCS require non-linear parameter constraints and are out of scope for these approaches.

Tomlin et al. [15] show a game-theoretic semi-decision algorithm for hybrid controller synthesis. For systems like ETCS, which are more general than linear or o-minimal hybrid automata, they suggest numerical approximations. We give exact results for fully parametric ETCS using symbolic techniques.

Peleska et al. [16] and Meyer et al. [1] verify properties of trains. They do not verify hybrid dynamics or the actual movement of trains. The physical dynamics is crucial for faithful train models and for showing actual collision freedom, because, after all, collision freedom is a property of controlled movement.

Cimatti et al. [17] analyze consistency of informal requirements on ETCS expressed as temporal properties including the continuous system dynamics using an approach based on the combination of temporal logic with regular expressions. Our work is complementary, as we focus on developing and verifying an actual hybrid systems controller that can be implemented later on, not the consistency of the requirement specification properties.

Davoren and Nerode [18, 19] extended the propositional modal μ-calculus with a semantics in hybrid systems and examine topological aspects. Propositional modalities are built from propositional actions A, B of unknown effect. We improve on this by generalizing unknown propositional actions to concrete first-order actions like $\tau.p'' = -b$ or $\tau.a := *; ? - b \leq \tau.a < 0$ with known effect, which we need to express concrete operational ETCS models.

In [20] Davoren and Tabuada study different bisimulations of hybrid program. The p-bisimulation characterized in that paper preserves GFL* semantics. They do not, however, construct bisimulations of hybrid systems where reachability can be computed. Davoren and Tabuada [20] present a notion of bisimulation that preserves the semantics of their logic GFL*. They do not yet give a construction for obtaining bisimulations of hybrid systems where reachability is sufficiently computational.

Other theorem provers for verifying hybrid systems integrate arithmetic reasoning into STeP [21] or PVS [22]. They compile a hybrid automaton and a given system invariant to a global verification condition. In contrast, with our logic $\mathsf{d}\mathcal{L}$ for hybrid systems, we can decompose the system successively and exploit the so preserved structure for deriving invariants and parametric constraints. In contrast to our work they compile a given hybrid automaton [6] and a global system invariant in a single step into a verification condition expressing that the invariant is preserved under all transitions of the hybrid automaton. This condition is a quantified mathematical formula, thus the hybrid aspects and transition structure vanish completely before the deduction starts. As presented in this paper, our dynamic logic works by symbolic decomposition and preserves the transition structure during the proof, which simplifies traceability of results considerably. The structure in this symbolic decomposition can be exploited for deriving invariants or parametric constraints as shown in Sect. 4. Consequently, in $\mathsf{d}\mathcal{L}$, invariants do not necessarily need to be given beforehand.

Structure of this Paper In Sect. 2 we summarize differential dynamic logic [10] which we use for modelling ETCS. We introduce a formal model for parametric ETCS in Sect. 3. Using symbolic decomposition analysis [10], we systematically derive parametric correctness constraints for ETCS and verify several correctness properties of parametric ETCS using these constraints in Sect. 4. More complex control models, namely PI controllers are the topic of Sect. 5. In Sect. 6, we generalize the physical transmission model to the presence of disturbances and verify ETCS with disturbances. Section 7 gives experimental results in our verification tool KeYmaera. We summarize in Sect. 8. Further properties and proofs are given in the Appendix.

2 Preliminaries: Differential Dynamic Logic

In this section, we survey *differential dynamic logic* $\mathsf{d}\mathcal{L}$ [10] which is tailored for specifying and verifying rich correctness properties of parametric hybrid systems.

Both its ability to express rich properties and the structural decomposition techniques for $\mathsf{d}\mathcal{L}$ are highly beneficial for expressing and discovering the required parameter constraints for ETCS. We only develop the theory as far as necessary and refer to [10] for more background on $\mathsf{d}\mathcal{L}$ and the sequent proof calculus for $\mathsf{d}\mathcal{L}$ which is implemented in KeYmaera [11].

The logic $\mathsf{d}\mathcal{L}$ is a first-order logic with built-in correctness statements about hybrid systems. It is designed such that parametric verification analysis can be carried out directly in $\mathsf{d}\mathcal{L}$. Generalizing the principle of dynamic logic to the hybrid case, $\mathsf{d}\mathcal{L}$ combines hybrid system operations and correctness statements about system states within a single specification and verification language. $\mathsf{d}\mathcal{L}$ uses *hybrid programs* (HP) [10] as a program notation for hybrid systems that is amenable to deductive structural decomposition in $\mathsf{d}\mathcal{L}$. In addition to standard operations of discrete programs, HPs have continuous evolution along differential equations as a basic operation. For example, the movement of a train braking with force b can be expressed by placing the differential equation $\tau.p'' = -b$ (where $\tau.p''$ is the second time-derivative of $\tau.p$) at the appropriate point inside a HP. Together with the change of variable domain from $\mathbb{N}$ to $\mathbb{R}$, differential equations constitute a crucial generalization from discrete dynamic logic to $\mathsf{d}\mathcal{L}$.

The syntax of hybrid programs is shown together with an informal semantics in Tab. 1. The basic terms (called θ in the table) are either real numbers, real-valued variables or arithmetic expressions built from those.

The effect of $x := \theta$ is an instantaneous discrete jump assigning θ to x. That of $x' = \theta \wedge \chi$ is an ongoing continuous evolution controlled by the differential equation $x' = \theta$ while remaining within the evolution domain χ. The evolution is allowed to stop at any point in χ but it must not leave χ. For unrestricted evolution, we write $x' = \theta$ for $x' = \theta \wedge \mathit{true}$. Systems of differential equations and higher-order derivatives are defined accordingly: $\tau.p' = v \wedge \tau.v' = -b \wedge \tau.v \geq 0$, for instance, characterizes the braking mode of a train with braking force b that holds within $\tau.v \geq 0$ and stops at speed $\tau.v \leq 0$ at the latest.

The test action $?\chi$ is used to define conditions. It completes without changing the state if χ is true in the current state, and it aborts all further evo-

Table 1: Statements of hybrid programs (F is a first-order formula, α, β are HPs)

Statement	Effect
$\alpha;\ \beta$	sequential composition, first performs α and then β afterwards
$\alpha\ \cup\ \beta$	nondeterministic choice, following either α or β
α^*	nondeterministic repetition, repeating α some $n \geq 0$ times
$x := \theta$	discrete assignment of the value of term θ to variable x (jump)
$x := *$	nondeterministic assignment of an arbitrary real number to x
$\big(x_1' \sim_1 \theta_1 \wedge \cdots \wedge x_n' \sim_n \theta_n \wedge F\big)$	continuous evolution of x_i along differential (in)equation system $x_i' \sim_i \theta_i$, with $\sim_i \in \{\leq, =\}$, restricted to evolution domain F
$?F$	check if formula F holds at current state, abort otherwise
if(F) then α	perform α if F is true, do nothing otherwise
if(F) then α else β	perform α if F is true, perform β otherwise

lution, otherwise. The nondeterministic choice $\alpha \cup \beta$ expresses alternatives in the behavior of the hybrid system. The if-statement can be expressed using the test action and the choice operator. Its semantics is that if the condition is true, the then-part is executed, otherwise the else-part is performed, if there is one, otherwise the statement is just skipped. The sequential composition $\alpha;\beta$ expresses that β starts after α finishes. Nondeterministic repetition α^* says that the hybrid program α repeats an arbitrary number of times. These operations can be combined to form any other control structure. For instance, $(?\tau.v \geq m.r;\tau.a := A) \cup (?\tau.v \leq m.r;\tau.a := -b)$ says that, depending on the relation of the current speed $\tau.v$ of some train and its recommended speed $m.r$, $\tau.a$ is chosen to be the maximum acceleration A if $m.e - \tau.p \geq 0$ or maximum deceleration $-b$ if $m.e - \tau.p \leq 0$. If both conditions are true (hence, $m.e - \tau.p = 0$) the system chooses either way. The random assignment $x := *$ nondeterministically assigns any value to x, thereby expressing unbounded nondeterminism, e.g., in choices for controller reactions. For instance, the idiom $\tau.a := *; ?\tau.a > 0$ randomly assigns any positive value to the acceleration $\tau.a$.

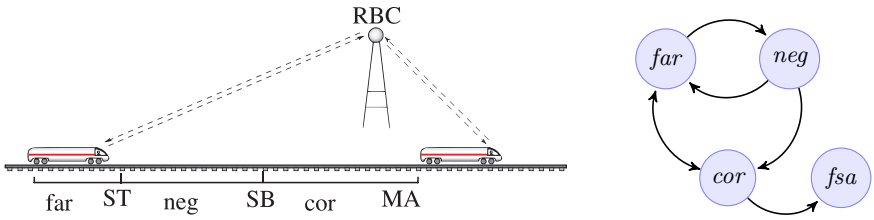
The $\mathsf{d}\mathcal{L}$*-formulas* are defined by the following grammar (θ_i are terms, x is a real-valued variable, $\sim \in \{<, \leq, =, \geq, >\}$ is one of the usual real predicates, ϕ and ψ are formulas, α is a HP):

$$\theta_1 \sim \theta_2 \mid \neg\phi \mid \phi \wedge \psi \mid \phi \vee \psi \mid \phi \rightarrow \psi \mid \phi \leftrightarrow \psi \mid \forall x\, \phi \mid \exists x\, \phi \mid [\alpha]\phi \mid \langle\alpha\rangle\phi$$

The formulas are designed as an extension of first-order logic over the reals with built-in correctness statements about HPs. They can contain propositional connectives $\wedge, \vee, \rightarrow, \leftrightarrow, \neg$ and real-valued quantifiers $\forall, \exists$ for quantifying over parameters and evolution times. For HP α, $\mathsf{d}\mathcal{L}$ provides correctness statements like $[\alpha]\phi$ and $\langle\alpha\rangle\phi$, where $[\alpha]\phi$ expresses that all traces of system α lead to states in which condition ϕ holds. Likewise, $\langle\alpha\rangle\phi$ expresses that there is at least one trace of α to a state satisfying ϕ. As $\mathsf{d}\mathcal{L}$ is closed under logical connectives, it provides conditional correctness statements like $\phi \rightarrow [\alpha]\psi$, saying that α satisfies ψ if condition ϕ holds at the initial state, or even nested statements like the reactivity statement $[\alpha]\langle\beta\rangle\phi$, saying that whatever HP α is doing, HP β can react in some way to ensure ϕ. As a closed logic, $\mathsf{d}\mathcal{L}$ can also express mixed quantified statements like $\exists m\, [\alpha]\phi$ saying that there is a choice of parameter m such that system α always satisfies ϕ, which is useful for determining parameter constraints.

3 Parametric European Train Control System

The European Train Control System (ETCS) [5, 1] is a standard to ensure safe and collision-free operation as well as high throughput of trains at speeds up to 320km/h. Correct functioning of ETCS is highly safety-critical, because the upcoming installation of ETCS level 3 will replace all previous track-side safety measures in order to achieve its high throughput objectives. In this section, we present a system skeleton, which corresponds to a simple representation of the



1a: Dynamic assignment of movement authorities 1b: Cooperation pattern

Fig. 1: ETCS train cooperation protocol

train dynamics and controller reflecting the informal ETCS cooperation protocol [5]. This system is actually unsafe. In Sect. 4, we will systematically augment this skeleton with the parameter constraints that are required for safety but not stated in the informal specification [5].

3.1 Overview of the ETCS Cooperation Protocol

ETCS level 3 follows the *moving block principle*, i.e., movement permissions are neither known beforehand nor fixed statically. They are determined based on the current track situation by a *Radio Block Controller* (RBC). Trains are only allowed to move within their current *movement authority* (MA), which can be updated by the RBC using wireless communication. Hence the train controller needs to regulate the movement of a train locally such that it always remains within its MA. After MA, there could be open gates, other trains, or speed restrictions due to tunnels. The automatic train protection unit (*atp*) dynamically determines a safety envelope around a train τ, within which it considers driving safe, and adjusts the train acceleration $\tau.a$ accordingly. Fig. 1a illustrates the dynamic assignment of MA. The ETCS controller switches according to the protocol pattern in Fig. 1b which corresponds to a simplified version of Damm et al. [2]. When approaching the end of its MA the train switches from *far* mode (where speed can be regulated freely) to negotiation (*neg*), which, at the latest, happens at the point indicated by *ST* (for *start talking*). During negotiation the RBC grants or denies MA-extensions. If the extension is not granted in time, the train starts braking in the correcting mode (*cor*) returning to *far* afterwards. Emergency messages announced by the RBC can also put the controller into *cor* mode. If so, the train switches to a failsafe state (*fsa*) after the train has come to a full stop and awaits manual clearance by the train operator.

Lemma 1 (Principle of separation by movement authorities). *If each train stays within its MA and, at any time, MAs issued by the RBC form a disjoint partitioning of the track, then trains can never collide (see Appendix B).*

Lemma 1 effectively reduces the verification of an unbounded number of traffic agents to a finite number. We exploit MAs to decouple reasoning about global

collision freedom to local cooperation of every traffic agent with its RBC. In particular, we verify correct coordination for a train without having to consider gates or railway switches, because these only communicate via RBC mediation and can be considered as special reasons for denial of MA-extensions. We only need to prove that the RBC handles all interaction between the trains by assigning or revoking MA correctly and that the trains respect their MA. However, to enable the RBC to guarantee disjoint partitioning of the track it has to rely on properties like appropriate safe rear end computation of the train. Additionally, safe operation of the train plant in conjunction with its environment depends on proper functioning of the gates. As these properties have a more static nature, they are much easier to show once the actual hybrid train dynamics and movements have been proven to be controlled correctly.

As trains are not allowed to drive backwards without clearance by track supervision personnel, the relevant part of the safety envelope is the closest distance to the end of its current MA. The point SB, for *start braking*, is the latest point where the train needs to start correcting its acceleration (in mode *cor*) to make sure it always stays within the bounds of its MA. In Sect. 4, we derive a necessary and sufficient constraint on SB that guarantees safe driving.

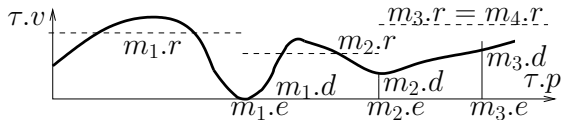


Fig. 2: ETCS track profile

We generalize the concept of MA to a vector $m = (d, e, r)$ meaning that beyond point $m.e$ the train must not have a velocity greater than $m.d$. Additionally, the train should try not to outspeed the *recommended speed* $m.r$ for the current track segment. Short periods of slightly higher speed are not considered safety-critical (They will occur during PI speed supervision). Fig. 2 shows an example of possible train behavior in conjunction with the current value of m that changes over time due to RBC communication.

For a train $\tau = (p, v, a)$ at position $\tau.p$ with current velocity $\tau.v$ and acceleration $\tau.a$, we want to determine sufficient conditions ensuring safety and formally verify that $\tau.v$ is always safe with respect to its current MA, thus satisfying:

$$\tau.p \geq m.e \;\rightarrow\; \tau.v \leq m.d \tag{$\mathcal{S}$}$$

Formula ($\mathcal{S}$) expresses that the train's velocity $\tau.v$ does not exceed the strict speed limit $m.d$ after passing the point $m.e$ (i.e., $\tau.p \geq m.e$). Generalized MA are a uniform composition of two safety-critical features. They are crucial aspects for ensuring collision free operation in ETCS (Lemma 1) and can take into account safety-critical velocity limits due to bridges, tunnels, or passing trains. For example high speed trains need to reduce their velocity while passing non-airtight or

$$
\begin{array}{ll}
ETCS_{\text{skel}} & : (train \cup rbc)^* \\
train & : spd;\ atp;\ drive \\
spd & : \ (?\tau.v \leq m.r;\ \tau.a := *;\ ? - b \leq \tau.a \leq A) \\
 & \ \cup (?\tau.v \geq m.r;\ \tau.a := *;\ ? - b \leq \tau.a \leq 0) \\
atp & : \text{if } (m.e - \tau.p \leq SB \vee rbc.message = emergency) \text{ then } \tau.a := -b \text{ fi} \\
drive & : t := 0;\ (\tau.p' = \tau.v \wedge \tau.v' = \tau.a \wedge t' = 1 \wedge \tau.v \geq 0 \wedge t \leq \varepsilon) \\
rbc & : (rbc.message := emergency)\ \cup\ (m := *;\ ?m.r > 0)
\end{array}
$$

Fig. 3: Formal model of parametric ETCS cooperation protocol (skeleton)

freight trains with a pressure-sensitive load within a tunnel. Our model captures this by reducing the speed component $m.d$ of m.

3.2 Formal Model of Fully Parametric ETCS

For analyzing the proper functioning of ETCS, we have developed a formal model of ETCS as a hybrid program (see Fig. 3) that is based on the informal specification [5]. RBC and train are independent distributed components running in parallel. They interoperate by message passing over wireless communication. As the RBC is a purely digital track-side controller and has no dependent continuous dynamics, we can express parallelism equivalently by interleaving using nondeterministic choice ($\cup$) and repetition (*): the decisions of the train controller only depend on the point in time where RBC messages arrive at the train, not the communication latency. Thus, the nondeterministic interleaving in ETCS where either the train or ($\cup$) the RBC chooses to take action faithfully models every possible arrival time without the need for an explicit channel model. The * at the end of $ETCS_{\text{skel}}$ indicates that the interleaving of train and RBC repeats arbitrarily often. Successive actions in each component are modelled using sequential composition ($;$). The train checks for its offset to the recommended speed (in *spd*) before checking if emergency breaking is necessary (in *atp*).

