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
[2] University of Oldenburg, Department of Computing Science, Germany

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 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

* *All propositions have been verified in KeYmaera!* This research was supported by DFG SFB/TR14 AVACS, and by NSF under grants no. CNS-0931985, CCF-0926181.

K. Breitman and A. Cavalcanti (Eds.): ICFEM 2009, LNCS 5885, pp. 246–265, 2009.

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]. Compared to our preliminary short report [12] we prove 12 additional properties including 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 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

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

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.

Batt et al. [3] give heuristics for splitting regions by linear constraints that can be used to determine parameter constraints. Frehse et al. [13] 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. [14] 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. [15] 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. [16] analyze consistency of informal requirements on ETCS expressed as temporal properties. 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.

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. We refine and verify it using symbolic decompositions [10] 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. Proofs are given in [17].

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 $\tau.a$.

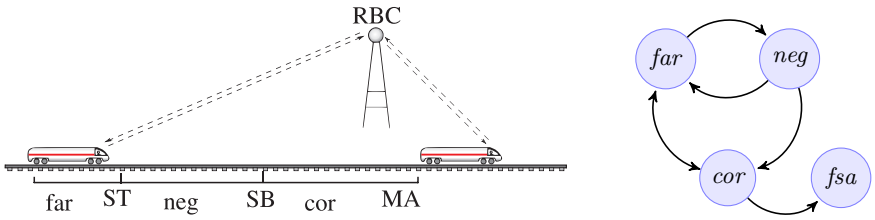
The $\mathsf{d}\mathcal{L}$*-formulas* are defined by the following grammar (θ_i are terms, x is a real-valued variable, $\sim \in \{<, \leq, =, \geq, >\}$, ϕ and ψ are formulas, α is a HP):

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

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 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 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. 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 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 [5].



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

Fig. 1: ETCS train cooperation protocol

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 (proof see [17]).*

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.

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. 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.

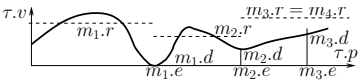


Fig. 2: ETCS track profile

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:

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

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 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

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

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

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. 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* using structural symbolic decomposition in $\mathsf{d}\mathcal{L}$: first, we decompose the uncontrolled system dynamics to a first-order formula characterizing the controllable state region, which specifies for which parameter combinations the system dynamics can actually be controlled safely by any control law. Next, we successively add partial control laws to the system while leaving its decision parameters (like SB or m) free and use structural symbolic decomposition again to discover parametric constraints that preserve controllability under these control laws. This step we repeat until the resulting system is proven safe. Finally, we 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 constitute good candidates for inductive invariants, and the parameter constraints discovered ensure that the control choices taken by the controller never leave the controllable domain.

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):

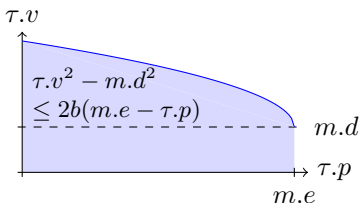


Fig. 4: Controllable region

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 252, 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 [18].

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*

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

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

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

This RBC controllability is characterized by the following valid formula:

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

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*

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

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

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

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

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. Consequently, 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.

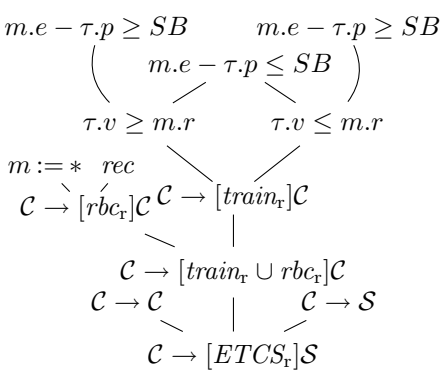


Fig. 6: Proof sketch for Proposition 5

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 ($\langle ETCS_r \rangle$). 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. It is easy to see from the domain restrictions ($\tau.v \geq 0 \wedge t \leq \varepsilon$) in drive that the assumptions ($\tau.v \geq 0$) and $\varepsilon > 0$ are necessary.

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 [19, 18] 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].*

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* [19], 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 [19, 18] 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 braking power b that limit the effective braking power to $(b - u)$:

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

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

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* $\mathcal{A} \land \tau.p \leq m.e \land b > u \geq 0 \land l \geq 0$ *as regularity assumptions, the following equivalence holds:*

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

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:*

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

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

Fig. 7: Parametric ETCS cooperation protocol with disturbances

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 [19, 18] with a time-dependent version of ($\mathcal{B}_d$) as differential invariant:

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

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 rules in column Steps. In most cases proofs can be found automatically [18].

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 dℒ 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
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 dℒ. 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. Proof sketches are presented in [17].

Acknowledgments. We like to thank Johannes Faber 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.

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 LNCS., Springer (2007)
3. Batt, G., Belta, C., Weiss, R.: Model checking genetic regulatory networks with parameter uncertainty. In Bemporad, A., Bicchi, A., Buttazzo, G., eds.: HSCC. Volume 4416 of LNCS., Springer (2007)
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. In Peled, D., Tsay, Y.K., eds.: ATVA. Volume 3707 of LNCS., Springer (2005) 217–233
8. Frehse, G.: PHAVer: Algorithmic verification of hybrid systems past HyTech. In Morari, M., Thiele, L., eds.: HSCC. Volume 3414 of LNCS., Springer (2005)
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. 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
14. Tomlin, C., Lygeros, J., Sastry, S.: A Game Theoretic Approach to Controller Design for Hybrid Systems. Proceedings of IEEE **88** (2000) 949–969
15. Peleska, J., Große, D., Haxthausen, A.E., Drechsler, R.: Automated verification for train control systems. In: FORMS/FORMAT. (2004)
16. Cimatti, A., Roveri, M., Tonetta, S.: Requirements validation for hybrid systems. In Bouajjani, A., Maler, O., eds.: CAV. Volume 5643 of LNCS., Springer (2009)
17. Platzer, A., Quesel, J.D.: European train control system: A case study in formal verification. Report 54, SFB/TR 14 AVACS (2009) ISSN: 1860-9821, avacs.org.
18. 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.
19. Platzer, A.: Differential-algebraic dynamic logic for differential-algebraic programs. J. Log. Comput. (2008) DOI 10.1093/logcom/exn070.