European Train Control System: A Case Study in Formal Verification

André Platzer[1] Jan-David Quesel[2]

[1]Carnegie Mellon University, Pittsburgh, PA

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

International Conference on Formal Engineering Methods (ICFEM),
Rio de Janeiro, 2009

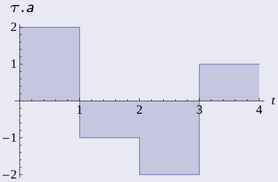
Deutsche Forschungsgemeinschaft DFG

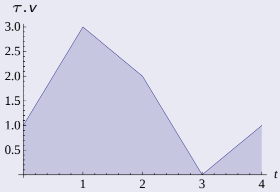
ETCS Control Verification

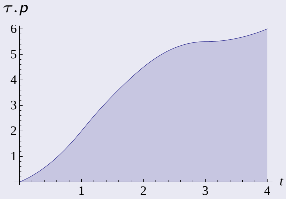
Problem

Hybrid System

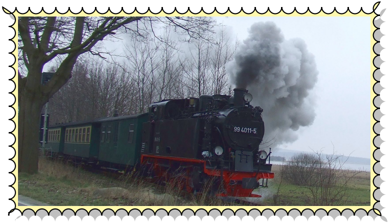
- Continuous evolutions (differential equations)
- Discrete jumps (control decisions)

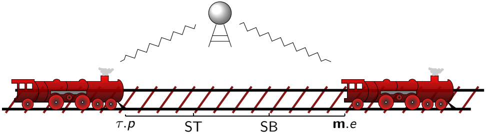






European Train Control System





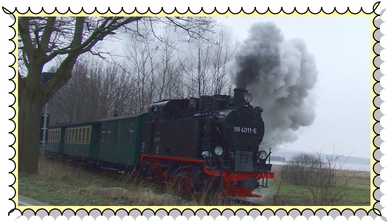
Objectives

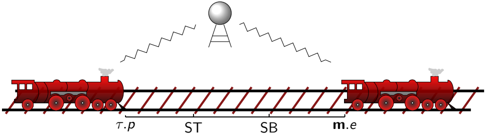
1. Collision free
2. Maximise throughput & velocity (300 km/h)
3. $2.1 * 10^6$ passengers/day

Overview

1. No static partitioning of track
2. Radio Block Controller (RBC) manages movement authorities dynamically
3. Moving block principle

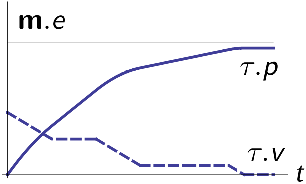
European Train Control System



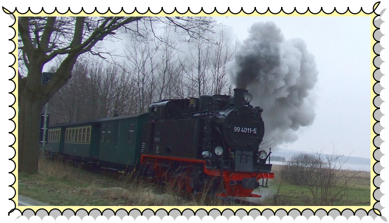


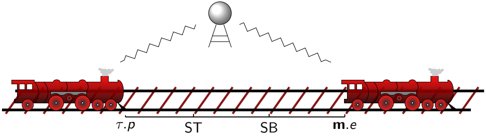
Parametric Hybrid Systems

continuous evolution along differential equations + discrete change



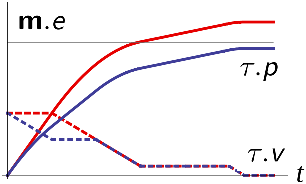
European Train Control System



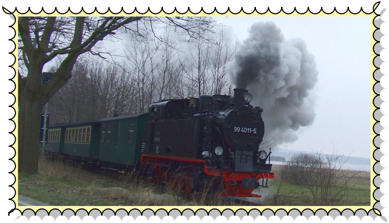


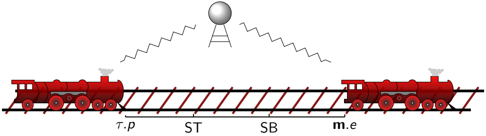
Parametric Hybrid Systems

continuous evolution along differential equations + discrete change



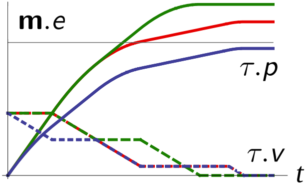
European Train Control System



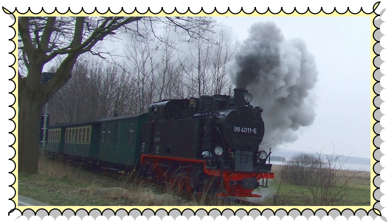


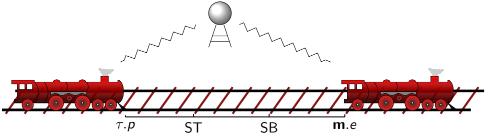
Parametric Hybrid Systems

continuous evolution along differential equations + discrete change



European Train Control System

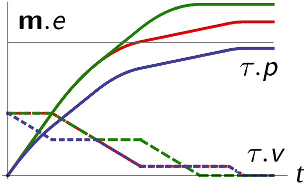




Parametric Hybrid Systems

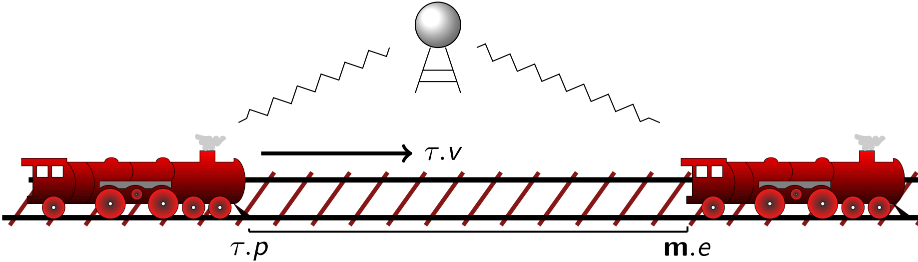
continuous evolution along differential equations + discrete change

- Parameters have nonlinear influence
- Handle *SB* as free symbolic parameter?
- Challenge: verification (falsifying is "easy")
- Which constraints for *SB*?



$\forall \mathbf{m}.e \, \exists SB$ "train always safe"

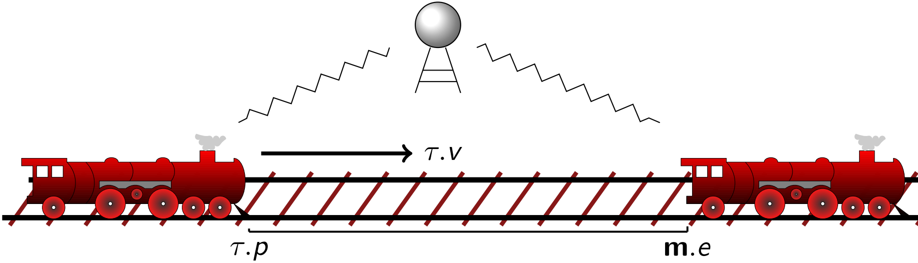
Differential Dynamic Logic (d$\mathcal{L}$)



Example

$$\underbrace{\qquad\qquad\qquad}_{\text{Precondition}} \rightarrow [\underbrace{\qquad\qquad\qquad\qquad\qquad\qquad}_{\text{Operation model}}](\underbrace{\qquad\quad}_{\text{Property}})$$

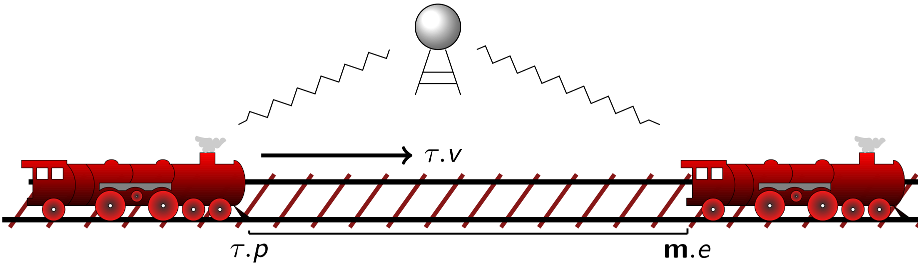
Differential Dynamic Logic (d$\mathcal{L}$)