Train Controller. As it is difficult to use highly detailed models for the train and its mechanical transmission like in [2] directly in the verification and parameter discovery process, we first approximate it by a controller with a ranged choice for the effective acceleration $\tau.a$ between its lower bound ($-b$) and upper bound (A). (We will refine the dynamics in Sect. 5 and 6.) This controller provides a model that we can use both to derive parameter constraints, and to overapproximate the choices made by the physical train controller [2]. For Sect. 3–4, we model the continuous train dynamics by the differential equation system

$$\tau.p' = \tau.v \wedge \tau.v' = \tau.a \wedge t' = 1 \wedge \tau.v \geq 0 \wedge t \leq \varepsilon \ . \qquad (\mathcal{I})$$

It formalizes the ideal-world physical laws for movement, restricted to the evolution domain $\tau.v \geq 0 \wedge t \leq \varepsilon$ in *drive*. The primed variables stand for the first time-derivative of the respective unprimed variable. Therefore, $\tau.p'$ gives the rate with which the position of the train changes, i.e., the velocity ($\tau.p' = \tau.v$).

The velocity itself changes continuously according to the acceleration $\tau.a$, i.e., $\tau.v' = \tau.a$. The train speeds up when $\tau.a > 0$ and brakes when $\tau.a < 0$. In particular, for $\tau.a < 0$, the velocity would eventually become negative, which would mean the train is driving backwards. But that is prohibited without manual clearance, so we restrict the evolution domain to non-negative speed ($\tau.v \geq 0$). Time can be measured by clocks, i.e. variables changing with constant slope 1 ($t' = 1$). To further account conservatively for delayed effects of actuators like brakes or for delays caused by cycle times of periodic sensor polling and sampled data discrete time controllers, we permit the continuous movement of the train to continue for up to $\varepsilon > 0$ time units until control decisions finally take effect. This is expressed using the invariant region $t \leq \varepsilon$ on the clock t that is reset using the discrete assignment $t := 0$ before the continuous evolution starts. It is used to keep track of the progress of time advancing with constant slope 1. When the system executes the system of differential equations in *drive*, it can follow a continuous evolution respecting the constraints of ($\mathcal{I}$).

The speed supervision *spd* has two choices ($\cup$). The first option in Fig. 3 can be taken if the test $?\tau.v \leq m.r$ succeeds, the second one if the check $?\tau.v \geq m.r$ is successful. If both succeed, either choice is possible. The *spd* chooses the acceleration $\tau.a$ to keep the recommended speed $m.r$ by a random assignment $\tau.a := *$, which assigns an arbitrary value to $\tau.a$. By the subsequent test $? -b \leq \tau.a \leq 0$ an acceleration is chosen from the interval $[-b, 0]$ if the current speed $\tau.v$ exceeds $m.r$ (otherwise the full range $[-b, A]$ is available). Our controller includes controllers optimizing speed and energy consumption as secondary objectives.

As a supervisory controller, the automatic train protection (*atp* in Fig. 3) checks whether the point SB has been passed ($m.e - \tau.p \leq SB$) or a message from the RBC was received notifying of a track-side emergency situation. Both events cause immediate braking with full deceleration $-b$. Thus, *atp* decisions take precedence over *spd* speed advisory. In the case where $m.e - \tau.p > SB$ but no emergency message arrived the decisions made by *spd* take effect.

Radio Block Controller. We model the RBC as a controller with two possible choices ($\cup$). It may choose to demand immediate correction by sending emergency messages ($rbc.message := emergency$) or update the MA by assigning arbitrary new values to its three components ($m := *$). These nondeterministic changes to m reflect different real-world effects like extending $m.e$ and $m.d$ if the heading train has advanced significantly or, instead, notify of a new recommended speed $m.r$ for a track segment. We will identify safety-critical constraints on MA updates in Sect. 4.2.

4 Parametric Verification of Train Control

The model in Fig. 3 from the informal specification is unsafe, i.e., it does not always prevent collisions. To correct this we identify free parameter constraints by analyzing increasingly more complex correctness properties of ETCS. Using these constraints we refine the train control model iteratively into a safe model

with constraints on design parameter choices and physical prerequisites on external parameters resulting from the safety requirements on the train dynamics.

Iterative Refinement Process. For discovering parametric constraints required for system correctness, we follow an *iterative refinement process* that is of more general interest:

1. *Controllability discovery*: Start with uncontrolled system dynamics. Then use structural symbolic decomposition in $\mathsf{d}\mathcal{L}$ until a first-order formula is obtained revealing the controllable state region, which specifies for which parameter combinations the system dynamics can actually be controlled safely by any control law.
2. *Control refinement*: Successively add partial control laws to the system while leaving its decision parameters (like SB or m) free. Use structural symbolic decomposition to discover parametric constraints which preserve controllability under these control laws.
3. *Safety convergence*: Repeat step 2 until the resulting system is proven safe.
4. *Liveness check*: Prove that the discovered parametric constraints do not over-constrain the system inconsistently by showing that it remains live.

In practice, variants of the controllable domain as discovered by step 1 constitute good candidates for inductive invariants, and the parameter constraints discovered by step 2 ensure that the actual control choices taken by the controller never leave the controllable domain. For step 4, liveness can be verified again by structural symbolic decomposition in $\mathsf{d}\mathcal{L}$ and no need for separate verification techniques or different models arises.

4.1 Controllability Discovery in Parametric ETCS

By analyzing the uncontrolled train dynamics, we obtain a controllability constraint on the external train parameters, i.e., a formula characterizing the parameter combinations for which the train dynamics can be controlled safely by any control law at all. For our analysis we choose the following assumptions

$$\tau.v \geq 0 \wedge m.d \geq 0 \wedge b > 0 \qquad (\mathcal{A})$$

stating that the velocity is non-negative, the movement authority issued by the RBC does not force the train to drive backwards, and the train has some positive braking power b. The controllability constraint is now obtained by applying the $\mathsf{d}\mathcal{L}$ proof calculus [10] to the following $\mathsf{d}\mathcal{L}$ formula:

$$(\mathcal{A} \wedge \tau.p \leq m.e) \rightarrow [\tau.p' = \tau.v \wedge \tau.v' = -b \wedge \tau.v \geq 0]\mathcal{S} \ .$$

This means that starting in some state where $(\mathcal{A})$ holds and the train has not yet passed $m.e$ $(\tau.p \leq m.e)$ every possible evolution of the train system that applies full brakes $(\tau.v' = -b)$ is safe, i.e. does not violate $(\mathcal{S})$. This $\mathsf{d}\mathcal{L}$ formula only holds if $\tau.v^2 - m.d^2 \leq 2b(m.e - \tau.p)$. We prove that the so discovered constraint, illustrated in Fig. 4, characterizes the set of states where the train dynamics can still respect MA by appropriate control choices (expressed by the left-hand side $\mathsf{d}\mathcal{L}$ formula):

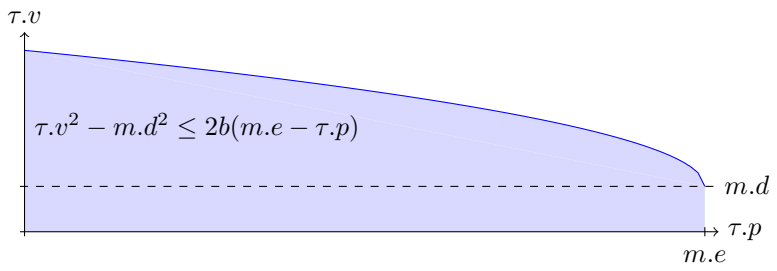


Fig. 4: Controllable region

Proposition 1 (Controllability). *The constraint* $\tau.v^2 - m.d^2 \leq 2b(m.e - \tau.p)$ *is a* controllability constraint *for the train* τ *with respect to property* $(\mathcal{S})$ *on page 8, i.e., the constraint retains the ability of the train dynamics to respect the safety property. Formally, with* $\mathcal{A} \wedge \tau.p \leq m.e$ *as regularity assumptions, the following equivalence is a valid* $\mathsf{d}\mathcal{L}$ *formula:*

$$\begin{aligned}&[\tau.p' = \tau.v \wedge \tau.v' = -b \wedge \tau.v \geq 0](\tau.p \geq m.e \rightarrow \tau.v \leq m.d)\\ \equiv\ &\tau.v^2 - m.d^2 \leq 2b(m.e - \tau.p)\end{aligned}$$

This formula expresses that *every run* of a train in braking mode satisfies $(\mathcal{S})$ if and only if condition $\tau.v^2 - m.d^2 \leq 2b(m.e - \tau.p)$ holds initially. Observe how the above equivalence reduces a $\mathsf{d}\mathcal{L}$ formula about future controllable train dynamics to a single constraint on the current state. We use this key reduction step from safe train dynamics to controllably safe state-constraints by analyzing whether each part of the ETCS controller preserves train controllability.

Definition 1 (Controllable state). *A train* τ *is in a* controllable state*, if the train is always able to stay within its movement authority* m *by appropriate control actions, which, by Proposition 1, is equivalent to*

$$\tau.v^2 - m.d^2 \leq 2b(m.e - \tau.p) \wedge \mathcal{A} \ . \tag{$\mathcal{C}$}$$

ETCS cannot be safe unless trains start and stay in controllable states. Hence we pick $(\mathcal{C})$ as a minimal candidate for an inductive invariant. This invariant will be used to prove safety of the system by induction even automatically using the technique in [23].

4.2 Iterative Control Refinement of ETCS Parameters

Starting from the constraints for controllable trains, we identify constraints for their various control decisions and refine the ETCS model correspondingly.

RBC Control Constraints. For a safe functioning of ETCS it is important that trains always respect their current MA. Consequently, RBCs are not allowed to issue MAs that are physically impossible for the train like instantaneous full stops. Instead RBCs are only allowed to send new MAs that remain within the controllable range of the train dynamics. For technical reasons the RBC does not reliably know the train positions and velocities in its domain of responsibility to a sufficient precision, because the communication with the trains has to be performed wirelessly with possibly high communication delay and message loss. Thus, we give a failsafe constraint for MA updates which is reliably safe even for loss of position recording communication.

Proposition 2 (RBC preserves train controllability). *The constraint*

$$m_0.d^2 - m.d^2 \leq 2b(m.e - m_0.e) \wedge m_0.d \geq 0 \wedge m.d \geq 0 \qquad (\mathcal{M})$$

ensures that the RBC preserves train controllability $(\mathcal{C})$ *when changing MA from* m_0 *to* m*, i.e., the following formula is valid:*

$$\forall\tau\Big(\mathcal{C} \rightarrow [m_0 := m;\ rbc]\ \big(\mathcal{M} \rightarrow \mathcal{C}\big)\Big)\ . \qquad (1)$$

This RBC controllability is characterized by the following valid formula:

$$m.d \geq 0 \wedge b > 0 \rightarrow [m_0 := m;\ rbc]\ \Big(\mathcal{M} \leftrightarrow\ \forall\tau\ \big((\langle m := m_0\rangle\mathcal{C}) \rightarrow \mathcal{C}\big)\Big)\ . \qquad (2)$$

Constraint $(\mathcal{M})$ characterizes that an extension is safe if it is possible to reduce the speed by braking with deceleration b from the old target speed $m_0.d$ to the new target speed $m.d$ within the extension range $m.e - m_0.e$, regardless of the current speed of train τ. It imposes constraints on feasible track profiles. Property (1) expresses that, for all trains in a controllable state $(\mathcal{C})$, every RBC change of MA m_0 to m that complies with $(\mathcal{M})$ enforces that the train is still in a controllable state $(\mathcal{C})$. Constraint $(\mathcal{M})$ is characterized by the equivalence (2), expressing that for every decision of *rbc*, $(\mathcal{M})$ holds for the RBC change from m_0 to m if and only if *all* trains $(\forall\tau)$ that were controllable $(\mathcal{C})$ for the previous MA (set using $\langle m := m_0\rangle$) remain controllable for the new MA m.

Train Control Constraints. Now that we found constraints characterizing when the cooperation of train and RBC is controllable, we need to find out under which circumstances the actual control choices by *spd* and *atp* retain controllability. In particular, the design parameter SB (start braking point relative to the end of the movement authority) needs to be chosen appropriately to preserve $(\mathcal{C})$. First we show that *there is* a choice of SB:

Proposition 3. *For all feasible RBC choices and all choices of speed control, there is a choice for* SB *that makes the train always stay within its MA, i.e., for controllable states, we can prove:*

$$\mathcal{C} \rightarrow [m_0 := m;\ rbc]\big(\mathcal{M} \rightarrow [spd]\langle SB := *\rangle[atp;\ drive]\mathcal{S}\big)\ .$$

The formula expresses that, starting in a controllable region $\mathcal{C}$, if the RBC updates the MA from m_0 to m respecting ($\mathcal{M}$), then after arbitrary *spd* choices, the train controller is still able to find some choice for SB ($\langle SB := * \rangle$) such that it always respect the fresh MA when following *atp* and *drive*. Since Proposition 3 is provable in KeYmaera we know that there is a safe solution for ETCS. On the formula level the assumptions are expressed using implications such that the formula does not make any proposition if either ($\mathcal{C}$) is not initially satisfied or the RBC does not respect ($\mathcal{M}$). The train controller is split up into the proposition that for all executions of the speed supervision ([*spd*]) there is a choice for SB ($\langle SB := * \rangle$) such that the automatic train protection unit (*atp*) always preserves safety during the execution of the trains movement in the *drive* phase. For *atp* and *drive* we again make a statement over all possible executions of the components. Only the choice of SB is existentially quantified.

To find a particular constraint on the choice of SB, we need to take the maximum reaction latency ε of the train controllers into account. With $\varepsilon > 0$, the point where the train needs to apply brakes to comply with m is not determined by ($\mathcal{C}$) alone, but needs additional safety margins to compensate for reaction delays. Therefore, we search for a constraint that characterizes that for every possible end of the movement authority ($\forall m.e$) and train position ($\forall \tau.v$), train movement with an acceleration of A preserves ($\mathcal{C}$) if it started in a state where ($\mathcal{C}$) holds and the point SB has not been passed yet ($m.e - \tau.p \geq SB \wedge \mathcal{C}$).

Proposition 4 (Reactivity constraint). *If the train is in a controllable state, the supervisory ETCS controller reacts appropriately in order to maintain controllability iff SB is chosen according to the following equivalence*

$$\Big(\forall m.e\, \forall \tau.p\, \big(m.e - \tau.p \geq SB \wedge \mathcal{C} \rightarrow [\tau.a := A;\ \mathit{drive}]\, \mathcal{C}\big)\Big)$$
$$\equiv SB \geq \frac{\tau.v^2 - m.d^2}{2b} + \left(\frac{A}{b} + 1\right)\left(\frac{A}{2}\varepsilon^2 + \varepsilon\ \tau.v\right) \quad . \tag{$\mathcal{B}$}$$

Constraint ($\mathcal{B}$) on SB is derived using a projection of the train behavior to the worst-case acceleration A in a state where SB has not been passed yet. We choose this projection because the train controller needs to ensure that it can drive safely with maximum acceleration A for ε time units even right before passing SB in order for an acceleration choice of A to be safe constraint ($\mathcal{B}$) is not obvious from the system model. After discovering constraint ($\mathcal{B}$), it can be explained in retrospect: It characterizes the relative braking distance required to reduce speed from $\tau.v$ to target speed $m.d$ with braking deceleration b, which corresponds to controllability and is expressed by the term $\frac{\tau.v^2 - m.d^2}{2b}$. In addition, it involves the distance travelled during one maximum reaction cycle of ε time units with acceleration A, including the additional distance needed to reduce the speed down to $\tau.v$ after accelerating with A for ε time units (expressed by $\left(\frac{A}{b} + 1\right)\left(\frac{A}{2}\varepsilon^2 + \varepsilon\ \tau.v\right)$). This extra distance results from speed changes and depends on the relation $\frac{A}{b}$ of maximum acceleration A and braking power b.

Propositions 1–4 prove equivalences. Hence, counterexamples exist for the ETCS skeleton in Fig. 3 whenever the parameter constraints are not met. Con-

$$
\begin{array}{lll}
ETCS_{\mathrm{r}} & : & (train_{\mathrm{r}} \cup rbc_{\mathrm{r}})^* \\
train_{\mathrm{r}} & : & spd;\ atp_{\mathrm{r}};\ drive \\
atp_{\mathrm{r}} & : & SB := \frac{\tau.v^2 - m.d^2}{2b} + \left(\frac{A}{b} + 1\right)\left(\frac{A}{2}\varepsilon^2 + \varepsilon\ \tau.v\right);\ atp \\
rbc_{\mathrm{r}} & : & (rbc.message := emergency) \\
 & & \cup \left(m_0 := m; m := *; ?m.r \geq 0 \wedge m.d \geq 0 \wedge m_0.d^2 - m.d^2 \leq 2b(m.e - m_0.e)\right)
\end{array}
$$

Fig. 5: Refined parametric ETCS cooperation protocol with bug-fixes to Fig. 3

sequently, these constraints must be respected for correctness of *any* model of ETCS controllers, including implementation refinements. It is, thus, important to identify these safety constraints early in the overall design and verification process.

