Logical Verification and Systematic Parametric Analysis in Train Control*

André Platzer and Jan-David Quesel

University of Oldenburg, Department of Computing Science, Germany
{platzer|quesel}@informatik.uni-oldenburg.de

Abstract. We formally verify hybrid safety properties of cooperation protocols in a fully parametric version of the *European Train Control System* (ETCS). We present a formal model using hybrid programs and verify correctness using our logic-based decomposition procedure. This procedure supports free parameters and parameter discovery, which is required to determine correct design choices for free parameters of ETCS.

Keywords: parametric verification, logic for hybrid systems, symbolic decomposition

1 Introduction

Most hybrid systems contain substantial degrees of freedom including how specific parameters are instantiated or adjusted [3, 2, 5]. Yet, virtually any hybrid system is only safe under certain constraints on these parameters. For instance, the *European Train Control System* (ETCS) has a wide range of different possible configurations of trains, track layouts, and different driving circumstances. Still, it only is safe under certain conditions on external parameters, e.g., when the speed of each train does not exceed its specific braking power given the remaining distance to the next train. Similarly, internal control design parameters for speed control and braking triggers need to be adjusted in accordance with the train dynamics. Moreover, parameters must be constrained such that the system remains correct when passing from instant reaction continuous models to *sampled data discrete time controllers* of hardware implementations. Yet, determining the range of external parameters and choice of internal design parameters for which ETCS is safe, is not possible just by looking at the model.

Likewise, it is difficult to read off the parameter constraints that are required for correctness from a failed verification attempt of model checkers [7], as these often exploit non-structural heuristic splits of the state space, which can lead to nonuniform parameter requirements for different states. Model checkers for hybrid systems, e.g. HyTech [1] and PHAVer [7], verify by exploring the state space of the system. For these model checkers concrete numbers for most of the parameters are necessary. To discover constraints on free parameters, we use a

* This research was partly supported by the German Research Council (DFG) of the Transregional Collaborative Research Center (SFB/TR 14 AVACS).

M. Egerstedt and B. Mishra (Eds.): HSCC 2008, LNCS 4981, pp. 646–649, 2008.

logic based approach and verify safety properties of the parametric ETCS case study with significant automation in our new verification tool KeYmaera.

Batt et al. [2] give heuristics for splitting regions by linear constraints that can be used to determine parameter constraints. This approach is not applicable in ETCS, which requires nonlinear parameter constraints for correctness.

2 Differential Dynamic Logic

The logic $\mathsf{d}\mathcal{L}$ [9, 10] 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 in $\mathsf{d}\mathcal{L}$. Generalizing the principle of dynamic logic [8] 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. For hybrid system α, $\mathsf{d}\mathcal{L}$ provides correctness statements like $[\alpha]\phi$, that expresses that all traces of system α lead to states in which condition ϕ holds. Further, $\mathsf{d}\mathcal{L}$ provides conditional correctness statements like $\phi \rightarrow [\alpha]\psi$, saying that α satisfies ψ if condition ϕ holds at the initial state.

As a uniform operational model, $\mathsf{d}\mathcal{L}$ provides *hybrid programs* (HP) as a program notation for hybrid systems that is amenable to deductive structural decomposition in $\mathsf{d}\mathcal{L}$ [9, 10]. HP of $\mathsf{d}\mathcal{L}$ can represent hybrid automata [1], unlike other logics [4]. Hybrid programs are regular combinations of basic actions: the assertion that ϕ holds is written as $?\phi$, $x := \theta$ to assign the value of θ to the variable x, random real numbers can be assigned using $x := *$, and $\dot{x} = \theta$ is used to express continuous evolutions along differential equations. The regular combination operators are $\alpha; \beta$ for sequential composition, $\alpha \cup \beta$ for non-deterministic choice and α^* to represent the repetition of hybrid automata transitions.

3 Fully Parametric European Train Control System

The European Train Control System (ETCS) [5, 6] is a standard to assure safe operation of trains and high throughput of high speed trains. ETCS level 3 follows the *moving block principle*, i.e., movement authorities are not known beforehand but determined based on the current track situation by a *Radio Block Controller* (RBC). Trains are only allowed to move within their current movement authority block (denoted by m), 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 m. The automatic train protection unit (*atp*) determines a safety envelope around the train, within which it considers driving safe, and adjusts the train acceleration a accordingly. Figure 1 illustrates the dynamic assignment of movement authorities. When approaching the end of its movement authority 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* (*start talking*). During negotiation the RBC grants or denies m-extensions. Instead, the RBC can announce emergencies, which force train controllers to switch to the recovery mode applying full

$$
\begin{array}{lcl}
\text{ETCS} & : & (\text{train} \cup \text{rbc})^* \\
\text{train} & : & \text{spd};\ \text{atp};\ \text{move} \\
\text{spd} & : & (?v \leq r;\ a := *;\ ?-b \leq a \leq A) \\
 & & \cup (?v \geq r;\ a := *;\ ?0 > a \geq -b) \\
\text{atp} & : & (?(m-p \leq SB \vee msg = stop);\ a := -b) \\
 & & \cup (?m-p \geq SB \wedge msg \neq stop) \\
\text{move} & : & t := 0;\ (\dot{p} = v, \dot{v} = a, \dot{t} = 1 \,\&\, v \geq 0 \wedge t \leq \varepsilon) \\
\text{rbc} & : & (msg := stop) \\
 & & \cup\ (m := *;\ ?v^2 \leq 2b(m-p);\ r := *;\ r > 0)
\end{array}
$$

