

Formal Verification of Train Control with Air Pressure Brakes

Stefan Mitsch¹ Marco Gario² Christof J. Budnik²
Michael Golm² André Platzer¹

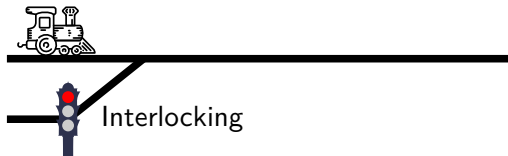
¹Computer Science Department, Carnegie Mellon University

²Siemens Corporate Technology, Princeton, NJ, USA

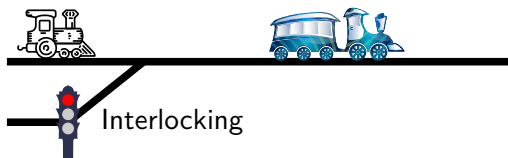
Reliability, Safety and Security of Railway Systems
November 15, 2017



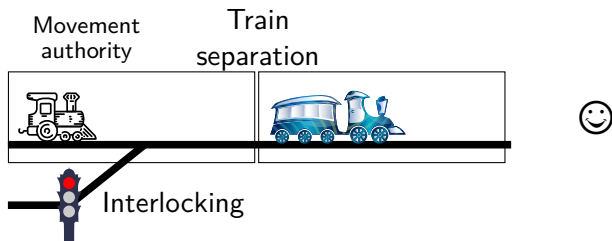
Railroad Safety: Train Separation and Train Control



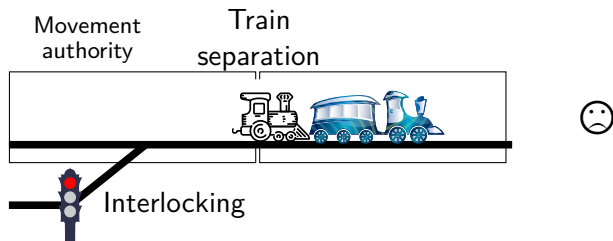
Railroad Safety: Train Separation and Train Control



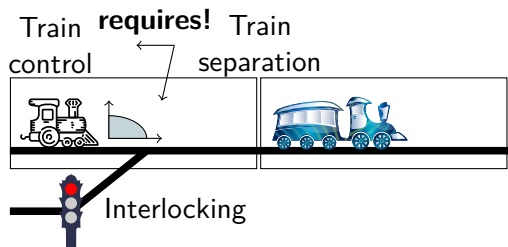
Railroad Safety: Train Separation and Train Control



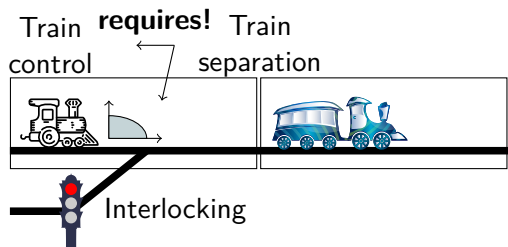
Railroad Safety: Train Separation and Train Control



Railroad Safety: Train Separation and Train Control



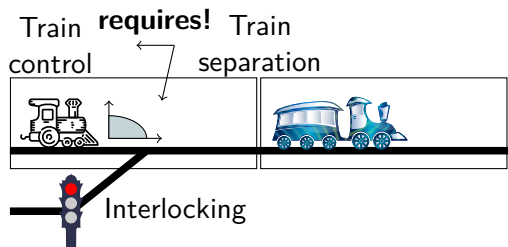
Railroad Safety: Train Separation and Train Control



Design **provably safe train control** considering physical train motion

- ▶ Federal Railroad Administration (FRA): motion and brake models
- ▶ No overshoot
- ▶ Limited undershoot

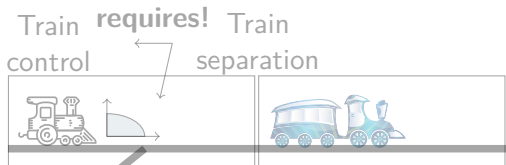
Railroad Safety: Train Separation and Train Control



Design **provably safe train control** considering physical train motion

- ▶ Federal Railroad Administration (FRA): motion and brake models
- ▶ No overshoot
- ▶ Limited undershoot
- ▶ **But underspecified control conditions**

Railroad Safety: Train Separation and Train Control



Approach

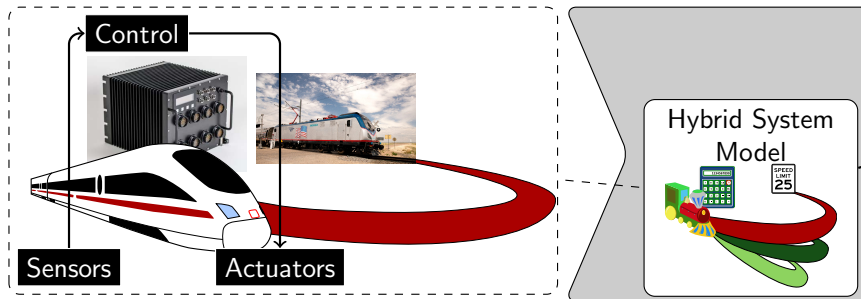
**Safe train separation
requires verified train control and motion!**