4.3 Safety Verification of Refined ETCS

By augmenting the system from Fig. 3 with the parametric constraints obtained from Propositions 1–4, we synthesize a safe system model completing the ETCS protocol skeleton. The refined model is presented in Fig. 5 which bug-fixes the model in Fig. 3 taken from the informal specification (*spd*, *atp*, *drive* as in Fig. 3).

Proposition 5 (Safety). *Starting in a controllable state, this global and unbounded-horizon safety formula about the refined ETCS system in Fig. 5 is valid:*

$$\mathcal{C} \rightarrow [ETCS_r](\tau.p \geq m.e \rightarrow \tau.v \leq m.d)\ .$$

This provable formula states that, starting in a controllable region ($\mathcal{C}$), the augmented ETCS model is safe, i.e., trains always respect their movement authority.

As an example to illustrate the proof structure for the verification of Proposition 5, consider the sketch in Fig. 6. By convention, such proofs start with the conjecture at the bottom and proceed by decomposition to the leaves. We need to prove that universal controllability ($\mathcal{C}$) implies safety ($\mathcal{S}$) at all times. As the system consists of a global loop, we prove that ($\mathcal{C}$) is an invariant of this loop and strong enough to imply ($\mathcal{S}$). It can be shown easily that the invariant ($\mathcal{C}$) is initially valid (left branch) and implies the postcondition ($\mathcal{S}$) (right branch). As usual, proving that invariant ($\mathcal{C}$) is preserved by the loop body is the most challenging part of the proof in KeYmaera (middle branch), which splits into two cases. For the left case, we have to show that the RBC preserves the invariant, which can be proven like Proposition 2. For the right case, we show that the train controller preserves the invariant. The proof splits due to the choice in the *spd* component depending on the relation of the current speed to the recommended speed ($\tau.v$ vs. $m.r$). The next split on both of these branches depends on the relation of ($m.e - \tau.p$) and SB. If the train has passed point SB (middle case) the system is safe (Proposition 1), because the invariant describes a controllable state and the *atp* applies brakes. The outer branches, where the train has not yet passed SB, can be proven using Proposition 4.

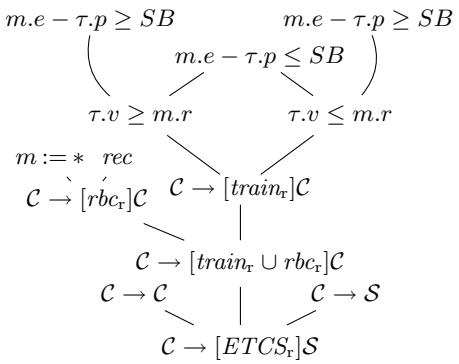


Fig. 6: Proof sketch for Proposition 5

4.4 Liveness Verification of Refined ETCS

In order to show that the discovered parameter constraints do not over-constrain the system inconsistently, we show liveness, i.e., that an ETCS train is able to reach every track position with appropriate RBC permissions.

Proposition 6 (Liveness). *The refined ETCS system is live, i.e., assuming the RBC can safely grant the required MAs because preceding trains are moving on, trains are able to reach any track position P by appropriate RBC choices:*

$$\tau.v \geq 0 \wedge \varepsilon > 0 \;\to\; \forall P \, \langle ETCS_r \rangle \, \tau.p \geq P$$

The formula expresses that, starting in a state where the velocity is non-negative and the maximum evolution time is positive, every point P ($\forall P$) can be reached ($\tau.p \geq P$) by some execution of the ETCS model ($\langle ETCS_{\mathrm{r}} \rangle$). Here the diamond modality is used to say that not all, but *some* appropriate execution reaches a state where the postcondition ($\tau.p \geq P$) holds. For showing that the system is live, a more liberal initial state is possible with regard to the controllability of the train. The only important restrictions on the initial region are those ensuring that the system of differential equations used for modelling the train movement can actually be followed for some positive amount of time. As usual the velocity of the train must be non-negative ($\tau.v \geq 0$). Additionally, ε has to be strictly greater than zero. If either of these assumptions is violated, the train may be unable to move as the evolution domain ($\tau.v \geq 0 \wedge t \leq \varepsilon$) of the differential equation system ($\mathcal{I}$) is left immediately and thus no continuous changes of the variables would be possible.

4.5 Full Correctness of ETCS

By collecting Propositions 1–6, we obtain the following main result of this paper, which demonstrates the feasibility of $\mathsf{d}\mathcal{L}$-based parametric discovery and verification supported by our theorem prover KeYmaera. It gives important insights

in the fully parametric ETCS case study and yields conclusive and fully verified choices for the free parameters in ETCS. By virtue of the parametric formulation, this result applies to all concrete instantiations of the ETCS cooperation protocol from Sect. 3, including controllers that further optimize speed or model refinements in hardware implementations.

Theorem 1 (Correctness of ETCS cooperation protocol). *The ETCS system augmented with constraints* $(\mathcal{B})$ *and* $(\mathcal{M})$ *is correct as given in Fig. 5. Starting in any controllable state respecting* $(\mathcal{C})$*, trains remain in the controllable region at any time. They safely respect movement authorities issued by the RBC so that ETCS is collision-free. Further, trains can always react safely to all RBC decisions respecting* $(\mathcal{M})$*. ETCS is live: When tracks become free, trains are able to reach any track position by appropriate RBC actions. Furthermore, the augmented constraints* $(\mathcal{C})$ *and* $(\mathcal{B})$ *are necessary and sharp: Every configuration violating* $(\mathcal{C})$ *or* $(\mathcal{B})$*, respectively, gives rise to a concrete counterexample violating safety property* $(\mathcal{S})$*. Finally, every RBC choice violating* $(\mathcal{M})$ *gives rise to a counterexample in the presence of lossy wireless communication channels.*

5 Inclusion and Safety of PI Controllers

Trains use *proportional-integral* (PI) controllers for speed supervision [2] like most physical control systems do. A PI uses a linear combination of the proportional and integral values of the difference between the current ($\tau.v$) and the target system state ($m.r$) to determine control actions. The proportional part uses the current error $\tau.v - m.r$ of the system state compared to the target state with some factor l, whereas the integral part sums up previous errors $\int(\tau.v - m.r)dt$ with some factor i. Damm et al. have identified a detailed train model with a PI controller [2]. The resulting PI corresponds to the differential equation system

$$\tau.p' = \tau.v \wedge \tau.v' = \min\Big(A, \max\big(-b,\ l(\tau.v - m.r) - i\,s - c\,m.r\big)\Big) \\ \wedge\ s' = \tau.v - m.r \wedge t' = 1 \wedge \tau.v \geq 0 \wedge t \leq \varepsilon\ . \quad (\mathcal{P})$$

The position of the train $\tau.p$ changes according to its velocity $\tau.v$ ($\tau.p' = \tau.v$) and $\tau.v$ changes according to the acceleration determined by PI equations. Variable s tracks the integral part of the controller: differential equation $s' = \tau.v - m.r$ corresponds to integral equation $s = \int(\tau.v - m.r)dt$. Thus $i\,s$ represents the integral share of the error scaled by i in the PI. Since trains do not drive backwards by braking, the system contains an evolution domain stating that the speed remains non-negative ($\tau.v \geq 0$). PI $\mathcal{P}$ influences the velocity by changing the acceleration of the train according to proportional and integral changes compared to recommended speed $m.r$. The parameters l, i and c are derived from the train physics and chosen in a way such that the controller does not oscillate. Note that classical PIs use $c = 0$. We also allow $c \neq 0$, which is

used in the refined PI controller identified in [2] for additional attenuation. Following [2], the controller further obeys physical bounds for the acceleration and is restricted to values between $-b < 0$ and $A > 0$ using min, max functions.

In this section we relate this model for the train control with the approximation ($\mathcal{I}$) used in Sect. 3–4. First, we prove that our abstraction is a valid overapproximation by showing that whatever the PI controller ($\mathcal{P}$) does, the ideal-world physical controller for ($\mathcal{I}$) can reach the same point within the same time. Unlike ($\mathcal{I}$), we cannot simply solve PI ($\mathcal{P}$) in polynomial arithmetic to prove properties. We use differential invariants [24, 23] instead for proofs.

Proposition 7 (PI inclusion). *Starting from 0, every possible execution of the PI controller ($\mathcal{P}$) can be imitated by the ranged controller*

$$spd_s := (\tau.a := *;\ ?\tau.a \geq -b \wedge \tau.a \leq A)$$

for the dynamics ($\mathcal{I}$) such that they are in the same place at the same time:

$$[\mathcal{P} \wedge t'_\pi = 1]\, \langle (spd_s;\ t := 0;\ \mathcal{I} \wedge t'_\tau = 1)^* \rangle\ (\pi.p = \tau.p \wedge t_\pi = t_\tau)$$

That is, for every evolution of ($\mathcal{P}$), spd_s can choose its options such that ($\mathcal{I}$) reaches the same point $\pi.p$ at the same time t_π. Here t_π is a clock ($t'_\pi = 1$) measuring the time the first controller ($\mathcal{P}$) consumes and t_τ measures the time needed by the second controller to reach the same position at the same time.

The ranged controller spd_s is less restrictive than spd, because it allows more liberal acceleration choices. As the previous propositions do not depend on the value of $m.r$ showing the inclusion property for spd_s is sufficient.

With the constraints in $ETCS_{\mathrm{r}}$, we verify that the fully parametric PI controller combined with the automatic train protection atp_{r} preserves safety:

Proposition 8 (Safety of the PI-controlled system). *For trains in controllable state, the $ETCS_r$ system with a PI controller for speed regulation is safe, i.e., when replacing drive by ($\mathcal{P}_e \wedge t' = 1 \wedge t \leq \varepsilon$) for (continuous) speed supervision and with emergency braking according to Fig. 5. This corresponds to the physical train model identified in [2]. Here $\mathcal{P}_e$ regulates speed like $\mathcal{P}$ in normal operation and is disabled if atp forces emergency braking:*

$$\mathcal{C} \rightarrow \Big[\big((atp_r;\ t := 0;\ \mathit{if}\ (\tau.a = -b)\ \mathit{then}\ \mathcal{I}\ \mathit{else}\ \mathcal{P} \wedge t' = 1 \wedge t \leq \varepsilon\ \mathit{fi}) \cup\ rbc_r\big)^*\Big]$$
$$(\tau.p \geq m.e \rightarrow \tau.v \leq m.d)$$

Note that the use of ($\mathcal{I}$) accounts for the fact that emergency measures simply apply full brakes instead of regulating the speed to some target value. This is modelled using the fixed deceleration of b.

6 Disturbance and the European Train Control System

In Sect. 3–4, we assumed direct control of acceleration. In reality, acceleration results from physical transmission of corresponding forces that depend on the

electrical current in the engine [2]. As a conservative overapproximation of these effects, we generalize the ETCS model to a model with *differential inequalities* [24], where we also take into account disturbances in the physical transmission of forces (including wind, friction etc.):

$$\tau.p' = \tau.v \wedge \tau.a - l \leq \tau.v' \leq \tau.a + u \wedge t' = 1 \wedge \tau.v \geq 0 \wedge t \leq \varepsilon \qquad (\mathcal{I}_d)$$

with a disturbance within the interval $[-l, u]$. That is, the acceleration $\tau.a$ chosen by the train controller can take effect with an error bounded by $-l$ and u, because the derivative $\tau.v'$ of the velocity will not need to be $\tau.a$ exactly in $(\mathcal{I}_d)$, but $\tau.v'$ can vary arbitrarily between $\tau.a - l$ and $\tau.a + u$ over time. We generalize the differential equation $(\mathcal{I})$ in component *train* from Fig. 3 and Fig. 5 by replacing it with the differential inequality $(\mathcal{I}_d)$ and denote the result by $train_d$.

Notice that, unlike $(\mathcal{I})$, we cannot simply solve differential inequality $(\mathcal{I}_d)$, because its actual solution depends on the precise value of the disturbance, which is a quantity that changes over time. Thus, solutions would only be relative to this disturbance function and a reachability analysis would have to consider all choices of this function, which would require higher-order logic. Instead, we verify using differential invariants [24, 23] as a sound first-order characterization.

6.1 Controllability in ETCS with Disturbances

The controllability characterization from Proposition 1 carries over to train control with disturbance when taking into account the maximum disturbance u on the maximum braking power b that limit the effective braking power to $(b - u)$:

Proposition 9 (Controllability despite disturbance). *The constraint*

$$\tau.v^2 - m.d^2 \leq 2(b-u)(m.e - \tau.p) \wedge m.d \geq 0 \wedge b > u \geq 0 \wedge l \geq 0 \qquad (\mathcal{C}_d)$$

is a controllability constraint *with respect to property* $(\mathcal{S})$ *for the train* τ *with disturbance* $(\mathcal{I}_d)$*, i.e., it retains the ability of the train dynamics to respect the safety property despite disturbance. Formally, with* $\mathcal{A} \wedge \tau.p \leq m.e \wedge b > u \geq 0 \wedge l \geq 0$ *as regularity assumptions, the following equivalence holds:*

$$\begin{aligned} &[\tau.p' = \tau.v \wedge \tau.a - l \leq \tau.v' \leq \tau.a + u \wedge t' = 1 \wedge \tau.v \geq 0 \wedge t \leq \varepsilon]\, \mathcal{S} \\ \equiv\ &\tau.v^2 - m.d^2 \leq 2(b-u)(m.e - \tau.p) \end{aligned}$$

Here $(\mathcal{C}_d)$ results from $(\mathcal{C})$ by replacing b with $(b-u)$. In worst case disturbance, the train cannot brake with deceleration $-b$ but instead might be off by u. To guarantee that the train is able to stay within its MA the controller has to assume maximum guaranteed deceleration $-(b - u)$ when making control decision.

6.2 Iterative Control Refinement of Parameters with Disturbances

When taking into account worst-case effects of disturbance on control, reactivity constraint $(\mathcal{B})$ carries over to the presence of disturbance in the train dynamics:

Proposition 10 (Reactivity constraint despite disturbance). *For trains in controllable state, the supervisory ETCS controller reacts appropriately despite disturbance in order to maintain controllability iff SB is chosen according to the following provable equivalence:*

$$\begin{aligned}&\Big(\forall m.e\,\forall \tau.p\,\big((m.e-\tau.p \geq SB \wedge \tau.v^2 - m.d^2 \leq 2(b-u)(m.e-\tau.p)) \rightarrow \\ &\qquad [\tau.a := A;\ drive_d](\tau.v^2 - m.d^2 \leq 2(b-u)(m.e-\tau.p))\big)\Big) \\ \equiv\ & SB \geq \frac{\tau.v^2 - m.d^2}{2(b-u)} + \left(\frac{A+u}{b-u}+1\right)\left(\frac{A+u}{2}\varepsilon^2 + \varepsilon\tau.v\right) \qquad (\mathcal{B}_d)\end{aligned}$$

For reactivity $(\mathcal{B}_d)$ not only the maximum deceleration but also the maximum acceleration matters. Therefore, we need to substitute every b by $(b-u)$ but also every A with $(A+u)$ which is the maximum acceleration under disturbance to get a (provable) reactivity constraint for the disturbed system.

6.3 Safety Verification of ETCS with Disturbances

When we augment the ETCS model by the constraints $(\mathcal{B}_d)$ and $(\mathcal{M}_d)$, where $(\mathcal{M}_d)$ results from $(\mathcal{M})$ by again replacing every b by $(b-u)$, ETCS is safe even in the presence of disturbance when starting in a state respecting $(\mathcal{C}_d)$.

Proposition 11 (Safety despite disturbance). *Assuming the train starts in a controllable state satisfying $(\mathcal{C}_d)$, the following global and unbounded-horizon safety formula about the ETCS system with disturbance from Fig. 7 is valid:*

$$\mathcal{C}_d \rightarrow [ETCS_d](\tau.p \geq m.e \rightarrow \tau.v \leq m.d)\ .$$

This safety proof generalizes to ETCS with disturbance, using differential induction [24, 23] with a time-dependent version of $(\mathcal{B}_d)$ as differential invariant for the acceleration case:

$$m.e - \tau.p \geq \frac{\tau.v^2 - m.d^2}{2(b-u)} + \left(\frac{A+u}{b-u}+1\right)\left(\frac{A+u}{2}(\varepsilon-t)^2 + (\varepsilon-t)\tau.v\right)$$

7 Experimental Results

Tab. 2 shows experimental results for verifying ETCS in our $\mathsf{d}\mathcal{L}$-based verification tool KeYmaera [11]. The results are from a system with two quad core Intel Xeon E5430 (2.66 GHz per core, using only one core) and 32 gigabyte of RAM. All correctness properties and parameter constraints of ETCS can be verified with 91% to 100% degree automation. More than 91% of the proof steps are fully automatic. The proofs are 100% automatic in 6 properties and require minor guidance in 7 more challenging cases. Tab. 2 gives the number of user interactions necessary in the column Int, for comparison the total number of applied proof

$$
\begin{array}{ll}
ETCS_{\mathrm{d}} & : (train_d \cup rbc_{\mathrm{d}})^* \\
train_d & : spd;\ atp_d;\ drive_d \\
atp_d & : SB := \frac{\tau.v^2 - m.d^2}{2(b-u)} + \left(\frac{A+u}{b-u} + 1\right)\left(\frac{A+u}{2}\varepsilon^2 + \varepsilon\tau.v\right); \\
 & \quad \text{if } (m.e - \tau.p \leq SB \vee rbc.message = emergency) \text{ then } \tau.a := -b \text{ fi} \\
drive_d & : t := 0;\ (\tau.p' = \tau.v \wedge \tau.a - l \leq \tau.v' \leq \tau.a + u \wedge t' = 1 \wedge \tau.v \geq 0 \wedge t \leq \varepsilon) \\
rbc_{\mathrm{d}} & : \ (rbc.message := emergency) \\
 & \quad \cup \left(m_0 := m; m := *; \right. \\
 & \qquad \left. ?m.r \geq 0 \wedge m.d \geq 0 \wedge m_0.d^2 - m.d^2 \leq 2(b-u)(m.e - m_0.e)\right)
\end{array}
$$

Fig. 7: Parametric ETCS cooperation protocol with disturbances

rules in column Steps. In most cases proofs can be found automatically [23]. For more complicated properties beyond the capabilities of currently available decision procedures for real arithmetic, KeYmaera needs more user guidance but they can still be verified with KeYmaera! We see that the formula complexity and symbolic state dimension (Dim) has more impact on the computational complexity than the number of proof steps in $\mathsf{d}\mathcal{L}$ decompositions, which indicates good scalability in terms of the size of the system model.