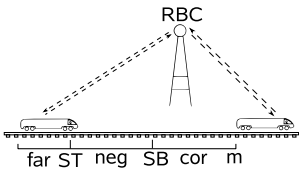

Fig. 1. ETCS train coordination protocol

emergency brakes. After the train has come to a full stop, the controller switches to a failsafe state and awaits manual clearance. If the RBC does not grant m-extension in time or messages are lost, the train starts immediate recovery after passing the point SB (*start braking*).

4 Parametric Verification of the ETCS System

After determining a correctness constraint $SB \geq \frac{v^2}{2b} + (\frac{A}{b} + 1)(\frac{A}{2}\varepsilon^2 + \varepsilon\, v)$ on the free parameters [10] we prove the following safety property of ETCS:

Proposition 1 (Safety). *Assuming the train starts in a controllable state, the following global and unbounded-horizon safety formula about the system in Fig. 1* [ETCS] $p \leq m$ *holds.*

As system invariant we choose $inv \equiv v^2 \leq 2b(m-p) \wedge \varepsilon > 0 \wedge v \geq 0$, which expresses that it is possible to completely stop the train within the distance left to the end of the movement authority. This constraint describes a controllable state of the train and therefore we choose inv as initial configuration of our system.

As an example to illustrate the proof structure for the verification of Proposition 1 in KeYmaera by automatic decomposition, consider the sketch in Fig. 2. By convention, such proofs start with the conjecture at the bottom and proceed by decomposition to the leafs. We need to prove that the assumption that the train is in a controllable state expressed by inv entails $p \leq m$. As the system consists of a global loop, we need to prove that inv is an invariant of this loop. Using KeYmaera it can be shown easily that the invariant is initially valid and implies the post condition. As usual, proving that the invariant is preserved by the loop-body is the most challenging part of the proof (lower middle branch). On the left branch we have to show that the RBC

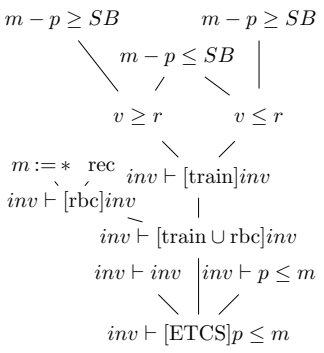

Fig. 2. Proof graph

preserves the invariant. On the right branch we have to show that the train controller also 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. The next split on both of these branches depends on the value of SB. If the train has passed the point SB, the train applies maximal brakes and the goal can be closed as consequence from *inv*. The outer branches, where the train has not passed SB, can be closed as train behavior before SB is not safety critical.

All correctness properties and parameter constraints of ETCS can be verified with 95.6% to 100% automation in our deductive verification tool KeYmaera, see Tab. 1 for experimental results.

Table 1. Experimental results for ETCS in the verification tool KeYmaera

Case study	Proof steps	Interactions	Time	Symbolic variables
Safety	190	1	4303s	15
Safety (simplified)	160	1	85s	15
Controllability	18	0	0.5s	5
RBC controllability	45	0	1.1s	13
Reactivity	150	0	8.8h	10
Liveness	112	5	21s	10
Reactivity corollary	344	14	289s	15

References

1. Alur, R., Henzinger, T.A., Ho, P.H.: Automatic symbolic verification of embedded systems. IEEE Trans. Software Eng. **22**(3) (1996) 181–201
2. 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)
3. Damm, W., Hungar, H., Olderog, E.R.: Verification of cooperating travel agents. International Journal of Control **79**(5) (May 2006) 395–421
4. Davoren, J.M., Nerode, A.: Logics for hybrid systems. Proceedings of the IEEE **88**(7) (July 2000) 985–1010
5. ERTMS User Group, UNISIG: ERTMS/ETCS System requirements specification. http://www.aeif.org/ccm/default.asp (2002) Version 2.2.2.
6. Faber, J., Meyer, R.: Model checking data-dependent real-time properties of the European Train Control System. In: FMCAD, IEEE Computer (2006) 76–77
7. Frehse, G.: PHAVer: Algorithmic verification of hybrid systems past HyTech. In Morari, M., Thiele, L., eds.: HSCC. Volume 3414 of LNCS., Springer (2005) 258–273
8. Harel, D., Kozen, D., Tiuryn, J.: Dynamic logic. MIT Press (2000)
9. Platzer, A.: Differential dynamic logic for verifying parametric hybrid systems. In Olivetti, N., ed.: TABLEAUX. Volume 4548 of LNCS., Springer (2007) 216–232
10. Platzer, A.: Differential dynamic logic for hybrid systems. J. Autom. Reasoning (2008) To appear.