Design **provably safe train control** considering physical train motion

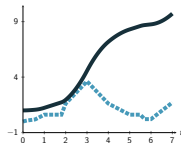
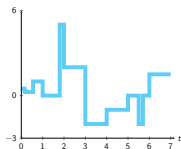
- ▶ Federal Railroad Administration (FRA): motion and brake models
- ▶ No overshoot
- ▶ Limited undershoot
- ▶ **But underspecified control conditions**

Approach: Hybrid Systems Theorem Proving

Analyze the physical effect of software

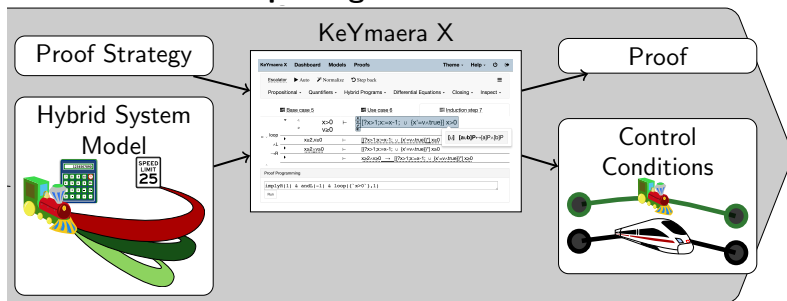


Discrete computation + continuous physics



Approach: Hybrid Systems Theorem Proving

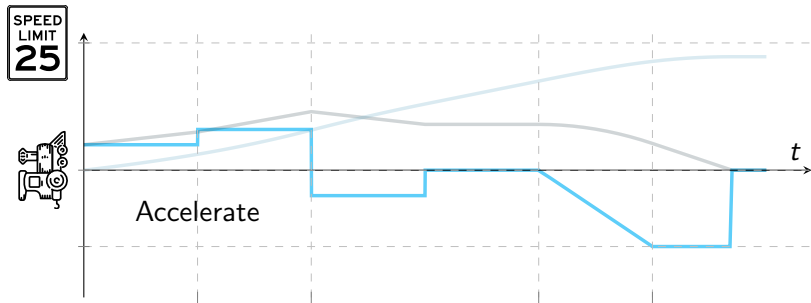
Theorem proving ensures correct model



Main results for

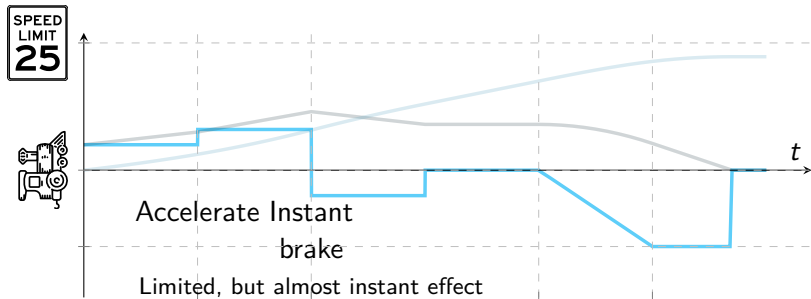
- ▶ **Certification:** Proofs
- ▶ System **architecture** and **implementation:** Models
- ▶ Control **engineering** and **testing:** Control conditions

Train Motion and Brake Model

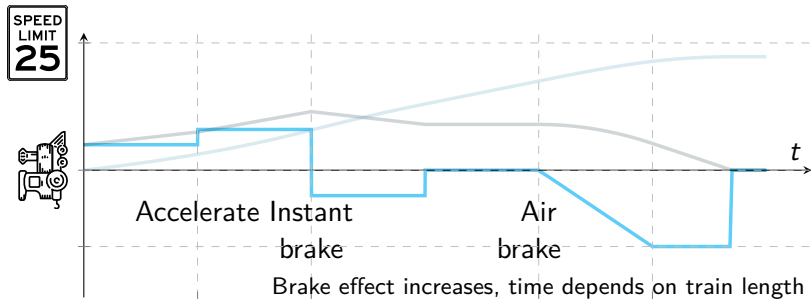


Brossaeu, J., Ede, B.M.: Development of an adaptive predictive braking enforcement algorithm. FRA/DOT/ORD-9/13 (2009)