Example

$$\underbrace{\tau.v^2 \le 2b(\mathbf{m}.e - \tau.p)}_{\text{Precondition}} \to [\underbrace{\qquad\qquad\qquad\qquad}_{\text{Operation model}}]\underbrace{(\tau.p \le \mathbf{m}.e)}_{\text{Property}}$$

Differential Dynamic Logic (d$\mathcal{L}$)

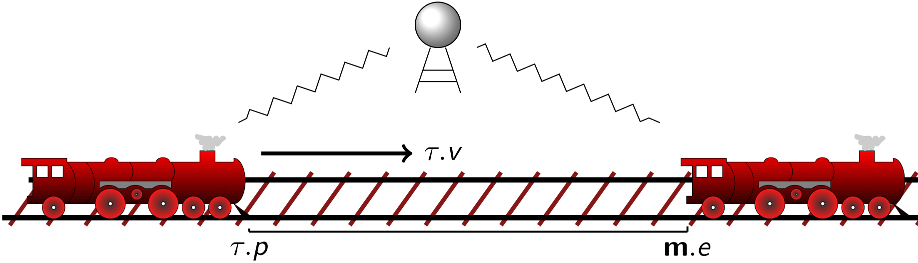


Example

$$\underbrace{\tau.v^2 \le 2b(\mathbf{m}.e - \tau.p)}_{\text{Precondition}} \rightarrow [\underbrace{\tau.p' = \tau.v, \tau.v' = \tau.a}_{\text{Operation model}}](\underbrace{\tau.p \le \mathbf{m}.e}_{\text{Property}})$$

Continuous evolution: differential equation

Differential Dynamic Logic (d$\mathcal{L}$)

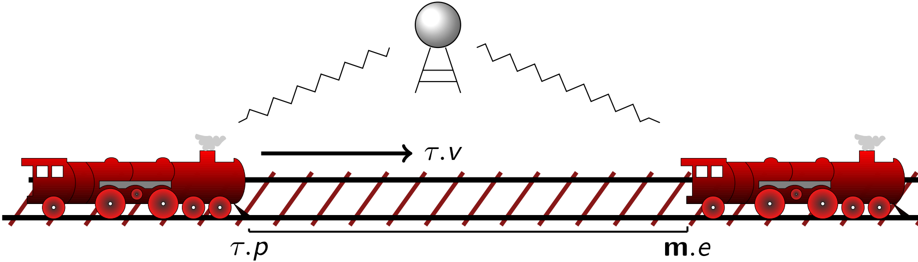


Example

$$\underbrace{\tau.v^2 \leq 2b(\mathbf{m}.e - \tau.p)}_{\text{Precondition}} \rightarrow \underbrace{[\tau.a := *; \quad \tau.p' = \tau.v, \tau.v' = \tau.a]}_{\text{Operation model}} \underbrace{(\tau.p \leq \mathbf{m}.e)}_{\text{Property}}$$

Random assignment

Differential Dynamic Logic (d$\mathcal{L}$)

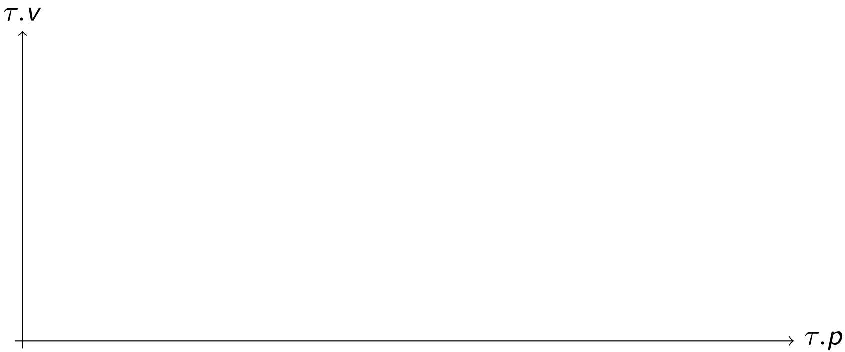


Example

$$\underbrace{\tau.v^2 \le 2b(\mathbf{m}.e - \tau.p)}_{\text{Precondition}} \rightarrow [\underbrace{\tau.a := *;\ ?\tau.a \le -b;\ \tau.p' = \tau.v, \tau.v' = \tau.a}_{\text{Operation model}}]\underbrace{(\tau.p \le \mathbf{m}.e)}_{\text{Property}}$$

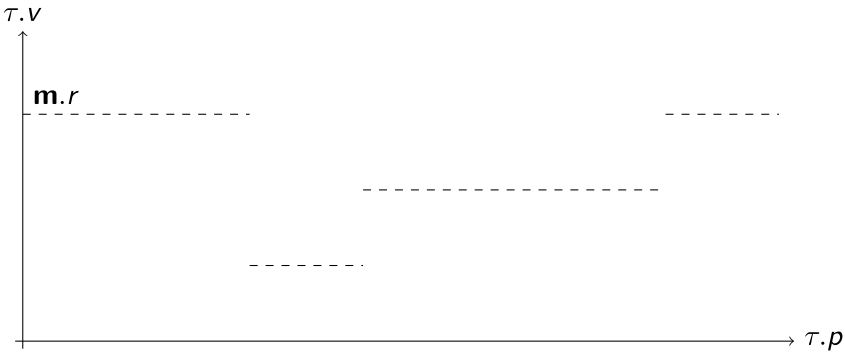
Test

3D Movement Authorities



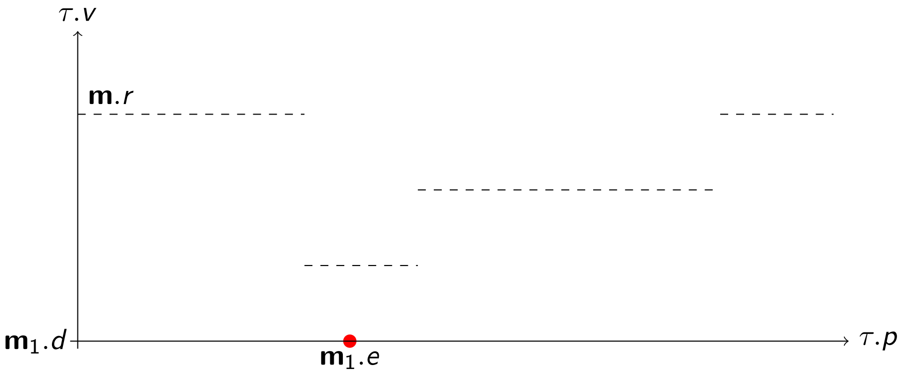
- Vectorial MA $\mathbf{m} = (d, e, r)$:
- Beyond point $\mathbf{m}.e$ train not faster than $\mathbf{m}.d$.
- Train should try not to keep *recommended speed* $\mathbf{m}.r$