Table 2: Experimental results for the European Train Control System

Case study			Int	Time(s)	Memory(MB)	Steps	Dim
Controllability	Proposition 1		0	1.3	29.6	14	5
Refinement	Proposition 2	eqn. (1)	0	1.7	29.0	42	12
RBC Control	Proposition 2	eqn. (2)	0	2.2	29.0	42	12
Reactivity	Proposition 3		8	133.4	118.7	229	13
Reactivity	Proposition 4		0	86.8	688.2	52	14
Safety	Proposition 5		0	249.9	127.8	153	14
Liveness	Proposition 6		4	27.3	100.7	166	7
Inclusion	Proposition 7	PI	19	766.2	354.4	301	25
Safety	Proposition 8	PI	16	509.0	688.2	183	15
Safety	Corollary 1	2 trains	4	≈2277.5	156.3	389	16
Controllability	Proposition 9	disturbed	0	5.6	30.8	37	7
Reactivity	Proposition 10	disturbed	2	34.6	74.3	78	15
Safety	Proposition 11	disturbed	5	389.9	41.7	88	16

8 Summary

As a case study for parametric verification of hybrid systems, we have verified controllability, reactivity, safety, and liveness of the fully parametric cooperation protocol of the European Train Control System. We have demonstrated the feasibility of logic-based verification of parametric hybrid systems and identified parametric constraints that are both sufficient and necessary for a safe

collision-free operation of ETCS. We have characterized these constraints on the free parameters of ETCS equivalently in terms of corresponding reachability properties of the underlying train dynamics. We have verified a corresponding fully parametric PI controller and proven that the system remains correct even when the train dynamics is subject to disturbances caused, e.g., by the physical transmission, friction, or wind.

We have shown how the properties of train control can be expressed in $\mathsf{d}\mathcal{L}$. Our experimental results with KeYmaera show a scalable approach by combining the power of completely automatic verification procedures with the intuition behind user guidance to tackle even highly parametric hybrid systems and properties with substantial quantifier alternation (reactivity or liveness) or disturbance.

We have verified all propositions formally in the KeYmaera tool. We present proof sketches in the Appendix.

Acknowledgments. We like to thank Johannes Faber, Sven Linker, and Ernst-Rüdiger Olderog for useful remarks on preliminary versions of this paper. Additionally, we like to thank the anonymous referees for their helpful comments on the conference version [25] of this paper.

References

1. Meyer, R., Faber, J., Hoenicke, J., Rybalchenko, A.: Model checking duration calculus: A practical approach. FACS **20**(4–5) (2008) 481–505
2. Damm, W., Mikschl, A., Oehlerking, J., Olderog, E.R., Pang, J., Platzer, A., Segelken, M., Wirtz, B.: Automating verification of cooperation, control, and design in traffic applications. In Jones, C.B., Liu, Z., Woodcock, J., eds.: Formal Methods and Hybrid Real-Time Systems. Volume 4700 of Lecture Notes in Computer Science., Springer (2007) 115–169
3. Batt, G., Belta, C., Weiss, R.: Model checking genetic regulatory networks with parameter uncertainty. [29] 61–75
4. Alur, R., Henzinger, T.A., Ho, P.H.: Automatic symbolic verification of embedded systems. IEEE Trans. Software Eng. **22**(3) (1996) 181–201
5. ERTMS User Group, UNISIG: ERTMS/ETCS System requirements specification. http://www.era.europa.eu (2002) Version 2.2.2.
6. Henzinger, T.A.: The theory of hybrid automata. In: LICS, IEEE CS Press (1996)
7. Mysore, V., Piazza, C., Mishra, B.: Algorithmic algebraic model checking ii: Decidability of semi-algebraic model checking and its applications to systems biology. In Peled, D., Tsay, Y.K., eds.: ATVA. Volume 3707 of Lecture Notes in Computer Science., Springer (2005) 217–233
8. Frehse, G.: Phaver: algorithmic verification of hybrid systems past hytech. STTT **10**(3) (2008) 263–279
9. Clarke, E., Grumberg, O., Jha, S., Lu, Y., Veith, H.: Counterexample-guided abstraction refinement for symbolic model checking. J. ACM **50**(5) (2003)
10. Platzer, A.: Differential dynamic logic for hybrid systems. J. Autom. Reasoning **41**(2) (2008) 143–189

11. Platzer, A., Quesel, J.D.: KeYmaera: A hybrid theorem prover for hybrid systems. In Armando, A., Baumgartner, P., Dowek, G., eds.: IJCAR. Volume 5195 of LNCS., Springer (2008) 171–178 http://symbolaris.com/info/KeYmaera.html.
12. Platzer, A., Quesel, J.D.: Logical verification and systematic parametric analysis in train control. In Egerstedt, M., Mishra, B., eds.: HSCC. LNCS, Springer (2008)
13. Fränzle, M.: Analysis of hybrid systems: An ounce of realism can save an infinity of states. In Flum, J., Rodríguez-Artalejo, M., eds.: CSL. Volume 1683 of LNCS., Springer (1999) 126–140
14. Frehse, G., Jha, S.K., Krogh, B.H.: A counterexample-guided approach to parameter synthesis for linear hybrid automata. In Egerstedt, M., Mishra, B., eds.: HSCC. Volume 4981 of LNCS., Springer (2008) 187–200
15. Tomlin, C., Lygeros, J., Sastry, S.: A Game Theoretic Approach to Controller Design for Hybrid Systems. Proceedings of IEEE **88** (2000) 949–969
16. Peleska, J., Große, D., Haxthausen, A.E., Drechsler, R.: Automated verification for train control systems. In: FORMS/FORMAT. (2004)
17. Cimatti, A., Roveri, M., Tonetta, S.: Requirements validation for hybrid systems. In Bouajjani, A., Maler, O., eds.: CAV. Volume 5643 of LNCS., Springer (2009)
18. Davoren, J.M.: On hybrid systems and the modal μ-calculus. In Antsaklis, P.J., Kohn, W., Lemmon, M.D., Nerode, A., Sastry, S., eds.: Hybrid Systems. Volume 1567 of LNCS., Springer (1997) 38–69
19. Davoren, J.M., Nerode, A.: Logics for hybrid systems. Proceedings of the IEEE **88**(7) (2000) 985–1010
20. Davoren, J.M., Tabuada, P.: On simulations and bisimulations of general flow systems. [29] 145–158
21. Manna, Z., Sipma, H.: Deductive verification of hybrid systems using STeP. In Henzinger, T.A., Sastry, S., eds.: HSCC. Volume 1386 of LNCS., Springer (1998) 305–318
22. Ábrahám-Mumm, E., Steffen, M., Hannemann, U.: Verification of hybrid systems: Formalization and proof rules in PVS. In: ICECCS, IEEE Computer (2001) 48–57
23. Platzer, A., Clarke, E.M.: Computing differential invariants of hybrid systems as fixedpoints. Form. Methods Syst. Des. **35**(1) (2009) 98–120 Special CAV'08 issue.
24. Platzer, A.: Differential-algebraic dynamic logic for differential-algebraic programs. J. Log. Comput. (2008) DOI 10.1093/logcom/exn070.
25. Platzer, A., Quesel, J.D.: European train control system: A case study in formal verification. In Cavalcanti, A., Breitman, K., eds.: Formal Methods and Software Engineering, 11th International Conference on Formal Engineering Methods, ICFEM 2009, Rio de Janeiro, December 9-12, 2009, Proceedings. Volume 5885 of LNCS., Heidelberg, Springer (2009) 246–265
26. Tarski, A.: A Decision Method for Elementary Algebra and Geometry. 2nd edn. University of California Press, Berkeley (1951)
27. Collins, G.E., Hong, H.: Partial cylindrical algebraic decomposition for quantifier elimination. J. Symb. Comput. **12**(3) (1991) 299–328
28. Platzer, A.: Combining deduction and algebraic constraints for hybrid system analysis. In Beckert, B., ed.: VERIFY'07 at CADE, Bremen, Germany. Volume 259 of CEUR Workshop Proceedings., CEUR-WS.org (2007) 164–178
29. Bemporad, A., Bicchi, A., Buttazzo, G.C., eds.: Hybrid Systems: Computation and Control, 10th International Workshop, HSCC 2007, Pisa, Italy, April 3-5, 2007, Proceedings. In Bemporad, A., Bicchi, A., Buttazzo, G.C., eds.: HSCC. Volume 4416 of Lecture Notes in Computer Science., Springer (2007)

A Semantics of Differential Dynamic Logic

The semantics of $\mathsf{d}\mathcal{L}$ is a Kripke semantics assigning real values to variables. A state is a map $\nu : V \to \mathbb{R}$; the set of all states is denoted by $\mathrm{Sta}(\Sigma, V)$. The truth-value of a formula ϕ depends on the state ν, we denote it by $val(\nu, \phi)$ (Def. 4). The semantics of HP α is captured by the state transitions that are possible by running α. For continuous evolutions, the transition relation holds for pairs of states that can be interconnected by a continuous flow respecting the differential equation and invariant region.

Definition 2 (Valuation of Terms). *The* valuation of terms *with respect to state ν is defined by:*

1. $val(\nu, x) = \nu(x)$ *if x is a variable*
2. $val(\nu, f(\theta_1, \ldots, \theta_n)) = f\big(val(\nu, \theta_1), \ldots, val(\nu, \theta_n)\big)$, *where f is the mathematical operation associated to function symbol f.*

Definition 3 (Transitions of hybrid programs). *The* transition relation, $\rho(\alpha)$, of a hybrid program α, *specifies which state ω is reachable from a state ν by operations of the hybrid system α and is defined as follows*

1. $(\nu, \omega) \in \rho(x := \theta)$ *iff the state ω is identical to ν except that $\omega(x) = val(\nu, \theta)$.*
2. $(\nu, \omega) \in \rho(x := *)$ *iff state ω is identical to ν except for the value of x, which can be any real number.*
3. $(\nu, \omega) \in \rho(x_1' = \theta_1 \wedge \cdots \wedge x_n' = \theta_n \wedge \chi)$ *iff there is a continuous function $f : [0, r] \to \mathrm{Sta}(\Sigma, V)$ going from $f(0) = \nu$ to $f(r) = \omega$, which solves the system of differential equations, i.e., for all $i \in [1, n]$ $val(f(\zeta), x_i)$ has a derivative of value $val(f(\zeta), \theta_i)$ at each time $\zeta \in (0, r)$. Other variables remain constant: $val(f(\zeta), y) = val(\nu, y)$ for $y \neq x_i$, for all $i \in [1, n]$ and $\zeta \in [0, r]$. And the evolution domain χ is respected: $val(f(\zeta), \chi) = true$ for each $\zeta \in [0, r]$.*
4. $\rho(?\chi) = \{(\nu, \nu) \ : \ val(\nu, \chi) = true\}$
5. $\rho(\alpha \cup \beta) = \rho(\alpha) \cup \rho(\beta)$
6. $\rho(\alpha; \beta) = \rho(\alpha) \circ \rho(\beta) = \{(\nu, \omega) : (\nu, z) \in \rho(\alpha), (z, \omega) \in \rho(\beta)$ *for some state* $z\}$
7. $(\nu, \omega) \in \rho(\alpha^*)$ *iff there are* $n \in \mathbb{N}$ *and* $\nu{=}\nu_0, \ldots, \nu_n{=}\omega$ *with* $(\nu_i, \nu_{i+1}) \in \rho(\alpha)$ *for all* $0 \leq i < n$.
8. $\rho(\text{if}(\chi) \text{ then } \alpha \text{ fi}) = \{(\nu, \omega) \ : \ val(\nu, \chi) = true \text{ and } (\nu, \omega) \in \rho(\alpha)\}$ $\cup \{(\nu, \nu) \ : \ val(\nu, \chi) = false\}$
9. $\rho(\text{if}(\chi) \text{ then } \alpha \text{ else } \beta \text{ fi}) = \{(\nu, \omega) \ : \ val(\nu, \chi) = true \text{ and } (\nu, \omega) \in \rho(\alpha)\}$ $\cup \{(\nu, \omega) \ : \ val(\nu, \chi) = false \text{ and } (\nu, \omega) \in \rho(\beta)\}$

For the semantics of differential inequality systems we refer the reader to [10].

Definition 4 (Valuation of Formulas). *The* valuation of formulas *with respect to ν is defined by:*

1. $val(\nu, p(\theta_1, \ldots, \theta_n)) = p\big(val(\nu, \theta_1), \ldots, val(\nu, \theta_n)\big)$, *where p is the mathematical relation associated to predicate p*
2. $val(\nu, \phi \wedge \psi)$ *is defined as usual, the same holds for* $\neg, \vee, \rightarrow, \leftrightarrow$
3. $val(\nu, \forall x\, \phi) = true :\Longleftrightarrow val(\nu[x \mapsto d], \phi) = true$ *for all* $d \in \mathbb{R}$
4. $val(\nu, \exists x\, \phi) = true :\Longleftrightarrow val(\nu[x \mapsto d], \phi) = true$ *for some* $d \in \mathbb{R}$
5. $val(\nu, [\alpha]\phi) = true :\Longleftrightarrow val(\omega, \phi) = true$ *for all* ω *with* $(\nu, \omega) \in \rho(\alpha)$
6. $val(\nu, \langle\alpha\rangle\phi) = true :\Longleftrightarrow val(\omega, \phi) = true$ *for some* ω *with* $(\nu, \omega) \in \rho(\alpha)$

B Proof for Lemma 1: Principle of MA Separation

In this section we present the formal proof for Lemma 1, which reduces the proof task for showing safety of an a priori unbound number of trains to the task where we only have to show that one train stays within its movement authority.

Proof (of Lemma 1). To simplify notation, we assume trains are points (the proof is a simple extension when each train has some maximal length l). Consider any point in time ζ. For some $n \in \mathbb{N}$, let $z_1, \ldots, z_n$ be the positions of all the trains 1 to n at ζ. Let M_i be the MA-range, i.e., the set of positions on the track for which train i has currently been issued MA. Suppose there was a collision at time ζ. Then $z_i = z_j$ at ζ for some $i, j \in \mathbb{N}$. However, by assumption, $z_i \in M_i$ and $z_j \in M_j$ at ζ, thus $M_i \cap M_j \neq \emptyset$, which contradicts the assumption of disjoint MA. □

C Additional Properties

Lemma 1 enables us to decouple safety proofs for arbitrarily many trains into an analysis of the interaction of one generic train with the RBC. To demonstrate that our proof method is capable of showing safety for a scaled version of our model with more than one train directly, we verify the following simple corollary to Lemma 1 and Proposition 5 from scratch. It is, in fact, a simple consequence of those two results. We verify it directly in KeYmaera without reference to Lemma 1 and Proposition 5 to analyze scalability of KeYmaera. To simplify the notation, we assume that the target speed $m.d$ is 0 for both trains.