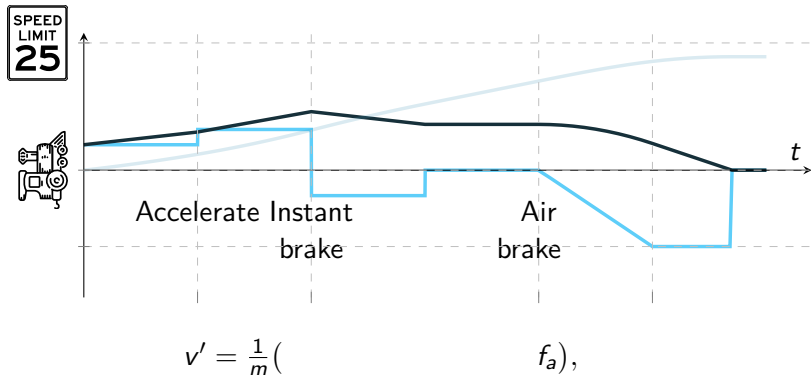
Train Motion and Brake Model



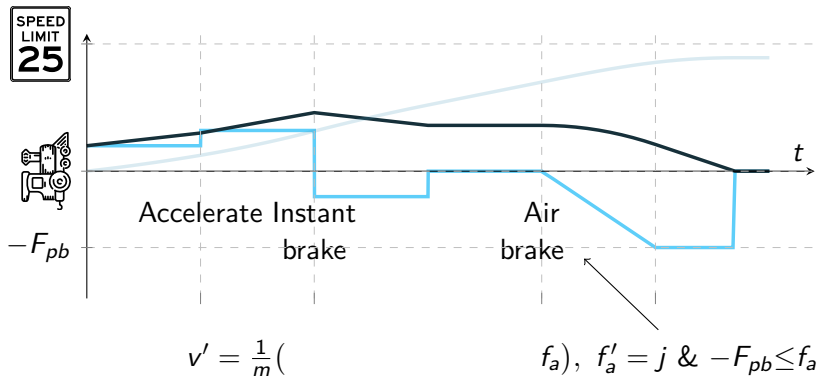
Train Motion and Brake Model



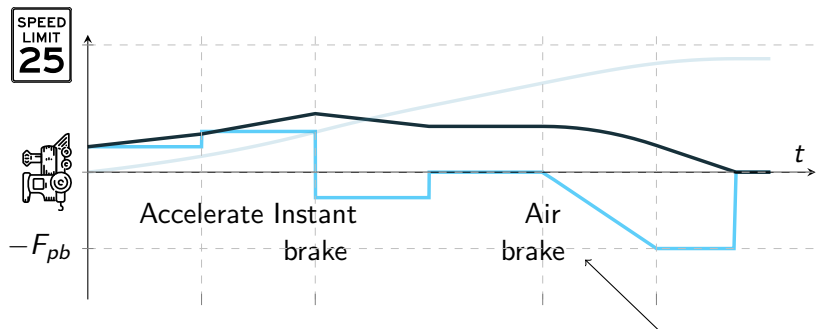
Train Motion and Brake Model



Train Motion and Brake Model

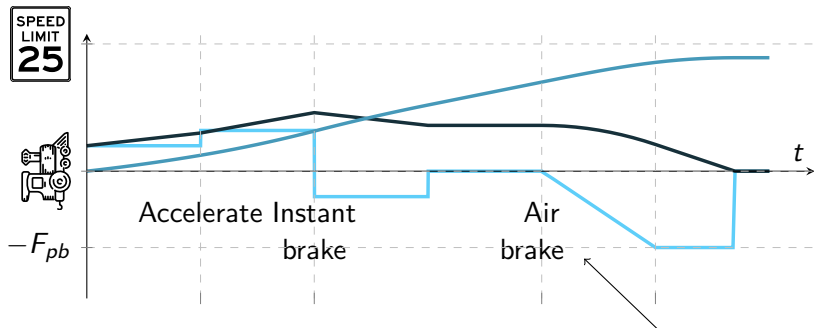


Train Motion and Brake Model



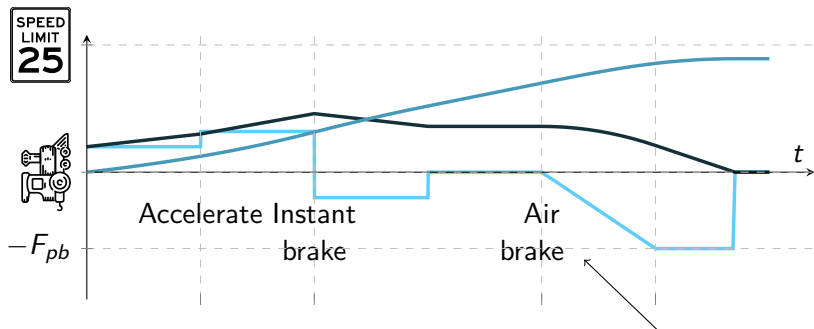
$$v' = \frac{1}{m} (-(F_g + F_r + F_c) + f_a), \quad f'_a = j \ \& \ -F_{pb} \leq f_a$$

Train Motion and Brake Model



$$x' = v, \quad v' = \frac{1}{m}(- (F_g + F_r + F_c) + f_a), \quad f'_a = j \ \& \ -F_{pb} \leq f_a$$

Train Motion and Brake Model

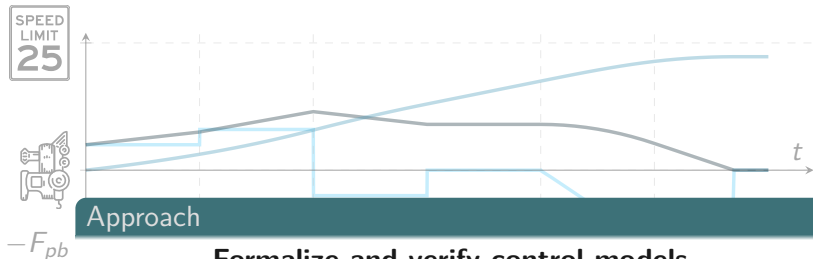


$$x' = v, \quad v' = \frac{1}{m} (-(F_g + F_r + F_c) + f_a), \quad f_a' = j \quad \& \quad -F_{pb} \leq f_a$$

Underspecified:

- ▶ What are safe control choices?
- ▶ How important is brake model fidelity?