3D Movement Authorities



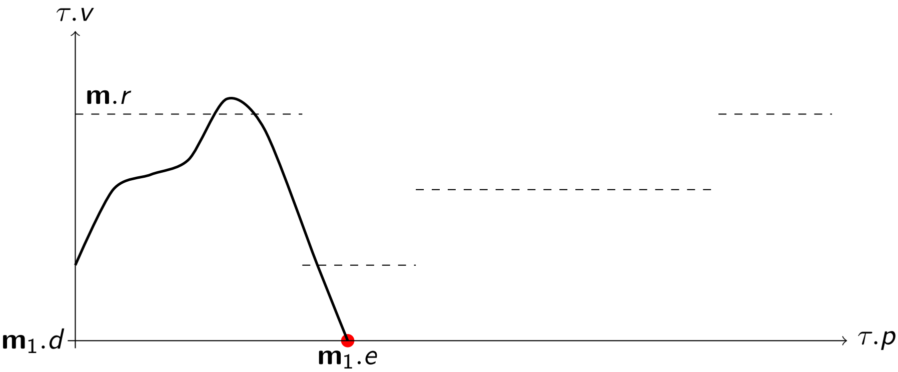
- Vectorial MA $\mathbf{m} = (d, e, r)$:
- Beyond point $\mathbf{m}.e$ train not faster than $\mathbf{m}.d$.
- Train should try not to keep *recommended speed* $\mathbf{m}.r$

3D Movement Authorities



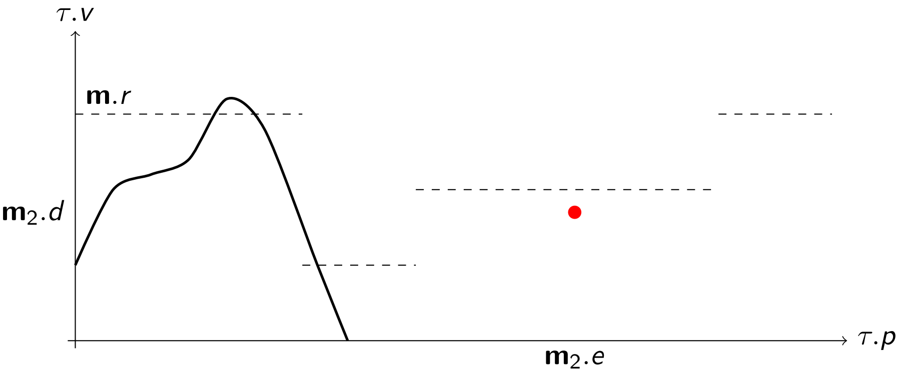
- Vectorial MA $\mathbf{m} = (d, e, r)$:
- Beyond point $\mathbf{m}.e$ train not faster than $\mathbf{m}.d$.
- Train should try not to keep *recommended speed* $\mathbf{m}.r$

3D Movement Authorities



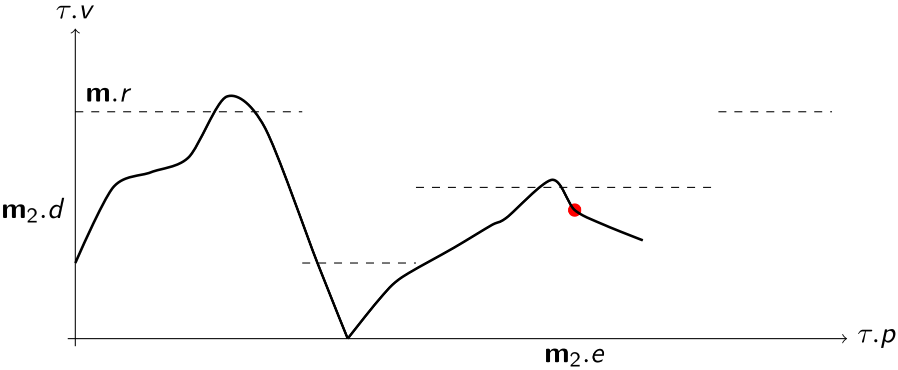
- Vectorial MA $\mathbf{m} = (d, e, r)$:
- Beyond point $\mathbf{m}.e$ train not faster than $\mathbf{m}.d$.
- Train should try not to keep *recommended speed* $\mathbf{m}.r$

3D Movement Authorities



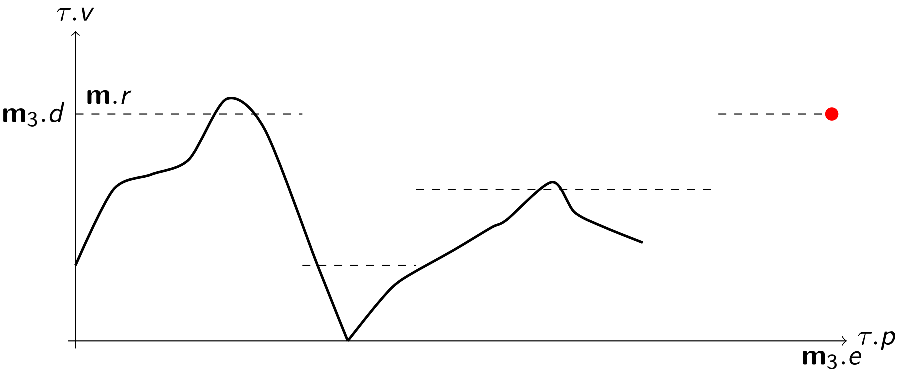
- Vectorial MA $\mathbf{m} = (d, e, r)$:
- Beyond point $\mathbf{m}.e$ train not faster than $\mathbf{m}.d$.
- Train should try not to keep *recommended speed* $\mathbf{m}.r$

3D Movement Authorities



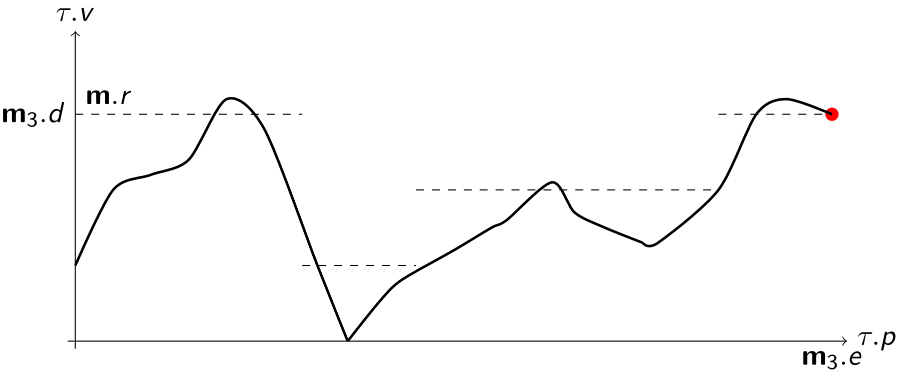
- Vectorial MA $\mathbf{m} = (d, e, r)$:
- Beyond point $\mathbf{m}.e$ train not faster than $\mathbf{m}.d$.
- Train should try not to keep *recommended speed* $\mathbf{m}.r$

3D Movement Authorities



- Vectorial MA $\mathbf{m} = (d, e, r)$:
- Beyond point $\mathbf{m}.e$ train not faster than $\mathbf{m}.d$.
- Train should try not to keep *recommended speed* $\mathbf{m}.r$

3D Movement Authorities

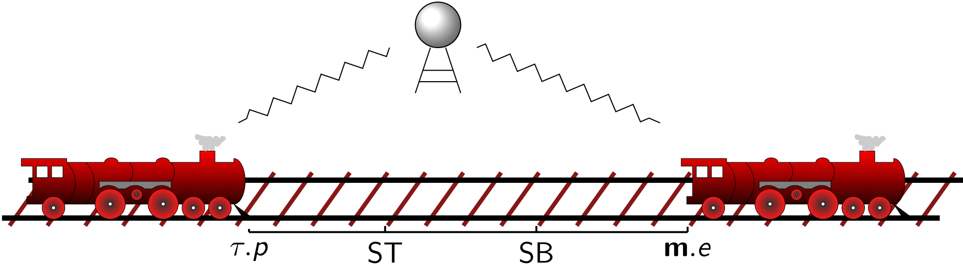


- Vectorial MA $\mathbf{m} = (d, e, r)$:
- Beyond point $\mathbf{m}.e$ train not faster than $\mathbf{m}.d$.
- Train should try not to keep *recommended speed* $\mathbf{m}.r$