Corollary 1 (Safety of two trains). *A system with one RBC and two trains (τ_1 and τ_2) in controllable state is safe, if the trains are initially separated by their movement authorities, i.e.*

$$
\begin{aligned}
&(\mathcal{C}_1 \wedge \mathcal{C}_2 \wedge \tau_1.p \geq \tau_2.p \wedge \tau_1.p \geq m_2.e) \rightarrow \\
&\quad [rbc_{1,2} \cup \big(spd_{1,2};\ atp_{1,2};\ t := 0;\ \tau_1.p' = \tau_1.v, \tau_1.v' = \tau_1.a, \tau_2.p' = \tau_2.v, \\
&\qquad\qquad \tau_2.v' = \tau_2.a, t' = 1, (\tau_1.v \geq 0 \wedge \tau_2.v \geq 0, t \leq \varepsilon)\big)] \\
&\qquad\qquad\qquad\qquad (\tau_1.p \leq m_1.e \wedge \tau_2.p \leq m_2.e \wedge \tau_1.p \geq \tau_2.p)
\end{aligned}
$$

We choose that initially the trains are positioned on the track such that train τ_1 has a greater position than train τ_2 ($\tau_1.p \geq \tau_2.p$) and the movement authority of the second train must end before it might crash into the first one ($\tau_1.p \geq m_2.e$). Additionally both trains need to be controllable ($\mathcal{C}_1 \wedge \mathcal{C}_2$). Here $\mathcal{C}_1$ and $\mathcal{C}_2$ denote the controllability constraints for the two trains, respectively, i.e. for $i \in \{1, 2\}$:

$$\mathcal{C}_i \equiv \tau_i.v^2 \leq 2b(m_i.e - \tau_i.p) \wedge \tau_i.v \geq 0 \wedge b > 0$$

The RBC is modified such that it assigns a movement authority m_1 to train τ_1 like the refined RBC in Fig. 5 and additionally is able to assign a movement authority m_2 to τ_2 where $m_2.e$ is subject to the constraint $m_2.e \leq \tau_1.p$:

$$\begin{aligned} rbc_{1,2} : {} & (rbc.message := emergency) \\ & \cup \big(m_0 := m_1; m_1 := *; ?m_1.r \geq 0 \wedge m_0.d^2 \leq 2b(m_1.e - m_0.e)\big) \\ & \cup \big(m_0 := m_2; m_2 := *; ?m_2.r \geq 0 \wedge m_0.d^2 \leq 2b(m_2.e - m_0.e) \\ & \qquad \wedge m_2.e < \tau_1.p\big) \end{aligned}$$

According to Lemma 1, the RBC should never extend movement authority m_2 into the region of train τ_1, hence the constraint $m_2.e \leq \tau_1.p$.

The controllers $spd_{1,2}$ and $atp_{1,2}$ are two direct instantiations of the controllers presented in Fig. 5 that are executed in sequential order. All variables in the instantiations are indexed with $i \in \{1, 2\}$ respectively to indicate which train they belong to:

$$\begin{aligned} spd_i : {} & (?\tau_i.v \leq m_i.r;\ \tau_i.a := *;\ ? - b \leq \tau_i.a \leq A) \\ & \cup (?\tau_i.v \geq m_i.r;\ \tau_i.a := *;\ ? - b \leq \tau_i.a \leq 0) \\ atp_i : {} & SB_i := \tfrac{\tau_i.v^2 - m_i.d^2}{2b} + \left(\tfrac{A}{b} + 1\right)\left(\tfrac{A}{2}\varepsilon^2 + \varepsilon\ \tau_i.v\right); \\ & \text{if } (m.e - \tau.p \leq SB_i \vee rbc.message = emergency) \text{ then } \tau.a := -b \text{ fi} \end{aligned}$$

For simplicity the two trains share the constants A, b and ε. The movement of the two trains is expressed as a joint differential equation system sharing the clock t. This means the position $\tau_1.p$ of the first train changes according to its specific velocity $\tau_1.v$ which also changes with rate $\tau_1.a$. For the second train variables of the vector τ_2 are used, but the movement law stays the same. As $m_1.d$ and $m_2.d$ are zero, safety of the system means that the trains do not pass the end of their respective movement authority ($\tau_i.p \leq m_i.e$) and, as there is only one track, the initial order of the trains is preserved ($\tau_1.p \geq \tau_2.p$). Note that, for simplicity, we do not model safe rear-end computation of trains, but assume trains are points instead for 1.

D Proof Sketches

In this section we illustrate the proofs of the propositions presented in the main article as found and conducted in KeYmaera. All these propositions have been performed and verified formally in the verification tool KeYmaera [11] using the proof calculus of $\mathsf{d}\mathcal{L}$ [10].

The $\mathsf{d}\mathcal{L}$ verification calculus is a sequent calculus. In the following proof sketches we will use the *sequent notation* $\Gamma \vdash \Delta$ to indicate that one set of formulas Γ implies another one Δ. The sequent is valid if and only if the conjunction of all formulas in the *antecedent* Γ implies the disjunction of the formulas in the *succedent* Δ. We say that we can *close* a goal, if it is a valid axiom $\Gamma, \phi \vdash \phi, \Delta$ or can be proven by quantifier elimination in real arithmetic [26, 27].

In the following we explain the proofs carried out in KeYmaera. Some of the proof sketches show a simplified version of the proof tree for illustration purposes. By convention, sequent calculus proof trees start with the root on the bottom of the tree and we will follow this convention in our proof sketches too. Note that the proof sketches shown in the sequel are simplified substantially for improved readability. The exact number of proof steps is shown in Tab. 2.

D.1 Proof Sketch for Proposition 1: ETCS Controllability

In this section, we sketch our KeYmaera proof for the controllability result Proposition 1. The property in Proposition 1 is a biimplication (equivalence), so the proof splits into an implication from left to right and from right to left.

In the following proof sketch we use I as an abbreviation for the initial region:

$$I \equiv (m.d \geq 0 \land b > 0 \land z \leq m \land \tau.v \geq 0)$$

The essential part of the sequent proof for Proposition 1 is as follows:

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\text{close by quantifier elimination}}{\begin{array}{c} I, \mathcal{C} \vdash \forall t \geq 0 : (\forall 0 \leq \tilde{t} \leq t : \\ -1 b\tilde{t} + v \geq 0) \rightarrow \\ (\frac{1}{2}(-bt^2 + 2t\tau.v + 2\tau.p) \geq m.e \\ \rightarrow -bt + \tau.v \leq m.d) \end{array}} \quad \dfrac{\text{close by quantifier elimination}}{\begin{array}{c} I, \forall t \geq 0 : (\forall 0 \leq \tilde{t} \leq t : \\ -1 b\tilde{t} + v \geq 0) \rightarrow \\ (\frac{1}{2}(-bt^2 + 2t\tau.v + 2\tau.p) \geq m.e \\ \rightarrow -bt + \tau.v \leq m.d \vdash \mathcal{C} \end{array}}}{\begin{array}{c} I, \mathcal{C} \vdash [\tau.p' = v, \tau.v' = -b, \tau.v \geq 0] \\ (\tau.p \geq m.e \rightarrow \tau.v \leq d) \end{array}} \quad \dfrac{}{\begin{array}{c} I, [\tau.p' = v, \tau.v' = -b, \tau.v \geq 0] \\ (\tau.p \geq m.e \rightarrow \tau.v \leq d) \vdash \mathcal{C} \end{array}}}{\begin{array}{c} I \vdash ([\tau.p' = v, \tau.v' = -b, \tau.v \geq 0](\tau.p \geq m.e \rightarrow \tau.v \leq d) \leftrightarrow \\ \tau.v^2 - m.d^2 \leq 2b(m.e - \tau.p)) \end{array}}}{\begin{array}{l} m.d \geq 0 \land b > 0 \land z \leq m \land \tau.v \geq 0 \vdash \big([\tau.p' = v, \tau.v' = -b, \tau.v \geq 0] \\ \qquad (\tau.p \geq m.e \rightarrow \tau.v \leq d)\big) \leftrightarrow \\ \qquad \tau.v^2 - m.d^2 \leq 2b(m.e - \tau.p) \end{array}}}{\begin{array}{l} \vdash \quad (m.d \geq 0 \land b > 0 \land z \leq m \land \tau.v \geq 0) \rightarrow \\ \qquad \big([\tau.p' = v, \tau.v' = -b, \tau.v \geq 0](\tau.p \geq m.e \rightarrow \tau.v \leq d)\big) \\ \qquad\qquad \leftrightarrow \tau.v^2 - m.d^2 \leq 2b(m.e - \tau.p) \end{array}}$$

The proof proceeds as follows. First, the toplevel implication is handled by moving the precondition to the antecedent of the sequent. Afterwards, the biimplication, which is now the toplevel operator in the succedent, causes the proof to split into two cases that show each direction of the biimplication separately. The modality contains in both cases just the differential equation system. As it has a polynomial solution, we can handle it by solving the system. This leaves us with pure first-order formulas over real arithmetic. Both sequents can be proven to be true using quantifier elimination and the branches can be closed.

D.2 Proof Sketch for Proposition 2: RBC Controllability

In this section we illustrate how KeYmaera proves that the actions performed by the RBC preserve controllability (Proposition 2).

For proving RBC Controllability (Proposition 2) we start with

$$\vdash (m.d \geq 0 \land b > 0) \rightarrow ([(m_0 := m;\ m := *) \cup (rbc.message := emergency)](\mathcal{M} \leftrightarrow (\forall \tau (v^2 - m_0.d \leq 2b(m_0.e - \tau.p) \land \tau.v \geq 0 \land m_0.d \geq 0 \land b > 0 \rightarrow \mathcal{C}))))$$

First the implication is handled by moving the assumption into the antecedent of the sequent.

$$m.d \geq 0 \land b > 0 \vdash [(m_0 := m;\ m := *) \cup (rbc.message := emergency)](\mathcal{M} \leftrightarrow (\forall \tau (v^2 - m_0.d \leq 2b(m_0.e - \tau.p) \land \tau.v \geq 0 \land m_0.d \geq 0 \land b > 0 \rightarrow \mathcal{C}))))$$

The two different choices of the RBC (movement authority change or emergency message) produce two proof objectives which can be proven separately in the following.

For proving that the equivalence holds we have to show both directions separately, which causes the proof to split again.

$$\dfrac{\dfrac{\dfrac{(m.d \geq 0 \land b > 0) \vdash \forall m_1 (\mathcal{M}[m \mapsto m_1][m_0 \mapsto m] \leftrightarrow (\forall \tau (v^2 - m_0.d \leq 2b(m_0.e - \tau.p) \land \tau.v \geq 0 \land m_0.d \geq 0 \land b > 0 \rightarrow \mathcal{C}[m \mapsto m_1][m_0 \mapsto m])))}{(m.d \geq 0 \land b > 0) \vdash ([m_0 := m;\ m := *](\mathcal{M} \leftrightarrow (\forall \tau (v^2 - m_0.d \leq 2b(m_0.e - \tau.p) \land \tau.v \geq 0 \land m_0.d \geq 0 \land b > 0 \rightarrow \mathcal{C}))))} \quad \dfrac{(m.d \geq 0 \land b > 0) \vdash (\mathcal{M} \leftrightarrow (\forall \tau (v^2 - m_0.d \leq 2b(m_0.e - \tau.p) \land \tau.v \geq 0 \land m_0.d \geq 0 \land b > 0 \rightarrow \mathcal{C})))}{(m.d \geq 0 \land b > 0) \vdash ([rbc.message := emergency](\mathcal{M} \leftrightarrow (\forall \tau (v^2 - m_0.d \leq 2b(m_0.e - \tau.p) \land \tau.v \geq 0 \land m_0.d \geq 0 \land b > 0 \rightarrow \mathcal{C}))))}}{(m.d \geq 0 \land b > 0) \vdash ([(m_0 := m;\ m := *) \cup (rbc.message := emergency)](\mathcal{M} \leftrightarrow (\forall \tau (v^2 - m_0.d \leq 2b(m_0.e - \tau.p) \land \tau.v \geq 0 \land m_0.d \geq 0 \land b > 0 \rightarrow \mathcal{C}))))}$$

$$\vdash (m.d \geq 0 \land b > 0) \rightarrow ([(m_0 := m;\ m := *) \cup (rbc.message := emergency)](\mathcal{M} \leftrightarrow (\forall \tau (v^2 - m_0.d \leq 2b(m_0.e - \tau.p) \land \tau.v \geq 0 \land m_0.d \geq 0 \land b > 0 \rightarrow \mathcal{C}))))$$

In theory, the two remaining goals could be closed by quantifier elimination. In practice, we observed that further decomposition is necessary to obtain simpler formulas for the doubly exponential quantifier elimination procedures. These decompositions have been found automatically by KeYmaera using its Iterative Background Closure proof procedures [11, 28].

D.3 Proof Sketch for Proposition 6: Liveness

Proposition 6 states that the system is live, i.e. the train can actually move to any point on the track by appropriate RBC interaction. In this section we illustrate

how KeYmaera performs the proof for this property. We will sketch the proof step by step instead of using the tree form because of its complexity.

We show that, provided that the movement authority was chosen far enough, the train can reach a positive speed. Afterwards, knowing that the speed is positive, we show that we can find a formula (called variant, see [10]) that is true for the initial state, that always decreases during some appropriate execution of one iteration of the control loop, and that a negative variant finally implies that the train has passed goal p. To show that the ETCS system is live, we show that the following formula is valid:

$$\begin{aligned}&(\tau.v \geq 0 \land \varepsilon > 0 \land b > 0 \land A > 0) \rightarrow \\ &\quad \forall p > 0\, ((\langle m_0 := m;\ m := *;\ ?\mathcal{M};\ \tau.a := A;\ \mathrm{atp_r};\ \mathrm{drive}\rangle\, \tau.v > 0) \\ &\qquad \land (\tau.v > 0 \rightarrow \langle \tau.p := 0;\ v_0 := \tau.v\rangle \\ &\qquad \langle (m_0 := m;\ m := *;\ ?\mathcal{M};\ \tau.a := A\ \mathrm{atp_r};\ \mathrm{drive})^*\rangle\, \tau.p \geq p))\end{aligned}$$

Formally, this formula results from Proposition 6 by unrolling the loop once and applying a sequential generalization, i.e. split the two modalities by showing that the first iteration can reach a state, where $\tau.v$ is greater than zero and that this is sufficient for the second program to satisfy the postcondition.

The proof performed in KeYmaera for showing that this formula is valid now proceeds as follows.

First the implication is handled by moving the assumption into the antecedent of the sequent.

$$\begin{aligned}&(\tau.v \geq 0 \land \varepsilon > 0 \land b > 0 \land A > 0) \vdash \\ &\quad \forall p > 0\, ((\langle m_0 := m;\ m := *;\ ?\mathcal{M};\ \mathrm{spd};\ \mathrm{atp_r};\ \mathrm{drive}\rangle\, \tau.v > 0) \\ &\qquad \land (\tau.v > 0 \rightarrow \langle \tau.p := 0;\ v_0 := \tau.v\rangle \\ &\qquad \langle (m_0 := m;\ m := *;\ ?\mathcal{M};\ \tau.a := A;\ \mathrm{atp_r};\ \mathrm{drive})^*\rangle\, \tau.p \geq p))\end{aligned}$$

Now KeYmaera performs Skolemization, which means that the quantifier for p and the variable itself is replaced by a new function symbol sk_p to enable the calculus to further decompose the proof goal. The idea is to store the information that the variable was universally quantified in the fact that the type replacing it is a special Skolem function. For formal details on this technique see [10].

$$\begin{aligned}&(\tau.v \geq 0 \land \varepsilon > 0 \land b > 0 \land A > 0) \vdash \\ &\quad sk_p > 0 \rightarrow ((\langle m_0 := m;\ m := *;\ ?\mathcal{M};\ \tau.a := A;\ \mathrm{atp_r};\ \mathrm{drive}\rangle\, \tau.v > 0) \\ &\qquad \land (\tau.v > 0 \rightarrow \langle \tau.p := 0;\ v_0 := \tau.v\rangle \\ &\qquad \langle (m_0 := m;\ m := *;\ ?\mathcal{M};\ \tau.a := A;\ \mathrm{atp_r};\ \mathrm{drive})^*\rangle\, \tau.p \geq sk_p))\end{aligned}$$

Afterwards, the implication which is now the top level operator on the right side of the sequent is transformed into the sequent representation, i.e. we move the

premises to the left side of the sequent.

$$\begin{array}{l}(\tau.v \geq 0 \land \varepsilon > 0 \land b > 0 \land A > 0 \land sk_p > 0) \vdash \\ \quad ((\langle m_0 := m;\ m := *;\ ?\mathcal{M};\ \tau.a := A;\ \mathrm{atp_r};\ \mathrm{drive}\rangle\, \tau.v > 0) \\ \qquad \land (\tau.v > 0 \rightarrow \langle \tau.p := 0;\ v_0 := \tau.v\rangle \\ \qquad \langle (m_0 := m;\ m := *;\ ?\mathcal{M};\ \tau.a := A;\ \mathrm{atp_r};\ \mathrm{drive})^*\rangle\, \tau.p \geq sk_p))\end{array}$$

Now the proof splits into two independent cases along the conjunction on right side of the sequent (when proving validity of a conjunction, both conjuncts can be proven separately):

1. Positive speed is reachable (which corresponds to the first unrolled loop iteration):

$$\begin{array}{l}(\tau.v \geq 0 \land \varepsilon > 0 \land b > 0 \land A > 0 \land sk_p > 0) \vdash \\ \quad (\langle m_0 := m;\ m := *;\ ?\mathcal{M};\ \tau.a := A;\ \mathrm{atp_r};\ \mathrm{drive}\rangle\, \tau.v > 0)\end{array}$$

2. Once reached, positive speed can be kept to reach goal sk_p:

$$\begin{array}{l}(\tau.v \geq 0 \land \varepsilon > 0 \land b > 0 \land A > 0 \land sk_p > 0) \vdash \\ \qquad (\tau.v > 0 \rightarrow \langle \tau.p := 0;\ v_0 := \tau.v\rangle \\ \quad \langle (m_0 := m;\ m := *;\ ?\mathcal{M};\ \tau.a := A;\ \mathrm{atp_r};\ \mathrm{drive})^*\rangle\, \tau.p \geq sk_p)\end{array}$$