Train Motion and Brake Model



$$x' = v, \quad v' = \frac{1}{m}(- (F_g + F_r + F_c) + f_a), \quad f_a' = j \quad \& \quad -F_{pb} \leq f_a$$

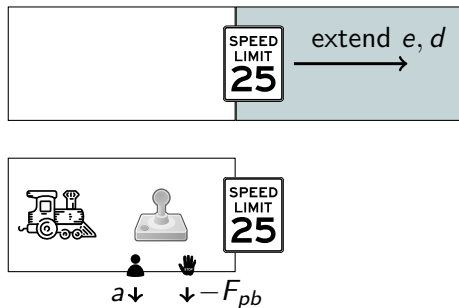
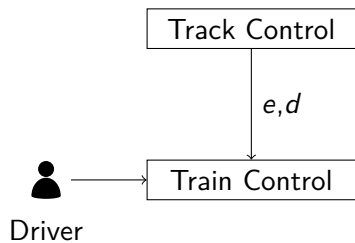
Underspecified:

- ▶ What are safe control choices? _____
- ▶ How important is brake model fidelity? _____

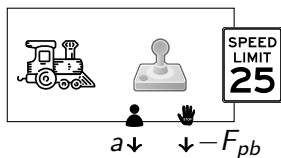
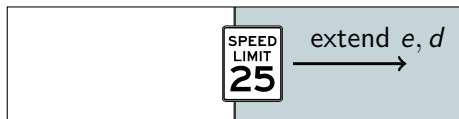
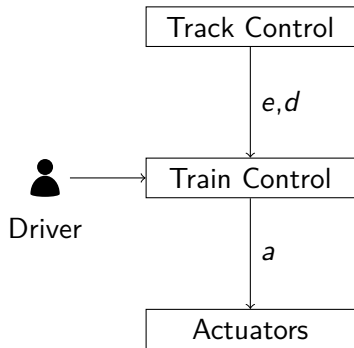
Control Model





Control Model

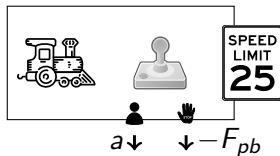
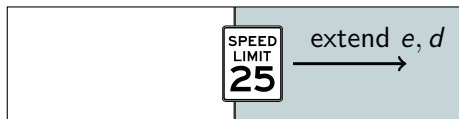
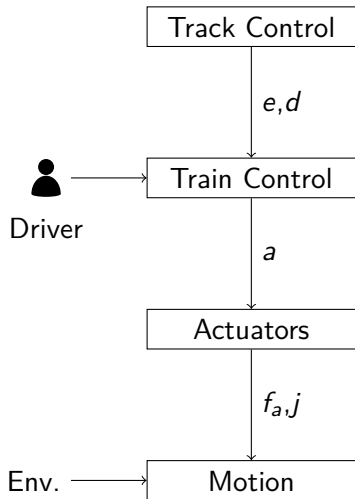




Control Model



Delay  vs. air brake 

Control Model

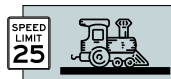


Delay  vs. air brake 

Formal Verification with dL: No Overshoot

Correctness property: **respect the speed limit**

$$\text{safe} \equiv (z \geq e \rightarrow v \leq 25)$$



Formal Verification with $d\mathcal{L}$: No Overshoot

Correctness property: **respect the speed limit**

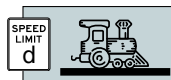
$$\text{safe} \equiv (z \geq e \rightarrow v \leq d)$$



Formal Verification with $d\mathcal{L}$: No Overshoot

Correctness property: **respect the speed limit**

$$\text{safe} \equiv (z \geq e \rightarrow v \leq d)$$

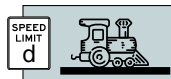


[]safe

Formal Verification with $d\mathcal{L}$: No Overshoot

Correctness property: **respect the speed limit**

$$\text{safe} \equiv (z \geq e \rightarrow v \leq d)$$

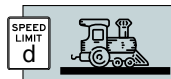


$$\left[\left(\text{joystick}; \text{graph}; \text{train} \right)^* \right] \text{safe}$$

Formal Verification with $d\mathcal{L}$: No Overshoot

Correctness property: **respect the speed limit**

$$\text{safe} \equiv (z \geq e \rightarrow v \leq d)$$

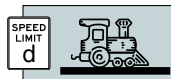


$$\left[\left(\left[\begin{array}{c} \text{SPEED} \\ \text{LIMIT} \\ d \end{array} \right] \cup \text{joystick} ; \text{graph} ; \text{train} \right)^* \right] \text{safe}$$