Separation Principle

Lemma (Principle of separation by movement authorities)

Each train respects its movement authority and
the RBC partitions into disjoint movement authorities
$\Rightarrow$ *trains can never collide.*



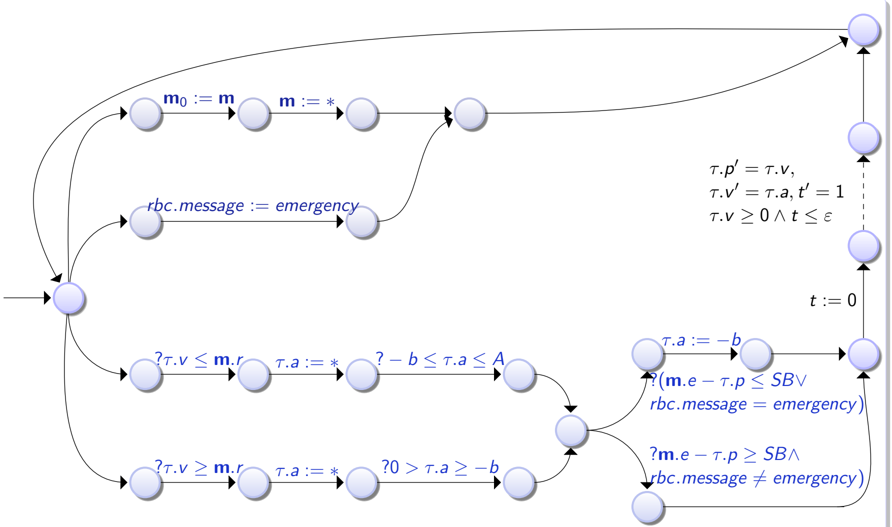
Parametric Skeleton of ETCS

Read from the informal specification...

$$
\begin{aligned}
ETCS_{skel} &: (train \cup rbc)^* \\
train &: spd;\ atp;\ drive \\
spd &: \quad (?\tau.v \le \mathbf{m}.r;\ \tau.a := *;\ ?-b \le \tau.a \le A) \\
&\quad \cup(?\tau.v \ge \mathbf{m}.r;\ \tau.a := *;\ ?-b \le \tau.a \le 0) \\
atp &: \mathtt{if}(\mathbf{m}.e - \tau.p \le SB \lor rbc.message = emergency)\, \tau.a := -b \\
drive &: t := 0;\ (\tau.p' = \tau.v, \tau.v' = \tau.a, t' = 1 \land \tau.v \ge 0 \land t \le \varepsilon) \\
rbc &: (rbc.message := emergency)\ \cup\ (\mathbf{m} := *;\ ?\mathbf{m}.r > 0)
\end{aligned}
$$

Parametric Skeleton of ETCS

As transition system...



Parametric Skeleton of ETCS

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

Task

Verify safety

Parametric Skeleton of ETCS

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

Task

Verify safety

Specification

$[ETCS_{skel}](\tau.p \geq \mathbf{m}.e \rightarrow \tau.v \leq \mathbf{m}.d)$

Parametric Skeleton of ETCS

$$
\begin{aligned}
ETCS_{skel} &: (train \cup rbc)^* \\
train &: spd;\ atp;\ drive \\
spd &: \quad (?\tau.v \le \mathbf{m}.r;\ \tau.a := *;\ ?-b \le \tau.a \le A) \\
&\quad \cup(?\tau.v \ge \mathbf{m}.r;\ \tau.a := *;\ ?-b \le \tau.a \le 0) \\
atp &: \texttt{if}(\mathbf{m}.e - \tau.p \le SB \lor rbc.message = emergency)\, \tau.a := -b \\
drive &: t := 0;\ (\tau.p' = \tau.v, \tau.v' = \tau.a, t' = 1 \land \tau.v \ge 0 \land t \le \varepsilon) \\
rbc &: (rbc.message := emergency)\ \cup\ (\mathbf{m} := *;\ ?\mathbf{m}.r > 0)
\end{aligned}
$$

Task

Verify safety

Specification

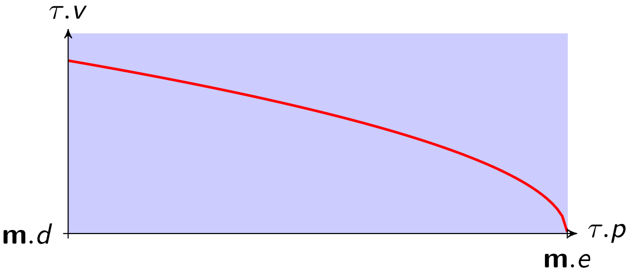
$[ETCS_{skel}](\tau.p \ge \mathbf{m}.e \to \tau.v \le \mathbf{m}.d)$

Issue

Lots of counterexamples!

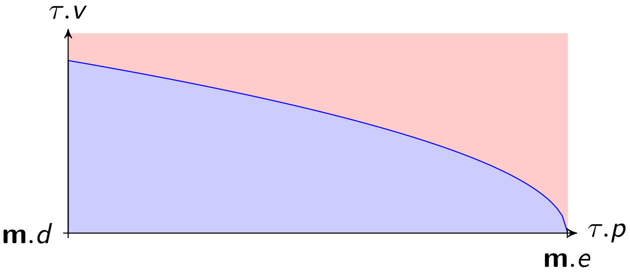
Iterative Control Refinement Process

Iterative Control Refinement Process



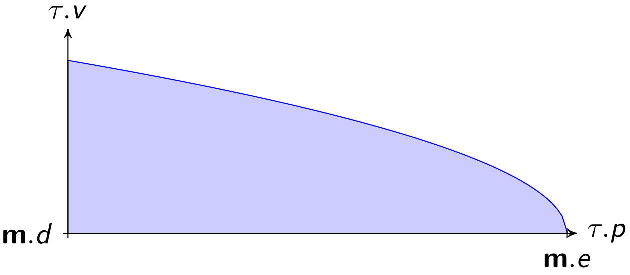
1. Controllability discovery

Iterative Control Refinement Process



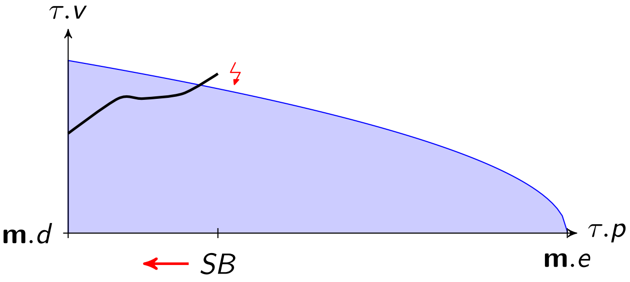
1. Controllability discovery

Iterative Control Refinement Process



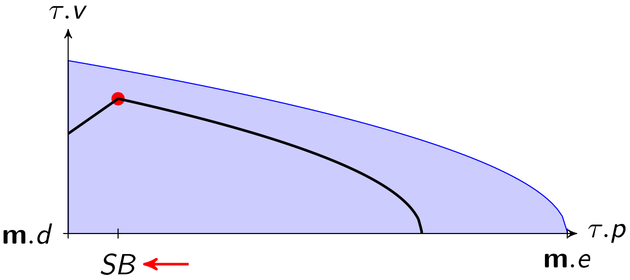
1. Controllability discovery

Iterative Control Refinement Process



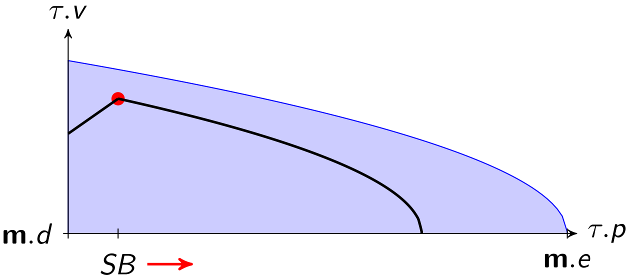
1. Controllability discovery
2. Control refinement