For case 1 the proof proceeds as follows:

$$\begin{array}{l}(\tau.v \geq 0 \land \varepsilon > 0 \land b > 0 \land A > 0 \land sk_p > 0) \vdash \\ \qquad (\langle m_0 := m;\ m := *;\ ?\mathcal{M};\ \tau.a := A;\ \mathrm{atp_r};\ \mathrm{drive}\rangle\, \tau.v > 0)\end{array}$$

The RBC part is transformed into first order formulas, by applying the assignments and transforming the test ($?\mathcal{M}$) into a conjunction with the following train controller modality. This test in a diamond property gives a conjunction by the semantics, which says that test formulas in diamond modalities have to hold true, otherwise there cannot be a run of the program (because every prefix of a run would end up being blocked by reaching a failed test statement).

$$\begin{array}{l}(\tau.v \geq 0 \land \varepsilon > 0 \land b > 0 \land A > 0 \land sk_p > 0) \vdash \\ \quad \exists m_1 (m.d^2 - m_1.d^2 \leq 2b(m_1.e - m.e) \land m.d \geq 0 \land m_1.d \geq 0) \land \\ \qquad \langle \tau.a := A; \mathrm{atp_r};\ \mathrm{drive}\rangle\, \tau.v > 0\end{array}$$

Now we drop the existential quantifiers and mark the previously bound variables as free variables (called *Meta-Variables* in KeYmaera). This marking is used to reintroduce the quantifiers later on. Again note that the details about

the quantifier handling by Skolemization and free variables for real arithmetic as well as their soundness are explained in [10].

$$(\tau.v \geq 0 \land \varepsilon > 0 \land b > 0 \land A > 0 \land sk_p > 0) \vdash \\ (m.d^2 - m_1.d^2 \leq 2b(m_1.e - m.e) \land m.d \geq 0 \land m_1.d \geq 0) \land \\ \langle \tau.a := A;\ \mathrm{atp_r};\ \mathrm{drive} \rangle\, \tau.v > 0$$

Here the proof splits again into the different cases of the conjunction in the succedent:

1.1 RBC constraint ($\mathcal{M}$) is feasible:

$$(\tau.v \geq 0 \land \varepsilon > 0 \land b > 0 \land A > 0 \land sk_p > 0) \vdash \\ (m.d^2 - m_1.d^2 \leq 2b(m_1.e - m.e) \land m.d \geq 0 \land m_1.d \geq 0)$$

1.2 Positive speed is reachable:

$$(\tau.v \geq 0 \land \varepsilon > 0 \land b > 0 \land A > 0 \land sk_p > 0) \vdash \\ \langle \tau.a := A;\ \mathrm{atp_r};\ \mathrm{drive} \rangle\, \tau.v > 0$$

For the first case (1.1) we cannot apply any useful rule, because it is blocked by the second goal, i.e. we cannot eliminate the quantifiers, as the Meta-Variables occur on (1.2) too. Therefore, we go on decomposing the second goal (1.2):

$$(\tau.v \geq 0 \land \varepsilon > 0 \land b > 0 \land A > 0 \land sk_p > 0) \vdash \\ \langle \tau.a := A;\ \mathrm{atp_r};\ \mathrm{drive} \rangle\, \tau.v > 0$$

First we expand the macro $\mathrm{atp_r}$:

$$(\tau.v \geq 0 \land \varepsilon > 0 \land b > 0 \land A > 0 \land sk_p > 0) \vdash \\ \langle \tau.a := A \rangle \left\langle SB := \frac{\tau.v^2 - m_1.d^2}{2b} + \left(\frac{A}{b} + 1\right)\left(\frac{A}{2}\varepsilon^2 + \varepsilon\, \tau.v\right);\right. \\ \left. \langle \text{if } (m_1.e - \tau.p \leq SB) \text{ then } \tau.a := -b \text{ fi};\ \mathrm{drive} \rangle\, \tau.v > 0 \right.$$

Then we apply the assignment to SB:

$$(\tau.v \geq 0 \land \varepsilon > 0 \land b > 0 \land A > 0 \land sk_p > 0) \vdash \\ \langle \tau.a := A \rangle \left\langle \text{if } \left(m_1.e - \tau.p \leq \frac{\tau_2.v^2 - m_1.d^2}{2b} + \left(\frac{A}{b} + 1\right)\left(\frac{A}{2}\varepsilon^2 + \varepsilon\, \tau.v\right)\right)\right. \\ \left. \text{then } \tau.a := -b \right\rangle \langle \mathrm{drive} \rangle\, \tau.v > 0$$

Next, we can split up the formula into two cases, one where the test condition in the if-statement holds, and one where it is false (for the diamond case, these two formulas are connected by a disjunction, because there is a transition if either case works):

$$
\begin{array}{l}
(\tau.v \geq 0 \land \varepsilon > 0 \land b > 0 \land A > 0 \land sk_p > 0) \vdash \\
\quad (m_1.e - \tau.p \leq \dfrac{\tau.v^2 - m_1.d^2}{2b} + \left(\dfrac{A}{b} + 1\right)\left(\dfrac{A}{2}\varepsilon^2 + \varepsilon\ \tau.v\right) \\
\qquad \land \langle \tau.a := -b \rangle \langle \text{drive} \rangle \tau.v > 0) \\
\qquad\qquad \lor \\
\quad (m_1.e - \tau.p > \dfrac{\tau.v^2 - m_1.d^2}{2b} + \left(\dfrac{A}{b} + 1\right)\left(\dfrac{A}{2}\varepsilon^2 + \varepsilon\ \tau.v\right) \\
\qquad\qquad \land \langle \tau.a := A \rangle \langle \text{drive} \rangle \tau.v > 0)
\end{array}
$$

Now we expand the differential equation system drive and apply the effect of the assignment to the acceleration $\tau.a$:

$$
\begin{array}{l}
(\tau.v \geq 0 \land \varepsilon > 0 \land b > 0 \land A > 0 \land sk_p > 0) \vdash \\
\quad (m_1.e - \tau.p \leq \dfrac{\tau.v^2 - m_1.d^2}{2b} + \left(\dfrac{A}{b} + 1\right)\left(\dfrac{A}{2}\varepsilon^2 + \varepsilon\ \tau.v\right) \\
\land \langle t := 0;\ (\tau.p' = \tau.v \land \tau.v' = -b \land t' = 1 \land \tau.v \geq 0 \land t \leq \varepsilon) \rangle \tau.v > 0) \\
\qquad\qquad \lor \\
\quad (m_1.e - \tau.p > \dfrac{\tau.v^2 - m_1.d^2}{2b} + \left(\dfrac{A}{b} + 1\right)\left(\dfrac{A}{2}\varepsilon^2 + \varepsilon\ \tau.v\right) \\
\land \langle t := 0;\ (\tau.p' = \tau.v \land \tau.v' = A \land t' = 1 \land \tau.v \geq 0 \land t \leq \varepsilon) \rangle \tau.v > 0)
\end{array}
$$

Now we apply rules for handling the systems of differential equations.

$$
\begin{array}{l}
(\tau.v \geq 0 \land \varepsilon > 0 \land b > 0 \land A > 0 \land sk_p > 0) \vdash \\
\quad (m_1.e - \tau.p \leq \dfrac{\tau.v^2 - m_1.d^2}{2b} + \left(\dfrac{A}{b} + 1\right)\left(\dfrac{A}{2}\varepsilon^2 + \varepsilon\ \tau.v\right) \\
\land (\exists t_1 \geq 0\ (\forall 0 \leq \tilde{t_1} \leq t_1\ (-b\tilde{t_1} + \tau.v \geq 0 \land \tilde{t_1} \leq \varepsilon) \land -bt_1 + \tau.v > 0)) \\
\qquad\qquad \lor \\
\quad (m_1.e - \tau.p > \dfrac{\tau.v^2 - m_1.d^2}{2b} + \left(\dfrac{A}{b} + 1\right)\left(\dfrac{A}{2}\varepsilon^2 + \varepsilon\ \tau.v\right) \\
\land (\exists t_1 \geq 0\ (\forall 0 \leq \tilde{t_1} \leq t_1\ (A\tilde{t_1} + \tau.v \geq 0 \land \tilde{t_1} \leq \varepsilon) \land At_1 + \tau.v > 0)))
\end{array}
$$

At this point we merge this goal with goal (1.1) again and reintroduce the existential quantifier (see [10] for formal details), leading to the following formula,

which can be proven valid by quantifier elimination.

$$
\begin{array}{l}
(\tau.v \geq 0 \land \varepsilon > 0 \land b > 0 \land A > 0 \land sk_p > 0) \vdash \\
\quad \exists m_1((m.d^2 - m_1.d^2 \leq 2b(m_1.e - m.e) \land m.d \geq 0 \land m_1.d \geq 0) \lor \\
\quad\quad (m_1.e - \tau.p \leq \dfrac{\tau.v^2 - m_1.d^2}{2b} + \left(\dfrac{A}{b} + 1\right)\left(\dfrac{A}{2}\varepsilon^2 + \varepsilon\ \tau.v\right) \\
\quad \land (\exists t_1 \geq 0\ (\forall 0 \leq \tilde{t_1} \leq t_1\ (-b\tilde{t_1} + \tau.v \geq 0 \land \tilde{t_1} \leq \varepsilon) \land -bt_1 + \tau.v > 0)) \\
\quad\quad \lor \\
\quad\quad (m_1.e - \tau.p > \dfrac{\tau.v^2 - m_1.d^2}{2b} + \left(\dfrac{A}{b} + 1\right)\left(\dfrac{A}{2}\varepsilon^2 + \varepsilon\ \tau.v\right) \\
\quad \land (\exists t_1 \geq 0\ (\forall 0 \leq \tilde{t_1} \leq t_1\ (A\tilde{t_1} + \tau.v \geq 0 \land \tilde{t_1} \leq \varepsilon) \land At_1 + \tau.v > 0)))))
\end{array}
$$

After closing this goal we can go on with the remaining open goal (2) from the first split. We have now shown that the train can reach a positive speed. Now goal 2 will show that starting with a positive speed the train can reach every position on the track by appropriate movement authority extension.

$$
\begin{array}{l}
(\tau.v \geq 0 \land \varepsilon > 0 \land b > 0 \land A > 0 \land sk_p > 0) \vdash \\
\quad\quad (\tau.v > 0 \rightarrow \langle \tau.p := 0;\ v_0 := \tau.v \rangle \\
\quad \langle (m_0 := m;\ m := *;\ ?\mathcal{M};\ \tau.a := A;\ \mathrm{atp_r};\ \mathrm{drive})^* \rangle\ \tau.p \geq sk_p)
\end{array}
$$

First we handle the implication in the succedent by moving its left side to the antecedent.

$$
\begin{array}{l}
(\tau.v > 0 \land \varepsilon > 0 \land b > 0 \land A > 0 \land sk_p > 0) \vdash \\
\quad\quad (\langle \tau.p := 0;\ v_0 := \tau.v \rangle \\
\quad \langle (m_0 := m;\ m := *;\ ?\mathcal{M};\ \tau.a := A;\ \mathrm{atp_r};\ \mathrm{drive})^* \rangle\ \tau.p \geq sk_p)
\end{array}
$$

For proving that there is a run of this system such that $\tau.p \geq sk_p$ holds, we show that the loop satisfies the variant $\varphi(n) \equiv \tau.p + n\,\varepsilon\,v_0 \geq sk_p \land \tau.v \geq v_0$, which expresses that sk_p is within reach of at most n cycles of duration ε at speed v_0 and that the speed $\tau.v$ does not decrease below the speed v_0 reached after the first cycle.

For variant proofs [10], we have to show that the variant is satisfiable, that for some $n \leq 0$ the post condition holds and that the variant decreases during each iteration of the loop.

2.1 The variant is satisfiable:

$$
\begin{array}{l}
(\tau.v > 0 \land \varepsilon > 0 \land b > 0 \land A > 0 \land sk_p > 0) \vdash \\
\quad\quad (\langle \tau.p := 0;\ v_0 := \tau.v \rangle\ \exists n\ (\tau.p + n\,\varepsilon\,v_0 \geq sk_p \land \tau.v \geq v_0)
\end{array}
$$

By applying the assignments we get:

$$(\tau.v > 0 \land \varepsilon > 0 \land b > 0 \land A > 0 \land sk_p > 0) \vdash \exists n\,(n\,\varepsilon\,\tau.v \geq sk_p \land \tau.v \geq \tau.v)$$

This can be closed by quantifier elimination.

2.2 Non-positive distance of the variant implies that the train has passed the point p.

$$\begin{array}{l}(\tau.v > 0 \land \varepsilon > 0 \land b > 0 \land A > 0 \land sk_p > 0) \vdash \\ \quad \langle \tau.p := 0;\ v_0 := \tau.v\rangle\, \forall \tau_1.v \forall \tau_1.p\, \langle \tau.p := \tau_1.p;\ \tau.v := \tau_1.v\rangle \\ \quad\quad ((\exists n{\leq}0\,(\,\tau.p + n\,\varepsilon\,\tau.v \geq sk_p \land \tau.v \geq v_0)) \rightarrow \tau.p \geq sk_p)\end{array}$$

Applying the assignments we get:

$$\begin{array}{l}(\tau.v > 0 \land \varepsilon > 0 \land b > 0 \land A > 0 \land sk_p > 0) \vdash \\ \quad \forall \tau_1.v \forall \tau_1.p((\exists n{\leq}0\,(\,\tau_1.p + n\,\varepsilon\,\tau_1.v \geq sk_p \land \tau_1.v \geq \tau.v)) \rightarrow \tau_1.p \geq sk_p)\end{array}$$

This, again, can be closed easily by quantifier elimination.

2.3 Variant progress: The hard part is showing that the variant can always decrease during each iteration of the loop so that the train successively reaches its goal.

$$\begin{array}{l}(\tau.v > 0 \land \varepsilon > 0 \land b > 0 \land A > 0 \land sk_p > 0) \vdash \\ \quad (\langle \tau.p := 0;\ v_0 := \tau.v\rangle\, \forall \tau, SB, t, m\, \forall n \geq 0\,(\tau.p + n\,\varepsilon\,v_0 \geq sk_p \rightarrow \\ \quad\quad \langle m_0 := m;\ m := *;\ ?\mathcal{M};\ \tau.a := A;\ \mathrm{atp_r};\ \mathrm{drive}\rangle \\ \quad\quad\quad (\tau.p + (n-1)\,\varepsilon\,v_0 \geq sk_p \land \tau.v \geq v_0))\end{array}$$

Expanding the macros this reads:

$$\begin{array}{l}(\tau.v > 0 \land \varepsilon > 0 \land b > 0 \land A > 0 \land sk_p > 0) \vdash \\ \quad (\forall \tau_2, SB, t, m, m_0\, \forall n \geq 0\,(\tau_2.p + n\,\varepsilon\,\tau.v \geq sk_p \rightarrow \\ \quad\quad \langle m_0 := m;\ m := *\rangle \\ \quad\quad \left\langle ?m_0.d^2 - m.d^2 \leq 2b(m.e - m_0.e) \land m_0.d \geq 0 \land m.d \geq 0)\right\rangle \\ \quad\quad \left\langle \tau_2.a := A;\ SB := \frac{\tau_2.v^2 - m_2.d^2}{2b} + \left(\frac{A}{b} + 1\right)\left(\frac{A}{2}\varepsilon^2 + \varepsilon\,\tau.v\right)\right\rangle \\ \quad\quad \langle \text{if } (m_1.e - \tau_2.p \leq SB) \text{ then } \tau.a := -b \text{ fi}\rangle \\ \quad\quad \langle t := 0;\ (\tau_2.p' = \tau_2.v \land \tau_2.v' = \tau_2.a \land t' = 1 \land \tau.v \geq 0 \land t \leq \varepsilon)\rangle \\ \quad\quad\quad (\tau_2.p + (n-1)\,\varepsilon\,\tau.v \geq sk_p \land \tau_2.v \geq \tau.v))\end{array}$$

Now we perform Skolemization, which replaces the quantifier over the train variables by Skolem function symbols sk_v etc., one for each component. The

same is done for n, where $\forall n \geq 0$ becomes $sk_n \geq 0$.

$$
\begin{array}{l}
(\tau.v > 0 \land \varepsilon > 0 \land b > 0 \land A > 0 \land sk_p > 0) \vdash \\
\quad (\langle \tau_2.p := sk_{p_2};\ \tau_2.v := sk_v;\ \tau_2.a := sk_a;\ SB := sk_{SB};\ t := sk_t \rangle \\
\quad \langle m.e := sk_{m.e};\ m.d := sk_{m.d};\ m_0.e := sk_{m_0.e};\ m_0.d := sk_{m_0.d} \rangle \\
\quad\quad sk_n \geq 0 \rightarrow (\tau_2.p + sk_n\, \varepsilon\, \tau.v \geq sk_p \rightarrow \langle m_0 := m;\ m := * \rangle \\
\quad\quad \left\langle ?m_0.d^2 - m.d^2 \leq 2b(m.e - m_0.e) \land m_0.d \geq 0 \land m.d \geq 0) \right\rangle \\
\quad \left\langle \tau_2.a := A;\ SB := \dfrac{\tau_2.v^2 - m_2.d^2}{2b} + \left(\dfrac{A}{b} + 1\right)\left(\dfrac{A}{2}\varepsilon^2 + \varepsilon\ \tau.v\right) \right\rangle \\
\quad\quad \langle \text{if } (m_1.e - \tau_2.p \leq SB) \text{ then } \tau.a := -b \text{ fi} \rangle \\
\quad \langle t := 0;\ (\tau_2.p' = \tau_2.v \land \tau_2.v' = \tau_2.a \land t' = 1 \land \tau.v \geq 0 \land t \leq \varepsilon) \rangle \\
\quad\quad (\tau_2.p + (sk_n - 1)\, \varepsilon\, \tau.v \geq sk_p \land \tau_2.v \geq \tau.v))
\end{array}
$$