Formal Verification with dL: No Overshoot

Correctness property: **respect the speed limit**

$$\text{safe} \equiv (z \geq e \rightarrow v \leq d)$$

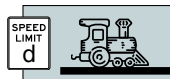


$$\text{init} \rightarrow \left[\left(\begin{array}{c} \text{SPEED} \\ \text{LIMIT} \\ d \end{array} \cup \begin{array}{c} \text{joystick} \\ \text{stick} \end{array} ; \begin{array}{c} \text{graph} \\ \text{line} \end{array} ; \begin{array}{c} \text{train} \\ \text{locomotive} \end{array} \right)^* \right] \text{safe}$$

Formal Verification with $d\mathcal{L}$: No Overshoot

Correctness property: **respect the speed limit**

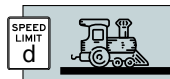
$$\text{safe} \equiv (z \geq e \rightarrow v \leq d)$$



$$\text{init} \rightarrow \left[\left(\begin{array}{c} \text{SPEED} \\ \text{LIMIT} \\ d \end{array} \right) \cup \underbrace{\left(\text{person} \cup \text{STOP} \right)}_{\text{signal}} ; \begin{array}{c} \text{graph} \\ \text{train} \end{array} \right]^* \text{safe}$$

Formal Verification with dL: No Overshoot

Correctness property: **respect the speed limit**



$$\text{safe} \equiv (z \geq e \rightarrow v \leq d)$$

$$\text{init} \rightarrow \left[\left(\begin{array}{c} \text{SPEED LIMIT} \\ d \end{array} \cup \underbrace{(\text{person} \cup \text{STOP})}_{\text{lock}}; \begin{array}{c} \text{graph} \\ \text{train} \end{array} \right)^* \right] \text{safe}$$

$$\text{person} \equiv f_a := *; ? - F_{sb} \leq f_a \leq A; ? \left(e - z \geq \frac{(v^2 - d^2)m}{2F_{sb}} + \left(\frac{A}{F_{sb}} + 1 \right) \left(\frac{A}{2m} \epsilon^2 + v\epsilon \right) \right) \quad (1)$$

$$\vee \text{slow}^- \vee \text{slow}^+ \vee \text{fast}^- \vee \text{fast}^+ \quad (2)$$

$$\text{slow}^- \equiv \neg \text{isFast}(v) \wedge f_a \leq 0 \wedge e - z \geq v\epsilon + m\text{Slow}(v) \quad (3)$$

$$\text{slow}^+ \equiv [u := v + \frac{f_a \epsilon}{m}] \left(\neg \text{isFast}(u) \wedge f_a \geq 0 \wedge e - z \geq v\epsilon + \frac{f_a \epsilon^2}{2m} + m\text{Slow}(u) \right) \quad (4)$$

$$\text{fast}^- \equiv \text{isFast}(v) \wedge f_a \leq 0 \wedge e - z \geq v\epsilon + m\text{Fast}(v) \quad (5)$$

$$\text{fast}^+ \equiv \text{isFast}(v) \wedge f_a \geq 0 \wedge e - z \geq v\epsilon + \frac{f_a \epsilon^2}{2m} + m\text{Fast} \left(v + \frac{f_a \epsilon}{m} \right) \quad (6)$$

$$\text{isFast}(v) \equiv v \geq \frac{F_{pb}^2}{2mJ} \quad m\text{Slow}(v) = \frac{2}{3} v \sqrt{2mv/J} \quad m\text{Fast}(v) = \frac{mv^2}{2F_{pb}} + \frac{vF_{pb}}{2J} - \frac{F_{pb}^3}{24mJ^2} \quad (7)$$

Formal Verification with $d\mathcal{L}$: No Overshoot

Correctness property: **respect the speed limit**



$$\text{safe} \equiv (z \geq e \rightarrow v \leq d)$$

$$\text{init} \rightarrow \left[\left(\begin{array}{c} \text{SPEED LIMIT} \\ d \end{array} \cup \underbrace{(\text{person} \cup \text{STOP})}_{\text{driver}}; \text{graph}; \text{train} \right)^* \right] \text{safe}$$

Result

Driving with verified control conditions ensures safety

How to find conditions: Proofs!

(1)

(2)

$$\text{slow}^- \equiv \neg \text{isFast}(v) \wedge f_a \leq 0 \wedge e - z \geq v\varepsilon + m\text{Slow}(v) \quad (3)$$

$$\text{slow}^+ \equiv [u := v + \frac{f_a \varepsilon}{m}] \left(\neg \text{isFast}(u) \wedge f_a \geq 0 \wedge e - z \geq v\varepsilon + \frac{f_a \varepsilon^2}{2m} + m\text{Slow}(u) \right) \quad (4)$$

$$\text{fast}^- \equiv \text{isFast}(v) \wedge f_a \leq 0 \wedge e - z \geq v\varepsilon + m\text{Fast}(v) \quad (5)$$

$$\text{fast}^+ \equiv \text{isFast}(v) \wedge f_a \geq 0 \wedge e - z \geq v\varepsilon + \frac{f_a \varepsilon^2}{2m} + m\text{Fast} \left(v + \frac{f_a \varepsilon}{m} \right) \quad (6)$$

$$\text{isFast}(v) \equiv v \geq \frac{F_{pb}^2}{2mJ} \quad m\text{Slow}(v) = \frac{2}{3}v \sqrt{2mv/J} \quad m\text{Fast}(v) = \frac{mv^2}{2F_{pb}} + \frac{vF_{pb}}{2J} - \frac{F_{pb}^3}{24mJ^2} \quad (7)$$