Iterative Control Refinement Process



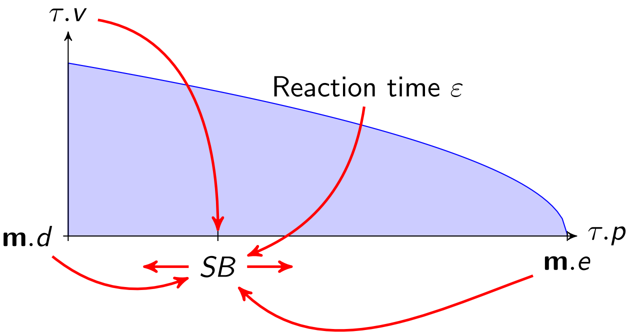
1. Controllability discovery
2. Control refinement

Iterative Control Refinement Process



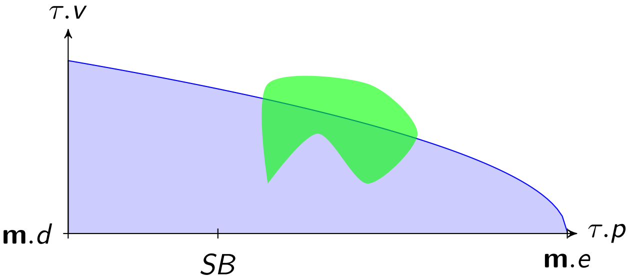
1. Controllability discovery
2. Control refinement

Iterative Control Refinement Process



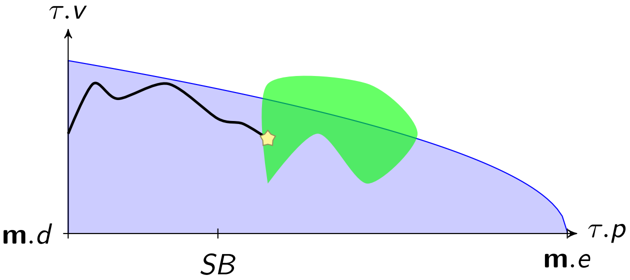
1. Controllability discovery
2. Control refinement

Iterative Control Refinement Process



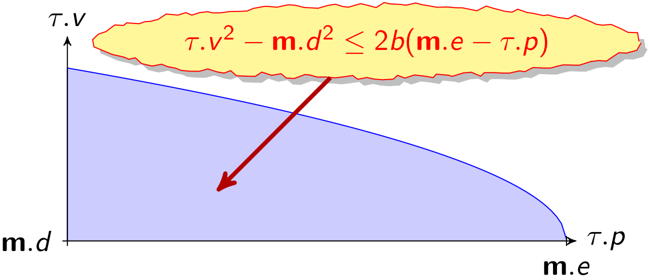
1. Controllability discovery
2. Control refinement
3. Repeat 2 until safety can be proven

Iterative Control Refinement Process



1. Controllability discovery
2. Control refinement
3. Repeat 2 until safety can be proven
4. Liveness check

ETCS Controllability

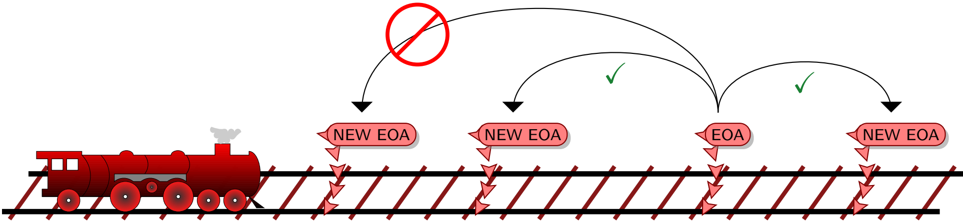


Proposition (Controllability)

$$[\tau.p' = \tau.v, \tau.v' = -b \wedge \tau.v \geq 0](\tau.p \geq \mathbf{m}.e \rightarrow \tau.v \leq \mathbf{m}.d)$$

$$\equiv \ \tau.v^2 - \mathbf{m}.d^2 \leq 2b(\mathbf{m}.e - \tau.p) \qquad (\mathcal{C})$$

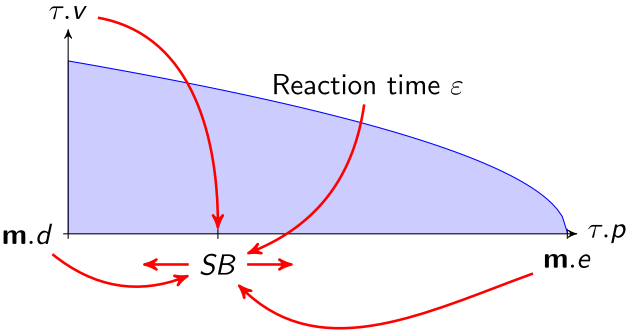
ETCS RBC Controllability



Proposition (RBC Controllability)

$$\mathbf{m}.d \geq 0 \land b > 0 \rightarrow [\mathbf{m}_0 := \mathbf{m};\ rbc] \Big(\mathbf{m}_0.d^2 - \mathbf{m}.d^2 \leq 2b(\mathbf{m}.e - \mathbf{m}_0.e) \land \mathbf{m}_0.d \geq 0 \land \mathbf{m}.d \geq 0 \leftrightarrow \forall \tau \left((\langle \mathbf{m} := \mathbf{m}_0 \rangle \mathcal{C}) \rightarrow \mathcal{C} \right) \Big)$$

ETCS Reactivity



Proposition (Reactivity)

$$\left(\forall \mathbf{m}.e\,\forall \tau.p\left(\mathbf{m}.e-\tau.p \geq SB \wedge \mathcal{C} \rightarrow [\tau.a := A;\ drive]\,\mathcal{C}\right)\right)$$

$$\equiv SB \geq \frac{\tau.v^2-\mathbf{m}.d^2}{2b}+\left(\frac{A}{b}+1\right)\left(\frac{A}{2}\varepsilon^2+\varepsilon\,\tau.v\right)$$

Refined ETCS Control

$$
\begin{array}{lcl}
ETCS_r: & & (train \cup rbc)^* \\
train & : & spd;\ atp;\ drive \\
spd & : & (?\tau.v \le \mathbf{m}.r;\ \tau.a := *;\ ?-b \le \tau.a \le A) \\
 & & \cup(?\tau.v \ge \mathbf{m}.r;\ \tau.a := *;\ ?0 > \tau.a \ge -b) \\
atp & : & SB := \frac{\tau.v^2 - \mathbf{m}.d^2}{2b} + \left(\frac{A}{b} + 1\right)\left(\frac{A}{2}\varepsilon^2 + \varepsilon\ \tau.v\right); \\
 & : & \mathtt{if}(\mathbf{m}.e - \tau.p \le SB \vee rbc.message = emergency)\, \tau.a := -b \\
drive & : & t := 0;\ (\tau.p' = \tau.v, \tau.v' = \tau.a, t' = 1 \wedge \tau.v \ge 0 \wedge t \le \varepsilon) \\
rbc & : & (rbc.message := emergency) \\
 & & \cup \big(\mathbf{m}_0 := \mathbf{m}; \mathbf{m} := *; \\
 & & \quad ?\mathbf{m}_0.d^2 - \mathbf{m}.d^2 \le 2b(\mathbf{m}.e - \mathbf{m}_0.e) \wedge \mathbf{m}.r \ge 0 \wedge \mathbf{m}.d \ge 0\big)
\end{array}
$$