Afterwards, we successively apply the assignments:

$$
\begin{array}{l}
(\tau.v > 0 \land \varepsilon > 0 \land b > 0 \land A > 0 \land sk_p > 0) \vdash \\
\quad\quad (sk_n \geq 0 \rightarrow (sk_{\tau_2.p} + sk_n\, \varepsilon\, \tau.v \geq sk_p \rightarrow \\
\quad \langle \tau_2.p := sk_{p_2};\ \tau_2.v := sk_v;\ \tau_2.a := sk_a;\ SB := sk_{SB};\ t := sk_t \rangle \\
\quad \langle m.e := sk_{m.e};\ m.d := sk_{m.d};\ m_0.e := sk_{m_0.e};\ m_0.d := sk_{m_0.d} \rangle \\
\quad\quad \langle m_0 := m;\ m := * \rangle \\
\quad\quad \left\langle ?m_0.d^2 - m.d^2 \leq 2b(m.e - m_0.e) \land m_0.d \geq 0 \land m.d \geq 0) \right\rangle \\
\quad \left\langle \tau_2.a := A;\ SB := \dfrac{\tau_2.v^2 - m_2.d^2}{2b} + \left(\dfrac{A}{b} + 1\right)\left(\dfrac{A}{2}\varepsilon^2 + \varepsilon\ \tau.v\right) \right\rangle \\
\quad\quad \langle \text{if } (m_1.e - \tau_2.p \leq SB) \text{ then } \tau.a := -b \text{ fi} \rangle \\
\quad \langle t := 0;\ (\tau_2.p' = \tau_2.v \land \tau_2.v' = \tau_2.a \land t' = 1 \land \tau.v \geq 0 \land t \leq \varepsilon) \rangle \\
\quad\quad (\tau_2.p + (sk_n - 1)\, \varepsilon\, \tau.v \geq sk_p \land \tau_2.v \geq \tau.v))
\end{array}
$$

Next, we move the premises of the implication to the antecedent.

$$
\begin{array}{l}
\tau.v > 0 \land \varepsilon > 0 \land b > 0 \land A > 0 \land sk_p > 0 \land sk_n \geq 0 \land sk_{\tau_2.p} + sk_n\, \varepsilon\, \tau.v \geq sk_p \vdash \\
\quad \langle \tau_2.p := sk_{p_2};\ \tau_2.v := sk_v;\ \tau_2.a := sk_a;\ SB := sk_{SB};\ t := sk_t \rangle \\
\quad\quad \langle m.e := sk_{m.e};\ m.d := sk_{m.d};\ m_0.e := sk_{m.e};\ m_0.d := sk_{m.d} \rangle \\
\quad\quad \exists m_1 \left\langle ?m_0.d^2 - m_1.d^2 \leq 2b(m_1.e - m_0.e) \land m_0.d \geq 0 \land m_1.d \geq 0 \right\rangle \\
\quad\quad \left\langle \tau_2.a := A;\ SB := \dfrac{\tau_2.v^2 - m_1.d^2}{2b} + \left(\dfrac{A}{b} + 1\right)\left(\dfrac{A}{2}\varepsilon^2 + \varepsilon\ \tau.v\right) \right\rangle \\
\quad\quad\quad \langle \text{if } (m_1.e - \tau_2.p \leq SB) \text{ then } \tau.a := -b \text{ fi} \rangle \\
\quad\quad \langle t := 0;\ (\tau_2.p' = \tau_2.v \land \tau_2.v' = \tau_2.a \land t' = 1 \land \tau.v \geq 0 \land t \leq \varepsilon) \rangle \\
\quad\quad\quad (\tau_2.p + (sk_n - 1)\, \varepsilon\, \tau.v \geq sk_p \land \tau_2.v \geq \tau.v)
\end{array}
$$

Now again we remove the existential quantifier and store the information, that m_1 is a Meta-Variable. This information will later on be used to remember that m_1 stands for an existential quantifier. For more details on the handling of quantifiers we refer the reader to [10].

$$\begin{array}{l}\tau.v > 0 \land \varepsilon > 0 \land b > 0 \land A > 0 \land sk_p > 0 \land sk_n \geq 0 \land sk_{\tau_2.p} + sk_n\,\varepsilon\,\tau.v \geq sk_p \vdash \\ \quad \langle \tau_2.p := sk_{p_2};\ \tau_2.v := sk_v;\ \tau_2.a := sk_a;\ SB := sk_{SB};\ t := sk_t\rangle \\ \quad \langle m.e := sk_{m.e};\ m.d := sk_{m.d};\ m_0.e := sk_{m.e};\ m_0.d := sk_{m.d}\rangle \\ \quad \left\langle ?m_0.d^2 - m_1.d^2 \leq 2b(m_1.e - m_0.e) \land m_0.d \geq 0 \land m_1.d \geq 0\right\rangle \\ \left\langle \tau_2.a := A;\ SB := \dfrac{\tau_2.v^2 - m_1.d^2}{2b} + \left(\dfrac{A}{b} + 1\right)\left(\dfrac{A}{2}\varepsilon^2 + \varepsilon\,\tau.v\right)\right\rangle \\ \quad \langle \text{if } (m_1.e - \tau_2.p \leq SB) \text{ then } \tau.a := -b \text{ fi}\rangle \\ \quad \langle t := 0;\ (\tau_2.p' = \tau_2.v \land \tau_2.v' = \tau_2.a \land t' = 1 \land \tau.v \geq 0 \land t \leq \varepsilon)\rangle \\ \quad\quad (\tau_2.p + (sk_n - 1)\,\varepsilon\,\tau.v \geq sk_p \land \tau_2.v \geq \tau.v)\end{array}$$

The test statement is transformed into a conjunction and after applying the assignments we get:

$$\begin{array}{l}\tau.v > 0 \land \varepsilon > 0 \land b > 0 \land A > 0 \land sk_p > 0 \land sk_n \geq 0 \land sk_{\tau_2.p} + sk_n\,\varepsilon\,\tau.v \geq sk_p \vdash \\ \quad (sk_{m.d}^2 - sk_{m_1.d}^2 \leq 2b(m_1.e - sk_{m.e}) \land sk_{m.d} \geq 0 \land m_1.d \geq 0) \land \\ \quad\quad \langle \tau_2.p := sk_{p_2};\ \tau_2.v := sk_v;\ \tau_2.a := A\rangle \\ \langle \text{if } \left(m_1.e - \tau_2.p \leq \dfrac{\tau_2.v^2 - m_1.d^2}{2b} + \left(\dfrac{A}{b} + 1\right)\left(\dfrac{A}{2}\varepsilon^2 + \varepsilon\,\tau.v\right)\right) \\ \quad\quad \text{then } \tau.a := -b\rangle \\ \quad \langle t := 0;\ (\tau_2.p' = \tau_2.v \land \tau_2.v' = \tau_2.a \land t' = 1 \land \tau.v \geq 0 \land t \leq \varepsilon)\rangle \\ \quad\quad (\tau_2.p + (sk_n - 1)\,\varepsilon\,\tau.v \geq sk_p \land \tau_2.v \geq \tau.v)\end{array}$$

Now the proof splits along this fresh conjunction, which is now the toplevel operator in the succedent.

2.3.1 The first case does not contain any program statements

$$\begin{array}{l}\tau.v > 0 \land \varepsilon > 0 \land b > 0 \land A > 0 \land sk_p > 0 \land sk_n \geq 0 \land sk_{\tau_2.p} + sk_n\,\varepsilon\,\tau.v \geq sk_p \vdash \\ \quad (sk_{m.d}^2 - sk_{m_1.d}^2 \leq 2b(m_1.e - sk_{m.e}) \land sk_{m.d} \geq 0 \land m_1.d \geq 0)\end{array}$$

but we cannot apply any useful rules to this goal at the moment as it it not a valid formula. As it it contains variables that are marked as Meta-Variables, we still got a chance to close this goal using informations obtained on other branches by reuniteing the branches under an existential quantifier later on. Therefore, we go on decomposing the second goal.

2.3.2 The second case still contains program structure. Therefore, we can go on with structural decomposition.

$$\begin{array}{c}\tau.v > 0 \land \varepsilon > 0 \land b > 0 \land A > 0 \land sk_p > 0 \land sk_n \geq 0 \land sk_{\tau_2.p} + sk_n\,\varepsilon\,\tau.v \geq sk_p \vdash \\ \langle \tau_2.p := sk_{p_2};\ \tau_2.v := sk_v;\ \tau_2.a := A \rangle \\ \left\langle \text{if } \left(m_1.e - \tau.p \leq \frac{\tau_2.v^2 - m_1.d^2}{2b} + \left(\frac{A}{b} + 1\right)\left(\frac{A}{2}\varepsilon^2 + \varepsilon\,\tau.v\right)\right) \right. \\ \left. \text{then } \tau.a := -b \right\rangle \\ \langle t := 0;\ (\tau_2.p' = \tau_2.v \land \tau_2.v' = \tau_2.a \land t' = 1 \land \tau.v \geq 0 \land t \leq \varepsilon) \rangle \\ (\tau_2.p + (sk_n - 1)\,\varepsilon\,\tau.v \geq sk_p \land \tau_2.v \geq \tau.v)\end{array}$$

Here the proof splits depending on the remaining distance to the end of the MA as this is the condition in the toplevel if statement. The split accounts to the fact that the semantic of the if statement in the diamond case is a disjunction of conjunctions which would lead to 4 different cases. One can be closed immediately, as it states that the test condition is true or false. Another one states that either the post condition holds immediately or the conditional program reaches a state where the postcondition is valid. This is subsumed in the two following cases:

2.3.2.1 For the first case we assume that the test condition holds and show that the execution of $\tau.a := b$ reaches a state where its postcondition is valid.

$$\begin{array}{c}\tau.v > 0 \land \varepsilon > 0 \land b > 0 \land A > 0 \land sk_p > 0 \land sk_n \geq 0 \land sk_{\tau_2.p} + sk_n\,\varepsilon\,\tau.v \geq sk_p \\ \land\, m_1.e - \tau.p > \frac{\tau_2.v^2 - m_1.d^2}{2b} + \left(\frac{A}{b} + 1\right)\left(\frac{A}{2}\varepsilon^2 + \varepsilon\,\tau.v\right) \vdash \\ \langle \tau_2.p := sk_{p_2};\ \tau_2.v := sk_v;\ \tau_2.a := A \rangle \\ \langle t := 0;\ (\tau_2.p' = \tau_2.v \land \tau_2.v' = \tau_2.a \land t' = 1 \land \tau.v \geq 0 \land t \leq \varepsilon) \rangle \\ (\tau_2.p + (sk_n - 1)\,\varepsilon\,\tau.v \geq sk_p \land \tau_2.v \geq \tau.v)\end{array}$$

First, we apply the pending assignments:

$$\begin{array}{c}\tau.v > 0 \land \varepsilon > 0 \land b > 0 \land A > 0 \land sk_p > 0 \land sk_n \geq 0 \land sk_{\tau_2.p} + sk_n\,\varepsilon\,\tau.v \geq sk_p \\ \land\, m_1.e - \tau.p > \frac{\tau_2.v^2 - m_1.d^2}{2b} + \left(\frac{A}{b} + 1\right)\left(\frac{A}{2}\varepsilon^2 + \varepsilon\,\tau.v\right) \vdash \\ \langle \tau_2.p := sk_{p_2};\ \tau_2.v := sk_v \rangle \\ \langle t := 0;\ (\tau_2.p' = \tau_2.v \land \tau_2.v' = A \land t' = 1 \land \tau.v \geq 0 \land t \leq \varepsilon) \rangle \\ (\tau_2.p + (sk_n - 1)\,\varepsilon\,\tau.v \geq sk_p \land \tau_2.v \geq \tau.v)\end{array}$$

Afterwards we use the solution of the differential equation system to get rid of the remaining modality.

$$\begin{array}{c}\tau.v > 0 \land \varepsilon > 0 \land b > 0 \land A > 0 \land sk_p > 0 \land sk_n \geq 0 \land sk_{\tau_2.p} + sk_n\,\varepsilon\,\tau.v \geq sk_p \\ \land\, m_1.e - \tau.p > \frac{\tau_2.v^2 - m_1.d^2}{2b} + \left(\frac{A}{b}+1\right)\left(\frac{A}{2}\varepsilon^2 + \varepsilon\,\tau.v\right) \vdash \\ \exists t \geq 0\,(\forall 0 \leq \tilde{t} \leq t\; A\tilde{t} + sk_v \geq 0 \land \tilde{t} \leq \varepsilon) \land \\ (\frac{1}{2}(At^2 + 2tsk_v + 2sk_{p_2}) + (sk_n - 1)\,\varepsilon\,\tau.v \geq sk_p \land At + sk_v \geq \tau.v)\end{array}$$

This goal can be closed by hiding the constraint $(m_1.e - \tau.p > \frac{\tau_2.v^2 - m_1.d^2}{2b} + \left(\frac{A}{b}+1\right)\left(\frac{A}{2}\varepsilon^2 + \varepsilon\,\tau.v\right))$ and applying quantifier elimination as if all variables were universally quantified. This is sound, because a universal quantifier is a sound overapproximation of an existential quantifier, see [10].

2.3.2.2 In this case we show that the postcondition is already satisfied if the test condition does not hold.

$$\begin{array}{c}\tau.v > 0 \land \varepsilon > 0 \land b > 0 \land A > 0 \land sk_p > 0 \land sk_n \geq 0 \land sk_{\tau_2.p} + sk_n\,\varepsilon\,\tau.v \geq sk_p \\ \land\, m_1.e - \tau.p \leq \frac{\tau_2.v^2 - m_1.d^2}{2b} + \left(\frac{A}{b}+1\right)\left(\frac{A}{2}\varepsilon^2 + \varepsilon\,\tau.v\right) \vdash \\ \langle \tau_2.p := sk_{p_2};\; \tau_2.v := sk_v;\; \tau_2.a := -b \rangle \\ \langle t := 0;\; (\tau_2.p' = \tau_2.v \land \tau_2.v' = \tau_2.a \land t' = 1 \land \tau.v \geq 0 \land t \leq \varepsilon) \rangle \\ (\tau_2.p + (sk_n - 1)\,\varepsilon\,\tau.v \geq sk_p \land \tau_2.v \geq \tau.v)\end{array}$$

Again, we first apply the pending assignments:

$$\begin{array}{c}\tau.v > 0 \land \varepsilon > 0 \land b > 0 \land A > 0 \land sk_p > 0 \land sk_n \geq 0 \land sk_{\tau_2.p} + sk_n\,\varepsilon\,\tau.v \geq sk_p \\ \land\, m_1.e - \tau.p \leq \frac{\tau_2.v^2 - m_1.d^2}{2b} + \left(\frac{A}{b}+1\right)\left(\frac{A}{2}\varepsilon^2 + \varepsilon\,\tau.v\right) \vdash \\ \langle \tau_2.p := sk_{p_2};\; \tau_2.v := sk_v \rangle \\ \langle t := 0;\; (\tau_2.p' = \tau_2.v \land \tau_2.v' = -b \land t' = 1 \land \tau.v \geq 0 \land t \leq \varepsilon) \rangle \\ (\tau_2.p + (sk_n - 1)\,\varepsilon\,\tau.v \geq sk_p \land \tau_2.v \geq \tau.v)\end{array}$$

Afterwards we use the solution of the differential equation system to get rid of the remaining modality.

$$\begin{array}{c}\tau.v > 0 \land \varepsilon > 0 \land b > 0 \land A > 0 \land sk_p > 0 \land sk_n \geq 0 \land sk_{\tau_2.p} + sk_n\,\varepsilon\,\tau.v \geq sk_p \\ \land\, m_1.e - \tau.p > \frac{\tau_2.v^2 - m_1.d^2}{2b} + \left(\frac{A}{b}+1\right)\left(\frac{A}{2}\varepsilon^2 + \varepsilon\,\tau.v\right) \vdash \\ \exists t \geq 0\,(\forall 0 \leq \tilde{t} \leq t\; -b\tilde{t} + sk_v \geq 0 \land \tilde{t} \leq \varepsilon) \land \\ (\frac{1}{2}(-bt^2 + 2tsk_v + 2sk_{p_2}) + (sk_n - 1)\,\varepsilon\,\tau.v \geq sk_p \land -bt + sk_v \geq \tau.v)\end{array}$$

Now we instantiate t with 0, since the train should already have passed the point p when approaching the end of the movement authority (for appropriate RBC choices). This gives:

$$\begin{aligned}\tau.v > 0 \land \varepsilon > 0 \land b > 0 \land A > 0 \land sk_p > 0 \land sk_n \geq 0 \land sk_{\tau_2.p} + sk_n\,\varepsilon\,\tau.v \geq sk_p \\ \land\, m_1.e - \tau.p > \frac{\tau_2.v^2 - m_1.d^2}{2b} + \left(\frac{A}{b}+1\right)\left(\frac{A}{2}\varepsilon^2 + \varepsilon\,\tau.v\right) \vdash \\ (sk_v \geq 0 \land 0 \leq \varepsilon) \land \\ ((sk_{p_2}) + (sk_n - 1)\,\varepsilon\,\tau.v \geq sk_p \land sk_v \geq \tau.v))\end{aligned}$$

Now KeYmaera recombines this goal with goal 2.3.1 which reads

$$\begin{aligned}\tau.v > 0 \land \varepsilon > 0 \land b > 0 \land A > 0 \land sk_p > 0 \land sk_n \geq 0 \land sk_{\tau_2.p} + sk_n\,\varepsilon\,\tau.v \geq sk_p \vdash \\ (sk^2_{m.d} - sk^2_{m_1.d} \leq 2b(m_1.e - sk_{m.e}) \land sk_{m.d} \geq 0 \land m_1.d \geq 0)\end{aligned}$$

and reintroduces the existential quantifiers over m_1 (which is sound because m_1 is a Meta-Variable [10]) such that we get:

$$\begin{aligned}\tau.v > 0 \land \varepsilon > 0 \land b > 0 \land A > 0 \land sk_p > 0 \land sk_n \geq 0 \land sk_{\tau_2.p} + sk_n\,\varepsilon\,\tau.v \geq sk_p \vdash \\ \exists m_1 (m_1.e - \tau.p > \frac{\tau_2.v^2 - m_1.d^2}{2b} + \left(\frac{A}{b}+1\right)\left(\frac{A}{2}\varepsilon^2 + \varepsilon\,\tau.v\right) \rightarrow \\ (sk_v \geq 0 \land 0 \leq \varepsilon) \land ((sk_{p_2}) + (sk_n - 1)\,\varepsilon\,\tau.v \geq sk_p \land sk_v \geq \tau.v)) \\ \land\, (sk^2_{m.d} - sk^2_{m_1.d} \leq 2b(m_1.e - sk_{m.e}) \land sk_{m.d} \geq 0 \land m_1.d \geq 0)\end{aligned}$$

KeYmaera closes this last remaining goal by quantifier elimination and thus finishes the proof for Proposition 6. Therefore KeYmaera has shown that the ETCS system is not over-constrained but live.