Systematically Derive Safe Control Conditions in $d\mathcal{L}$

- ▶ **Partial model, unknown condition ?true**

$\left[\text{person} ; ?\text{true} ; \text{train} \right] \text{safe}$

Systematically Derive Safe Control Conditions in dL

- ▶ **Partial model, unknown condition ?true**

$\left[\text{person} ; ?\text{true} ; \text{train} \right] \text{safe}$

- ▶ **Run and observe “parallel universe”**

$\left[\text{person} ; \underbrace{?[\text{train}] \text{safe}} ; \text{train} \right] \text{safe}$

test for desired outcome



↓ copy&paste



Systematically Derive Safe Control Conditions in dL

- ▶ **Partial model, unknown condition ?true**

$\left[\text{person} ; ?\text{true} ; \text{train} \right] \text{safe}$

- ▶ **Run and observe “parallel universe”**

$\left[\text{person} ; \underbrace{?[\text{train}] \text{safe}} ; \text{train} \right] \text{safe}$

test for desired outcome

- ▶ Obviously true but not helpful for implementation

$\left[\text{person} \right] \left([\text{train}] \text{safe} \rightarrow [\text{train}] \text{safe} \right)$



↓ copy&paste



Systematically Derive Safe Control Conditions in dL

- ▶ **Partial model, unknown condition ?true**

$$\left[\text{person} ; ?\text{true} ; \text{train} \right] \text{safe}$$

- ▶ **Run and observe “parallel universe”**

$$\left[\text{person} ; \underbrace{?[\text{train}] \text{safe}}_{\text{test for desired outcome}} ; \text{train} \right] \text{safe}$$

test for desired outcome

- ▶ Obviously true but not helpful for implementation

$$\left[\text{person} \right] \left([\text{train}] \text{safe} \rightarrow [\text{train}] \text{safe} \right)$$

- ▶ **Symbolically execute program with dL**

$$\left[\text{person} \right] \left(\text{safe} \left(z + vt + \frac{f_a}{2m} t^2, v + \frac{f_a}{m} t \right) \rightarrow [\text{train}] \text{safe} \right)$$



↓ copy&paste



$$\text{train} \equiv \sum_{i=1}^n \frac{v_i^{(end)}}{a}$$

Systematically Derive Safe Control Conditions in dL

- ▶ **Partial model, unknown condition ?true**

$\left[\text{person} ; ?\text{true} ; \text{train} \right] \text{safe}$

- ▶ **Run and observe “parallel universe”**

$\left[\text{person} ; \underbrace{?[\text{train}] \text{safe}}_{\text{test for desired outcome}} ; \text{train} \right] \text{safe}$

test for desired outcome

- ▶ Obviously true but not helpful for implementation

$\left[\text{person} \right] \left([\text{train}] \text{safe} \rightarrow [\text{train}] \text{safe} \right)$

- ▶ **Symbolically execute program with dL**

$\left[\text{person} \right] \left(\text{safe} \left(z + vt + \frac{f_a}{2m} t^2, v + \frac{f_a}{m} t \right) \rightarrow [\text{train}] \text{safe} \right)$

- ▶ **Use program effects as control conditions**

$\left[\text{person} ; ?\text{safe} \left(z + vt + \frac{f_a}{2m} t^2, v + \frac{f_a}{m} t \right) ; \text{train} \right] \text{safe}$



↓ copy&paste



$$\text{train} \equiv \sum_{i=1}^n \frac{v_i}{\omega}$$

Implementable

Systematically Derive Safe Control Conditions in dL

- ▶ Partial model, unknown condition ?true

$[\text{person} ; ?\text{true} ; \text{train}] \text{safe}$

- ▶ Run and observe “parallel universe”

$[\text{person} ; ?[\text{train}] \text{safe} ; \text{train}] \text{safe}$



↓ copy&paste



Result

- ▶ O



**Augment partial model
with implementable control conditions
derived by proof**

- ▶ S_y

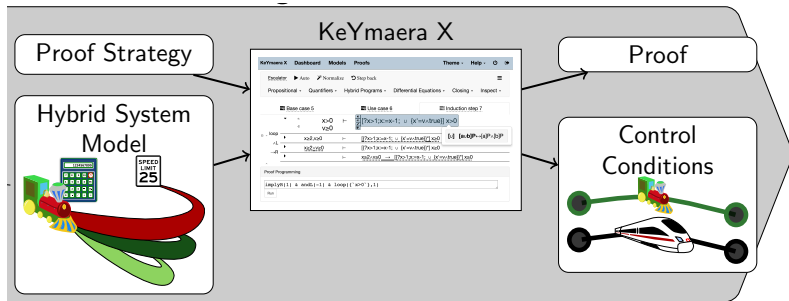
$[\text{person}] \left(\text{safe} \left(z + vt + \frac{f_a}{2m} t^2, v + \frac{f_a}{m} t \right) \rightarrow [\text{train}] \text{safe} \right)$