Refined ETCS Control

$$
\begin{array}{lll}
ETCS_r\colon & (train \cup rbc)^* \\
train & : & spd;\ atp;\ drive \\
spd & : & (?\tau.v \le \mathbf{m}.r;\ \tau.a := *;\ ? - b \le \tau.a \le A) \\
 & & \cup(?\tau.v \ge \mathbf{m}.r;\ \tau.a := *;\ ?0 > \tau.a \ge -b) \\
atp & : & SB := \frac{\tau.v^2 - \mathbf{m}.d^2}{2b} + \left(\frac{A}{b} + 1\right)\left(\frac{A}{2}\varepsilon^2 + \varepsilon\ \tau.v\right); \\
 & : & \mathtt{if}(\mathbf{m}.e - \tau.p \le SB \vee rbc.message = emergency)\ \tau.a := -b \\
drive & : & t := 0;\ (\tau.p' = \tau.v, \tau.v' = \tau.a, t' = 1 \wedge \tau.v \ge 0 \wedge t \le \varepsilon) \\
rbc & : & (rbc.message := emergency) \\
 & & \cup \big(\mathbf{m}_0 := \mathbf{m}; \mathbf{m} := *; \\
 & & \quad ?\mathbf{m}_0.d^2 - \mathbf{m}.d^2 \le 2b(\mathbf{m}.e - \mathbf{m}_0.e) \wedge \mathbf{m}.r \ge 0 \wedge \mathbf{m}.d \ge 0\big)
\end{array}
$$

Specification

$$\tau.v^2 - \mathbf{m}.d^2 \le 2b(\mathbf{m}.e - \tau.p) \to [ETCS_r](\tau.p \ge \mathbf{m}.e \to \tau.v \le \mathbf{m}.d)$$

Refined ETCS Control

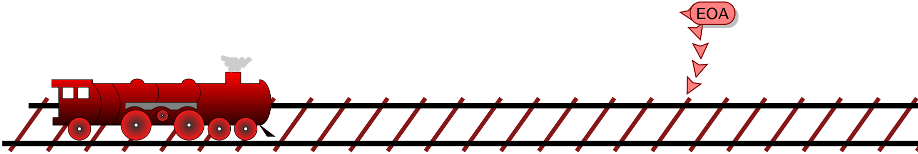
Necessary for safety

$$
\begin{array}{lll}
ETCS_r: & (train \cup rbc)^* \\
train & : & spd;\ atp;\ drive \\
spd & : & (?\tau.v \le \mathbf{m}.r;\ \tau.a := *;\ ?-b \le \tau.a \le A) \\
 & & \cup(?\tau.v \ge \mathbf{m}.r;\ \tau.a := *;\ ?0 > \tau.a \ge -b) \\
atp & : & SB := \frac{\tau.v^2 - \mathbf{m}.d^2}{2b} + \left(\frac{A}{b} + 1\right)\left(\frac{A}{2}\varepsilon^2 + \varepsilon\ \tau.v\right); \\
 & : & \mathtt{if}(\mathbf{m}.e - \tau.p \le SB \lor rbc.message = emergency)\ \tau.a := -b \\
drive & : & t := 0;\ (\tau.p' = \tau.v, \tau.v' = \tau.a, t' = 1 \land \tau.v \ge 0 \land t \le \varepsilon) \\
rbc & : & (rbc.message := emergency) \\
 & & \cup\ \big(\mathbf{m}_0 := \mathbf{m}; \mathbf{m} := *; \\
 & & \quad ?\mathbf{m}_0.d^2 - \mathbf{m}.d^2 \le 2b(\mathbf{m}.e - \mathbf{m}_0.e) \land \mathbf{m}.r \ge 0 \land \mathbf{m}.d \ge 0\big)
\end{array}
$$

Specification

$$\tau.v^2 - \mathbf{m}.d^2 \le 2b(\mathbf{m}.e - \tau.p) \to [ETCS_r](\tau.p \ge \mathbf{m}.e \to \tau.v \le \mathbf{m}.d)$$

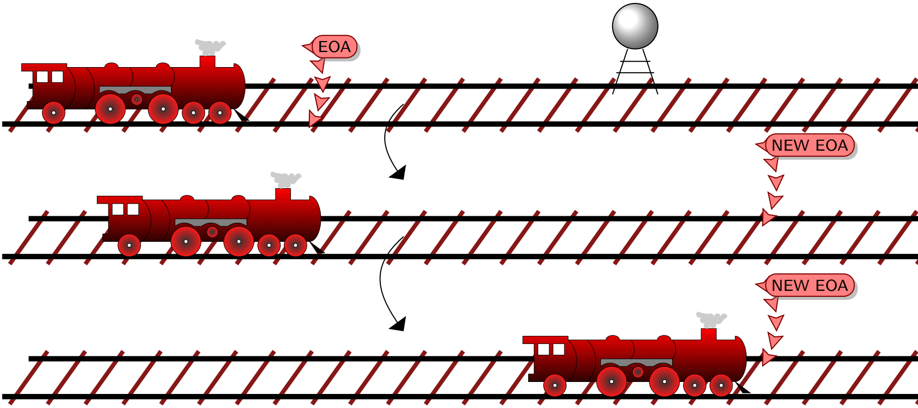
ETCS Safety



Proposition (Safety)

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

ETCS Liveness



Proposition (Liveness)

$$\tau.v \geq 0 \land \varepsilon > 0 \rightarrow \forall P \, \langle ETCS_r \rangle \, \tau.p \geq P$$

Safety Despite Disturbances

So far: no wind, friction, etc.

Direct control of the acceleration

Safety Despite Disturbances

So far: no wind, friction, etc.

Direct control of the acceleration

Issue

This is unrealistic!

Safety Despite Disturbances

So far: no wind, friction, etc.

Direct control of the acceleration

Issue

This is unrealistic!

Solution Take disturbances into account.

Theorem

ETCS is controllable, reactive, and safe in the presence of disturbances.

Safety Despite Disturbances

So far: no wind, friction, etc.

Direct control of the acceleration

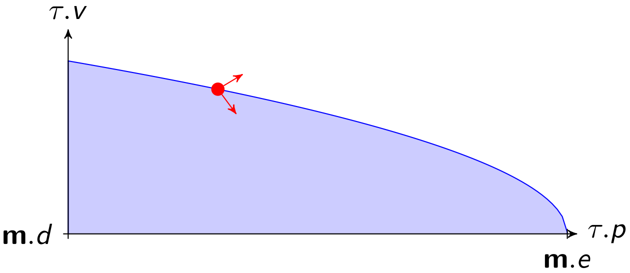
Issue

This is unrealistic!

Solution Take disturbances into account.

Theorem

ETCS is controllable, reactive, and safe in the presence of disturbances.



Safety Despite Disturbances

So far: no wind, friction, etc.

Direct control of the acceleration

Issue

This is unrealistic!

Solution Take disturbances into account.

Theorem

ETCS is controllable, reactive, and safe in the presence of disturbances.

Proof sketch

The system now contains $\tau.a - l \leq \tau.v' \leq \tau.a + u$ instead of $\tau.v' = \tau.a$.

$\rightsquigarrow$ We cannot solve the differential equations anymore.

$\rightsquigarrow$ Use differential invariants for approximation. For details see paper.

Platzer, A.:
Differential-algebraic dynamic logic for differential-algebraic programs.
J. Log. Comput. (2008) DOI 10.1093/logcom/exn070.

Realistic Speed Control

So far

Almost completely non-deterministic control.

Realistic Speed Control

So far

Almost completely non-deterministic control.