D.4 Proof Sketch for Proposition 8: Safety with PI Controller

The Proposition 8 states that the ETCS is safe when controlled by a PI controller. In the following we sketch how KeYmaera proves this property. This can be proven similar to the safety proof for our simplified controller. The main difference is that the system of differential equations cannot be solved using polynomial arithmetic. Instead we overapproximate it conservatively using a system that allows more behavior but can be handled more easily and use differential invariants [24, 23] to reason about its traces. Generally, PI controllers and PID controllers do not have polynomial solutions, which makes verification very difficult. We overcome this challenge using differential invariants.

We start with the following formula which is a formalization of Proposition 8:

$$\begin{aligned}\vdash \mathcal{C} \rightarrow \Big[\big((atp_{\mathrm{r}};\ t := 0;\ \text{if } (\tau.a = -b) \text{ then } \mathcal{I} \text{ else } \mathcal{P} \land t' = 1 \land t \leq \varepsilon \text{ fi}) \\ \cup\, rbc_{\mathrm{r}}\big)^*\Big]\,(\tau.p \geq m.e \rightarrow \tau.v \leq m.d)\end{aligned}$$

This system uses PI controller ($\mathcal{P}$) for speed regulation according to the recommended speed $m.r$ unless *atp* issued an emergency braking request ($\tau.a = b$) where the PI ($\mathcal{P}$) for recommended speed supervision is disabled and the train brakes according to differential equation system ($\mathcal{I}$) instead. Further note that *spd* can be removed from the system in this case, as speed supervision for recommended speed $m.r$ is taken care by the PI ($\mathcal{P}$) rather than the discrete *spd* controller. See [2] for details on train control.

First the implication is translated into the sequent structure.

$$\mathcal{C} \vdash [((atp_{\mathrm{r}};\ t := 0;\ \text{if } (\tau.a = -b) \text{ then } \mathcal{I} \text{ else } \mathcal{P} \wedge t' = 1 \wedge t \leq \varepsilon \text{ fi}) \cup rbc_{\mathrm{r}})^*]\,(\tau.p \geq m.e \rightarrow \tau.v \leq m.d)$$

To reason about the control loop, we need an invariant. Like in the other safety proof, we show that the controllability constraint ($\mathcal{C}$) is preserved during the loop. Initially $\mathcal{C}$ is valid, because it is already available as assumption in the antecedent. The use-case is identical to the one in the proof for Proposition 5. Therefore, we focus on the case, where we have to show, that the invariant is preserved by the loop body.

$$\mathcal{C} \vdash [(atp_{\mathrm{r}};\ t := 0;\ \text{if } (\tau.a = -b) \text{ then } \mathcal{I} \text{ else } \mathcal{P} \wedge t' = 1 \wedge t \leq \varepsilon \text{ fi}) \cup rbc_{\mathrm{r}}]\,\mathcal{C}$$

The choice splits up into cases, where we have to reason about the RBC choices. As the RBC is unchanged compared to Proposition 5 we omit these steps here.

$$\mathcal{C} \vdash [(atp_{\mathrm{r}};\ t := 0;\ \text{if } (\tau.a = -b) \text{ then } \mathcal{I} \text{ else } \mathcal{P} \wedge t' = 1 \wedge t \leq \varepsilon \text{ fi})]\,\mathcal{C}$$

We start with expanding the macro atp_{r}.

$$\mathcal{C} \vdash \Big[(SB := \frac{\tau.v^2 - m.d^2}{2b} + \left(\frac{A}{b} + 1\right)\left(\frac{A}{2}\varepsilon^2 + \varepsilon\ \tau.v\right);\ atp;\ t := 0;\ \text{if } (\tau.a = -b) \text{ then } \mathcal{I} \text{ else } \mathcal{P} \wedge t' = 1 \wedge t \leq \varepsilon \text{ fi})\Big]\mathcal{C}$$

We directly expand the macro *atp*.

$$\mathcal{C} \vdash \Big[((SB := \frac{\tau.v^2 - m.d^2}{2b} + \left(\frac{A}{b} + 1\right)\left(\frac{A}{2}\varepsilon^2 + \varepsilon\ \tau.v\right);\ \text{if } (m.e - \tau.p \leq SB \vee rbc.message = emergency) \text{ then } \tau.a := -b \text{ fi};\ t := 0;\ \text{if } (\tau.a = -b) \text{ then } \mathcal{I} \text{ else } \mathcal{P} \wedge t' = 1 \wedge t \leq \varepsilon \text{ fi})\Big]\mathcal{C}$$

From this point similar decomposition steps like in the previous proofs are performed (see, for instance, Sect. D.1). The case where $m.e - \tau.p \leq SB$ is very

similar to the proof for Proposition 5 and not shown here. For the other case the formula is decomposed along the program structure ending in the following case, where $m.e - \tau.p > SB$ and the PI controller regulates speed.

$$\mathcal{C} \wedge m.e - \tau.p > \frac{\tau.v^2 - m.d^2}{2b} + \left(\frac{A}{b} + 1\right)\left(\frac{A}{2}\varepsilon^2 + \varepsilon\ \tau.v\right) \vdash \\ [t := 0; \mathcal{P} \wedge t' = 1 \wedge t \leq \varepsilon]\,\mathcal{C}$$

Expanding the macro ($\mathcal{P}$) we get:

$$\mathcal{C} \wedge m.e - \tau.p > \frac{\tau.v^2 - m.d^2}{2b} + \left(\frac{A}{b} + 1\right)\left(\frac{A}{2}\varepsilon^2 + \varepsilon\ \tau.v\right) \vdash \\ \Big[t := 0; \tau.p' = \tau.v \wedge \tau.v' = \min\Big(A, \max\big(-b,\ p(\tau.v - m.r) - is - cm.r\big)\Big) \wedge \\ s' = \tau.v - m.r \wedge \tau.v \geq 0 \wedge t' = 1 \wedge t \leq \varepsilon\Big]\mathcal{C}$$

As quantifier elimination does not understand the min, max functions we rewrite this system with a loop over different choices for the differential equation for the velocity:

$$\tau.v' = \begin{cases} -b & \text{if } p(\tau.v - m.r) - is - cm.r \leq -b \\ A & \text{if } p(\tau.v - m.r) - is - cm.r \geq A \\ p(\tau.v - m.r) - is - cm.r & \text{otherwise} \end{cases}$$

In the following we abbreviate $p(\tau.v - m.r) - is - cm.r$ with PI

$$PI \equiv (p(\tau.v - m.r) - is - cm.r)$$

This gives the following formula:

$$\mathcal{C} \wedge m.e - \tau.p > \frac{\tau.v^2 - m.d^2}{2b} + \left(\frac{A}{b} + 1\right)\left(\frac{A}{2}\varepsilon^2 + \varepsilon\ \tau.v\right) \vdash \\ [t := 0; \\ \big((?A \geq PI \geq -b; \tau.p' = \tau.v \wedge \tau.v' = PI \wedge s' = \tau.v - m.r \wedge \tau.v \geq 0 \\ \wedge t' = 1 \wedge t \leq \varepsilon \wedge A \geq PI \geq -b) \\ \cup (?PI \geq A;\ \tau.p' = \tau.v \wedge \tau.v' = A \wedge s' = \tau.v - m.r \wedge \tau.v \geq 0 \\ \wedge t' = 1 \wedge t \leq \varepsilon \wedge PI \geq A) \\ \cup (?PI \leq -b;\ \tau.p' = \tau.v \wedge \tau.v' = -b \wedge s' = \tau.v - m.r \wedge \tau.v \geq 0 \\ \wedge t' = 1 \wedge t \leq \varepsilon \wedge PI \leq -b))^*\,]\,\mathcal{C}$$

As invariant for the inner loop resulting from the representation of ($\mathcal{P}$), we also show that it preserves ($\mathcal{C}$). For the two cases where the controller chooses

either A or $-b$ it can be shown directly by solving the system of differential equations and applying quantifier elimination. For the remaining case which contains the PI equation, we strengthen the system by overapproximating this differential equation with $\tau.v' \leq A$ (see [24] for formal details) and we add the invariant

$$m.e - \tau.p \geq \frac{\tau.v^2 - m.d^2}{2b} + \left(\frac{A}{b} + 1\right)\left(\frac{A}{2}(\varepsilon - t)^2 + (\varepsilon - t)\tau.v\right) \quad .$$

Using the differential induction rule [24], it can be proven easily that the overapproximation and this resulting differential invariant are valid.

With this differential invariant at hand, KeYmaera easily proves that controllability ($\mathcal{C}$) is preserved by the inner loop and therefore by the outer control loop too. This completes the safety proof for the PI controlled train system as stated in Proposition 8.

D.5 Proof Sketch for Proposition 9: Controllability despite Disturbance

In this section we sketch an example proof performed by KeYmaera for Proposition 9. For the disturbed case the proof for the controllability characterization starts like the proof in Sect. D.1 for Proposition 1. The primary difference to Proposition 1 is that—due to the presence of disturbances—we cannot use differential equation solving but use differential invariants [24, 23] instead. Differential invariants do not need solutions of differential equations and can even prove properties of differential equations with disturbances.

Again, we use I as an abbreviation for the initial region:

$$I \equiv (m.d \geq 0 \wedge b > 0 \wedge z \leq m \wedge \tau.v \geq 0)$$

The essential part of the sequent proof for Proposition 1 is shown in Fig. 8. From the point where KeYmaera computed the solution to the differential equation system in the first proof, the proof changes.

Case $(*)$ *in Fig. 8.* This case corresponds to showing necessity of the ($\mathcal{C}$) constraint. Here, we weaken the differential inequality to the equality case [24]. This is possible, because the differential equation systems occurs in a box modality (all behavior) in the antecedent (assumptions) of the sequent. Therefore, we just have to find one run, that satisfies the succedent of the sequent. Like in the undisturbed version of the proposition, this holds for the worst-case, where maximum brakes are applied, i.e. the case, where the disturbance adds up to increasing the braking power.

For showing that we can weaken the differential inequality to the equality case, we have to show two goals [24]:

$$\dfrac{\dfrac{(*)}{I, \big([\tau.p' = v, -b - l \le \tau.v' \le -b + u, \tau.v \ge 0] \; (\tau.p \ge m.e \to \tau.v \le d)\big) \vdash \tau.v^2 - m.d^2 \le 2(b-u)(m.e - \tau.p)} \quad \dfrac{(**)}{I, \tau.v^2 - m.d^2 \le 2(b-u)(m.e-\tau.p) \vdash \big([\tau.p' = v, -b - l \le \tau.v' \le -b + u, \tau.v \ge 0] \; (\tau.p \ge m.e \to \tau.v \le d)\big)}}{I \vdash \big([\tau.p' = v, -b - l \le \tau.v' \le -b + u, \tau.v \ge 0] \; (\tau.p \ge m.e \to \tau.v \le d)\big) \leftrightarrow \tau.v^2 - m.d^2 \le 2(b-u)(m.e - \tau.p)}}{m.d \ge 0 \land b > 0 \land z \le m \land \tau.v \ge 0 \vdash \big([\tau.p' = v, -b - l \le \tau.v' \le -b + u, \tau.v \ge 0] \; (\tau.p \ge m.e \to \tau.v \le d)\big) \leftrightarrow \tau.v^2 - m.d^2 \le 2(b-u)(m.e - \tau.p)}$$

$$\vdash \; (m.d \ge 0 \land b > 0 \land z \le m \land \tau.v \ge 0) \to \big([\tau.p' = v, b - l \le \tau.v' \le -b + u, \tau.v \ge 0](\tau.p \ge m.e \to \tau.v \le d)\big) \leftrightarrow \tau.v^2 - m.d^2 \le 2(b-u)(m.e - \tau.p)$$

Fig. 8: Essential part of the proof for Proposition 1

1. First, we have to show system entailment, i.e. the runs of the new system are a subset of those of the original system, which is expressed by:

$$I \vdash \forall v_0 (p_{1,0} = \tau.v \land v1_0 = -b + u \land \tau.v \ge 0) \to -b - l \le v1_0 \le -b + u \land \tau.v \ge 0$$

   This goal closes by quantifier elimination.
2. The second goal uses the modified differential equation systems

$$I, \big([\tau.p' = v, \tau.v' = -b + u, \tau.v \ge 0] \, (\tau.p \ge m.e \to \tau.v \le d)\big) \vdash \tau.v^2 - m.d^2 \le 2(b-u)(m.e - \tau.p)$$

   Here we can use the solution of the differential equation system and the goal closes by quantifier elimination.

Case $(**)$ *in Fig. 8.* This case shows that $\mathcal{C}$ is sufficient. Here, we cannot simply reduce the system to an equality case, as we have to show that the postcondition is satisfied for *every* possible evolution corresponding to *all* possible disturbances! Neither is there a closed-form solution to use for this system. Therefore, instead of weakening the system, we strengthen its invariant part by adding the auxiliary evolution domain $\tau.v^2 - m.d^2 \le 2(b-u)(m.e - \tau.p)$, see [24] for soundness of this kind of reasoning.

$$I, \tau.v^2 - m.d^2 \leq 2(b-u)(m.e - \tau.p) \vdash \big([\tau.p' = v, -b-l \leq \tau.v' \leq -b+u, \tau.v \geq 0]\,(\tau.p \geq m.e \rightarrow \tau.v \leq d)\big)$$

For strengthening the differential inequality system, in a sound way, we have to show that the added auxiliary invariant is true all along the dynamics and afterwards, that the augmented system is strong enough to imply the postcondition [24].

1. For showing that the invariant is valid, we use the differential induction rule [24].

$$I, \tau.v^2 - m.d^2 \leq 2(b-u)(m.e - \tau.p) \vdash \big([\tau.p' = v, -b-l \leq \tau.v' \leq -b+u, \tau.v \geq 0] \big(\tau.v^2 - m.d^2 \leq 2(b-u)(m.e - \tau.p)\big)\big)$$

   (a) First, we show that the differential invariant is initially valid:

$$I, \tau.v^2 - m.d^2 \leq 2(b-u)(m.e-\tau.p) \vdash \tau.v^2 - m.d^2 \leq 2(b-u)(m.e-\tau.p)$$

   This goal can be closed, as the same formula occurs on both sides of the sequent.

   (b) Second, we have to show that the differential invariant is preserved by the differential equation [24]:

$$I, \tau.v^2 - m.d^2 \leq 2(b-u)(m.e - \tau.p) \vdash \forall \tau_0.v, \delta_v \big((-d-l \leq \delta_v \wedge \delta_v \leq -b+u \wedge \tau_0.v \geq 0) \rightarrow (\delta_v v_0 \leq (u-b)v_0)\big)$$

   This closes by quantifier elimination.

2. Now that we have shown that the auxiliary invariant is a valid differential invariant, we can replace the differential inequality system by its invariant region using differential weakening [24].

$$I, \tau.v^2 - m.d^2 \leq 2(b-u)(m.e - \tau.p) \vdash \forall \tau_1 \ \big(\big(\tau_1.v \geq 0 \wedge \tau_1.v^2 - m.d^2 \leq 2(b-u)(m.e - \tau_1.p)\big) \rightarrow (\tau.p \geq m.e \rightarrow \tau_1.v \leq m.d)\big)$$

This goal then closes by quantifier elimination as well, thereby proving Proposition 9.