$\frac{v^2}{2} + \frac{a(n+1)}{2}$

- ▶ Use program effects as control conditions

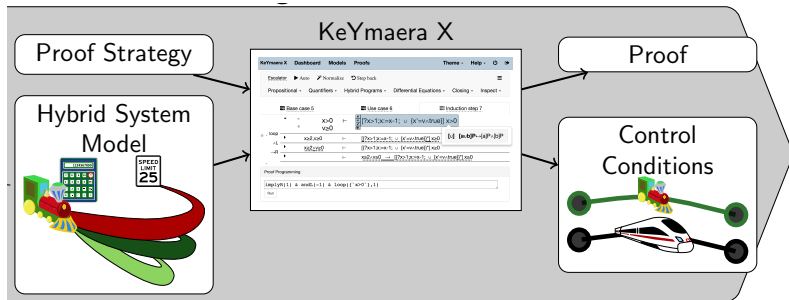
$[\text{person} ; ?\text{safe} \left(z + vt + \frac{f_a}{2m} t^2, v + \frac{f_a}{m} t \right) ; \text{train}] \text{safe}$

Implementable



No overshoot ✓

- ▶ Verified models: safely control brake delay and air brakes
- ▶ Symbolic control conditions to select between free driving and braking



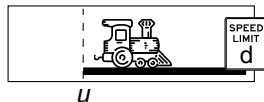
No overshoot ✓ + Limited undershoot

- ▶ Verified models: safely control brake delay and air brakes
- ▶ Symbolic control conditions to select between free driving and braking

Formal Verification with $d\mathcal{L}$: Limited Undershoot

Correctness: **when done braking, train is after undershoot limit**

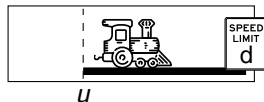
efficient $\equiv (\text{brakesEngaged} \wedge v \leq d) \rightarrow (z \geq e - u)$



Formal Verification with $d\mathcal{L}$: Limited Undershoot

Correctness: **when done braking, train is after undershoot limit**

efficient $\equiv (\text{brakesEngaged} \wedge v \leq d) \rightarrow (z \geq e - u)$



Needs change in control priority

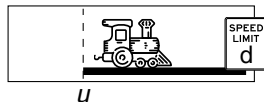
- ▶ No overshoot: braking allowed but spoils efficiency

init $\rightarrow \left[\left(\left(\text{person} \cup \text{stop_hand} \right) ; \text{signal_L} ; \text{train} \right)^* \right] \text{efficient}$

Formal Verification with $d\mathcal{L}$: Limited Undershoot

Correctness: **when done braking, train is after undershoot limit**

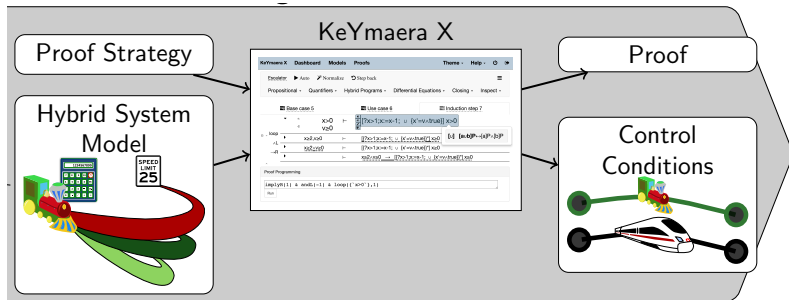
efficient $\equiv (\text{brakesEngaged} \wedge v \leq d) \rightarrow (z \geq e - u)$



Needs change in control priority

- ▶ No overshoot: braking allowed but spoils efficiency
- ▶ Limited undershoot: only brake if necessary for safety

$$\text{init} \rightarrow \left[\left((\text{if } (\text{mustBrake}) \text{ STOP} \text{ else } \text{person}) ; \text{L} ; \text{train} \right)^* \right] \text{efficient}$$



No overshoot ✓ + Limited undershoot ✓

- ▶ Verified models: safely control brake delay and air brakes
- ▶ Symbolic control conditions to select between free driving and braking
- ▶ Control favors free driving for efficiency

Experiments

Compare **brake engage distance to endpoint**

- ▶ control assuming delayed brakes
- ▶ control assuming air brakes



Instantiate symbolic models with concrete parameters from FRA

| | Parameter | Value | vs. |
|------------|------------------|--|------------------|
| l_z | Length | Medium = 2 345ft (40 cars) | short,long |
| m | Mass | Loaded = 10 520klb ($263 \frac{\text{klb}}{\text{car}}$) | empty |
| v | Speed | Fast = 60mph | slow |
| F_{pb} | Emergency brake | Loaded = 1 430klbf ($35 750 \frac{\text{lb}}{\text{car}}$) | empty,unknown |
| t_{appl} | Time | 50 s | length-dependent |
| A | Acceleration | $5 \frac{\text{mph}}{\text{min}} = 391.91\text{klbf}$ | |
| f_a | Brake force | $1.75 \frac{\text{mph}}{\text{min}} = 136.76\text{klbf}$ | |
| ϵ | Control cycle | 100ms | |