Issue

This is unrealistic!

Realistic Speed Control

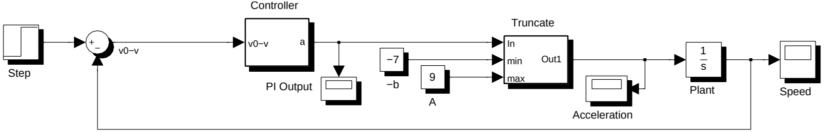
So far

Almost completely non-deterministic control.

Issue

This is unrealistic!

Solution Verify proportional-integral (PI) controllers used in trains.

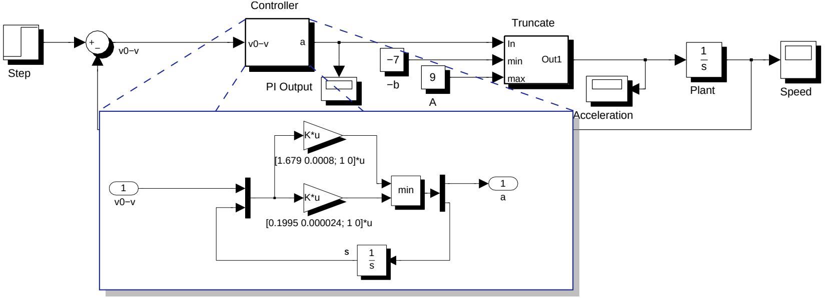


Realistic Speed Control

So far
Almost completely non-deterministic control.

Issue
This is unrealistic!

Solution Verify proportional-integral (PI) controllers used in trains.

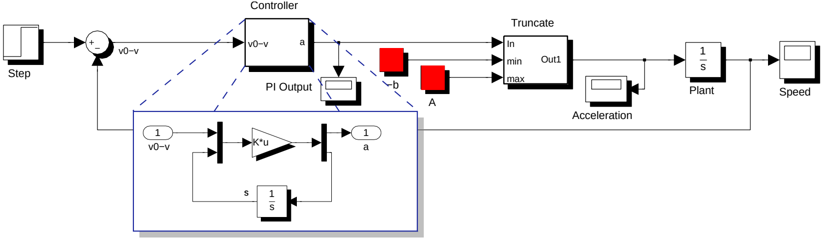


Realistic Speed Control

So far
Almost completely non-deterministic control.

Issue
This is unrealistic!

Solution Verify proportional-integral (PI) controllers used in trains.



Differential equation system

$$\tau.v' = \min\Big(A, \max\big(-b,\ l(\tau.v - \mathbf{m}.r) - i\,s - c\,\mathbf{m}.r\big)\Big) \land s' = \tau.v - \mathbf{m}.r$$

Realistic Speed Control

So far

Almost completely non-deterministic control.

Issue

This is unrealistic!

Solution Verify proportional-integral (PI) controllers used in trains.

Theorem

The ETCS system remains safe when speed is controlled by a PI controller.

Proof sketch

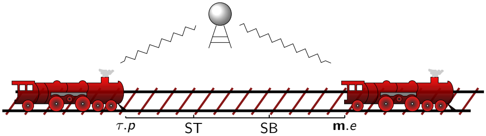
Cannot solve differential equations really. Differential invariants are to be used. For details see paper.

Platzer, A.:
Differential-algebraic dynamic logic for differential-algebraic programs.
J. Log. Comput. (2008) DOI 10.1093/logcom/exn070.

Experimental Results (KeYmaera)

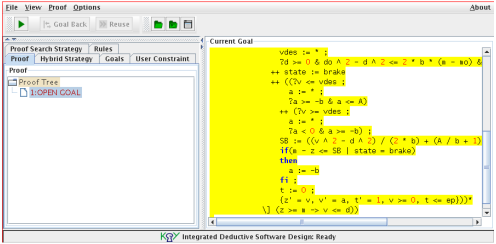
Case study	Int	Time(s)	Steps	Dim
Controllability	0	1.3	14	5
RBC Controllability	0	1.7	42	12
RBC Control (characterization)	0	2.2	42	12
Reactivity (existence)	8	133.4	229	13
Reactivity	0	86.8	52	14
Safety	0	249.9	153	14
Liveness	4	27.3	166	7
Inclusion (PI)	19	766.2	301	25
Safety (PI)	16	509.0	183	15
Controllability (disturbed)	0	5.6	37	7
Reactivity (disturbed)	2	34.6	78	15
Safety (disturbed)	5	389.9	88	16

Summary



Formally verified a major case study with KeYmaera:

- discovered necessary safety constraints
- controllability, reactivity, safety and liveness properties
- Extensions for ETCS with disturbances and for ETCS with PI control



Literature

Platzer, A.:
Differential-algebraic dynamic logic for differential-algebraic programs.
J. Log. Comput. (2008) DOI 10.1093/logcom/exn070.

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.

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.

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.

Syntax of Differential Dynamic Logic

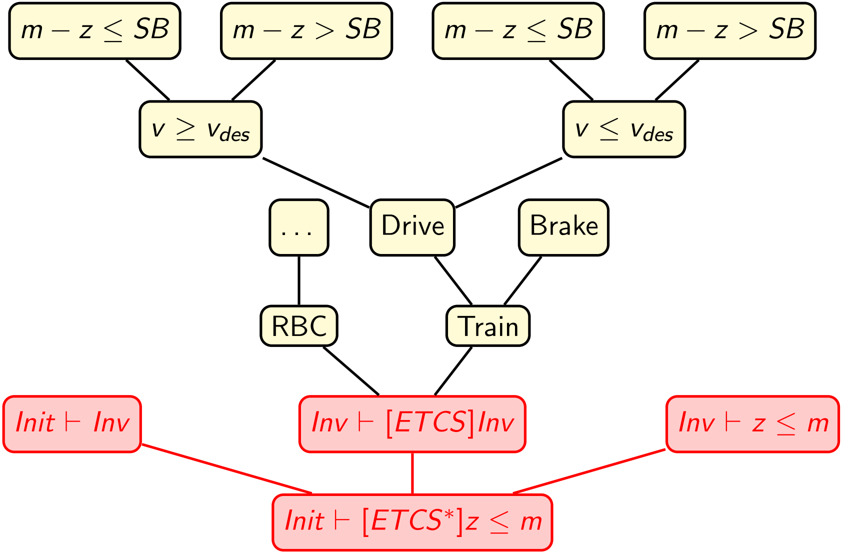
d$\mathcal{L}$ Formulas

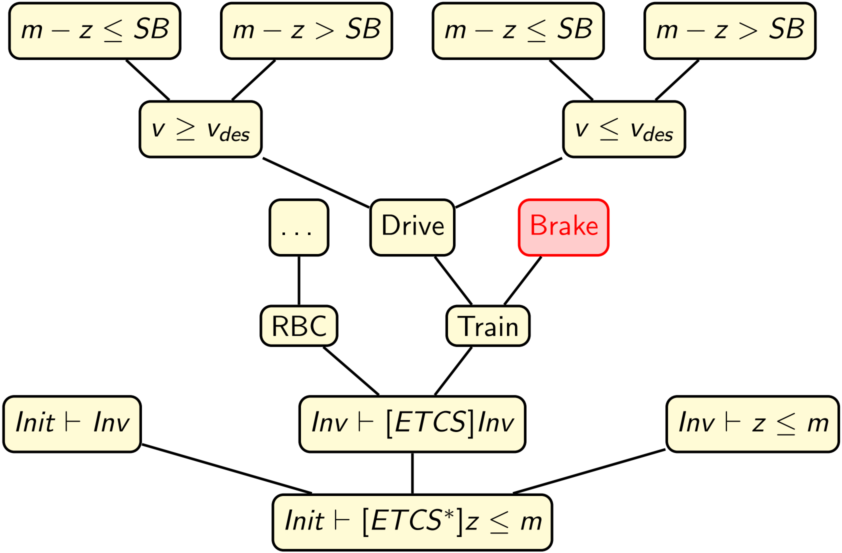
$$\phi ::= \theta_1 \sim \theta_2 \mid \neg\phi \mid \phi \wedge \psi \mid \forall x \phi \mid \exists x \phi \mid [\alpha]\phi \mid \langle\alpha\rangle\phi$$

Hybrid Program	Effect
$\alpha;\ \beta$	sequential composition
$\alpha \cup \beta$	nondeterministic choice
α^*	nondeterministic repetition
$x := \theta$	discrete assignment (jump)
$x := *$	nondeterministic assignment
$(x_1' = \theta_1, \ldots, x_n' = \theta_n, F)$	continuous evolution of x_i
$?F$	check if formula F holds

A. Platzer.
Differential Dynamic Logic for Hybrid Systems.
Journal of Automated Reasoning, 41(2), 2008.

Proof Sketch



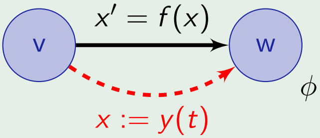


$m - z \leq SB$
$m - z > SB$
$m - z \leq SB$
$m - z > SB$
$v \geq v_{des}$
$v \leq v_{des}$
...
Drive
Brake
RBC
Train
$Init \vdash Inv$
$Inv \vdash [ETCS]Inv$
$Inv \vdash z \leq m$
$Init \vdash [ETCS^*]z \leq m$

Handling Differential Equations

Example

$$\frac{\forall t \geq 0\ [x := y(t)]\,\phi}{[x' = f(x)]\,\phi}$$

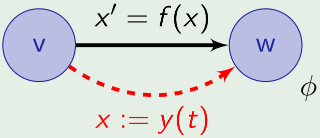


$$\ldots \vdash [z' = v, v' = -b]z \leq m$$

Handling Differential Equations

Example

$$\frac{\forall t \geq 0\ [x := y(t)]\,\phi}{[x' = f(x)]\,\phi}$$



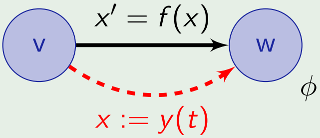
$$\ldots \vdash \forall t \geq 0\ [z := -\tfrac{1}{2}bt^2 + tv + z]z \leq m$$

$$\ldots \vdash [z' = v, v' = -b]z \leq m$$

Handling Differential Equations

Example

$$\frac{\forall t \geq 0\ [x := y(t)]\,\phi}{[x' = f(x)]\,\phi}$$



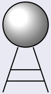
$$\ldots \vdash \forall t \geq 0\ (-\tfrac{1}{2}bt^2 + tv + z \leq m)$$
$$\ldots \vdash \forall t \geq 0\ [z := -\tfrac{1}{2}bt^2 + tv + z]z \leq m$$
$$\ldots \vdash [z' = v, v' = -b]z \leq m$$

Model/State Variables

Train τ ()

- $\tau.p$ Position
- $\tau.v$ Speed
- $\tau.a$ Acceleration
- (t model time)

RBC + MA



- $\mathbf{m}.e$ End of Authority
- $\mathbf{m}.d$ Speed limit
- $\mathbf{m}.r$ Recommended speed
- $rbc.message$ Channel

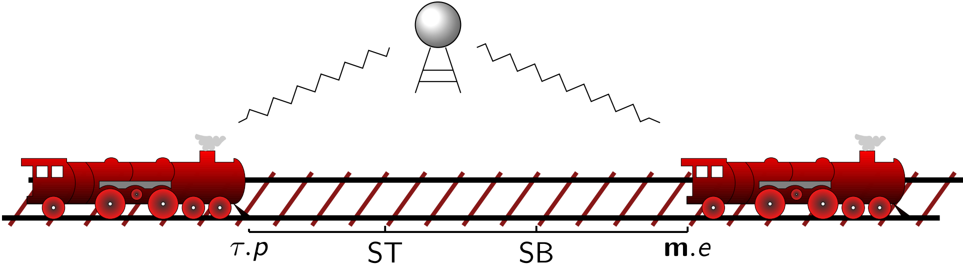
Parameters

- SB Start Braking
- b Braking power/deceleration
- A Maximum acceleration
- ε Maximum cycle time

Separation Principle

Lemma (Principle of separation by movement authorities)

Each train respects its movement authority and
the RBC partitions into disjoint movement authorities
$\Rightarrow$ *trains can never collide.*



Separation Principle

Lemma (Principle of separation by movement authorities)

Each train respects its movement authority and
the RBC partitions into disjoint movement authorities
$\Rightarrow$ *trains can never collide.*

Proof.

□

Separation Principle

Lemma (Principle of separation by movement authorities)

Each train respects its movement authority and
the RBC partitions into disjoint movement authorities
$\Rightarrow$ *trains can never collide.*

Proof.

- To simplify notation, assume trains are points.

□

Separation Principle

Lemma (Principle of separation by movement authorities)

Each train respects its movement authority and
the RBC partitions into disjoint movement authorities
$\Rightarrow$ *trains can never collide.*

Proof.

- To simplify notation, assume trains are points.
- Consider any point in time ζ.

□

Separation Principle

Lemma (Principle of separation by movement authorities)

Each train respects its movement authority and
the RBC partitions into disjoint movement authorities
$\Rightarrow$ *trains can never collide.*

Proof.

- To simplify notation, assume trains are points.
- Consider any point in time ζ.
- For $n \in \mathbb{N}$, let $z_1, \ldots, z_n$ be positions of all the trains 1 to n at ζ.

□

Separation Principle

Lemma (Principle of separation by movement authorities)

Each train respects its movement authority and
the RBC partitions into disjoint movement authorities
$\Rightarrow$ *trains can never collide.*

Proof.

- To simplify notation, assume trains are points.
- Consider any point in time ζ.
- For $n \in \mathbb{N}$, let $z_1, \ldots, z_n$ be 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.

□

Separation Principle

Lemma (Principle of separation by movement authorities)

Each train respects its movement authority and
the RBC partitions into disjoint movement authorities
$\Rightarrow$ *trains can never collide.*

Proof.

- To simplify notation, assume trains are points.
- Consider any point in time ζ.
- For $n \in \mathbb{N}$, let $z_1, \ldots, z_n$ be 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.

□

Separation Principle

Lemma (Principle of separation by movement authorities)

Each train respects its movement authority and
the RBC partitions into disjoint movement authorities
$\Rightarrow$ *trains can never collide.*

Proof.

- To simplify notation, assume trains are points.
- Consider any point in time ζ.
- For $n \in \mathbb{N}$, let $z_1, \ldots, z_n$ be 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 $i, j \in \mathbb{N}$.

□

Separation Principle

Lemma (Principle of separation by movement authorities)

Each train respects its movement authority and
the RBC partitions into disjoint movement authorities
$\Rightarrow$ *trains can never collide.*

Proof.

- To simplify notation, assume trains are points.
- Consider any point in time ζ.
- For $n \in \mathbb{N}$, let $z_1, \ldots, z_n$ be 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 $M_i \cap M_j \neq \emptyset$,

□

Separation Principle

Lemma (Principle of separation by movement authorities)

Each train respects its movement authority and
the RBC partitions into disjoint movement authorities
$\Rightarrow$ *trains can never collide.*

Proof.

- To simplify notation, assume trains are points.
- Consider any point in time ζ.
- For $n \in \mathbb{N}$, let $z_1, \ldots, z_n$ be 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 $M_i \cap M_j \neq \emptyset$,
- This contradicts the assumption of disjoint MA. □