Brossaeu, J., Ede, B.M.: Development of an adaptive predictive braking enforcement algorithm. FRA/DOT/ORD-9/13 (2009)

Experimental Results: Brake Engage Points

| Cars | Slow | | | | | | Fast | | | | | |
|-------------------|--|-------|------------|-------|-----|--------------|--------------|--------------|--------------|--------------|--------------|--------------|
| | Loaded | | | Empty | | | Loaded | | | Empty | | |
| | 10 | 40 | 100 | 10 | 40 | 100 | 10 | 40 | 100 | 10 | 40 | 100 |
| | Brake force for unknown load $F_{pb} = 23\,338 \frac{\text{lbf}}{\text{car}}$ | | | | | | | | | | | |
| Delay brakes | 726 | 1,110 | 1,942 | 446 | 830 | 1,662 | 15,436 | 17,742 | 22,730 | 5,369 | 7,676 | 12,664 |
| Air brakes | 541 | 710 | 1,017 | 239 | 345 | 503 | 14,364 | 15,494 | 17,880 | 4,278 | 5,334 | 7,383 |
| Difference | 185 | 400 | 925 | 207 | 485 | 1,161 | 1,072 | 2,248 | 4,850 | 1,091 | 2,342 | 5,281 |
| | Brake force for known load, loaded: $F_{pb} = 35\,750 \frac{\text{lbf}}{\text{car}}$, empty: $F_{pb} = 10\,575 \frac{\text{lbf}}{\text{car}}$ | | | | | | | | | | | |
| Delay brakes | 597 | 982 | 1,814 | 554 | 939 | 1,771 | 10,817 | 13,123 | 18,111 | 9,277 | 11,583 | 16,571 |
| Air brakes | 409 | 565 | 822 | 364 | 512 | 746 | 9,743 | 10,859 | 13,188 | 8,200 | 9,309 | 11,602 |
| Difference | 188 | 417 | 992 | 190 | 427 | 1,025 | 1,074 | 2,264 | 4,923 | 1,077 | 2,274 | 4,969 |



Experimental Results: Brake Engage Points

| Cars | Slow | | | | | | Fast | | | | | |
|--------------|--|-------|-------|-------|-----|-------|---------------|---------------|---------------|--------------|---------------|---------------|
| | Loaded | | | Empty | | | Loaded | | | Empty | | |
| | 10 | 40 | 100 | 10 | 40 | 100 | 10 | 40 | 100 | 10 | 40 | 100 |
| | Brake force for unknown load $F_{pb} = 23\,338 \frac{\text{lbf}}{\text{car}}$ | | | | | | | | | | | |
| Delay brakes | 726 | 1,110 | 1,942 | 446 | 830 | 1,662 | 15,436 | 17,742 | 22,730 | 5,369 | 7,676 | 12,664 |
| Air brakes | 541 | 710 | 1,017 | 239 | 345 | 503 | 14,364 | 15,494 | 17,880 | 4,278 | 5,334 | 7,383 |
| Difference | 185 | 400 | 925 | 207 | 485 | 1,161 | 1,072 | 2,248 | 4,850 | 1,091 | 2,342 | 5,281 |
| | Brake force for known load, loaded: $F_{pb} = 35\,750 \frac{\text{lbf}}{\text{car}}$, empty: $F_{pb} = 10\,575 \frac{\text{lbf}}{\text{car}}$ | | | | | | | | | | | |
| Delay brakes | 597 | 982 | 1,814 | 554 | 939 | 1,771 | 10,817 | 13,123 | 18,111 | 9,277 | 11,583 | 16,571 |
| Air brakes | 409 | 565 | 822 | 364 | 512 | 746 | 9,743 | 10,859 | 13,188 | 8,200 | 9,309 | 11,602 |
| Difference | 188 | 417 | 992 | 190 | 427 | 1,025 | 1,074 | 2,264 | 4,923 | 1,077 | 2,274 | 4,969 |

Experimental Results: Brake Engage Points

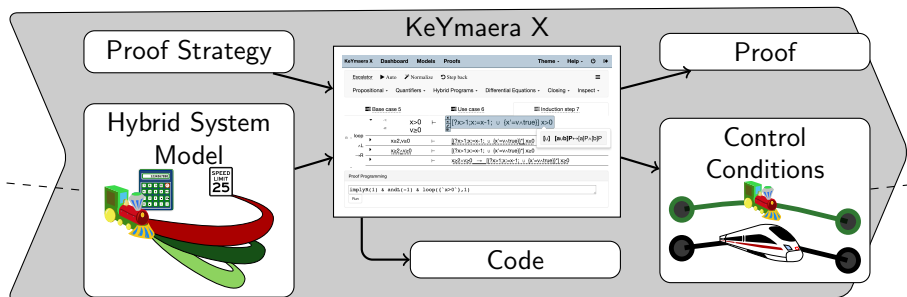
Air brake control conditions engage brakes considerably later

Fast trains

| | Cars | Loaded | | | Empty | | |
|--|------|--------------|--------------|--------------|--------------|--------------|--------------|
| | | 10 | 40 | 100 | 10 | 40 | 100 |
| Brake force for known load, loaded: $F_{pb} = 35\,750 \frac{\text{lbf}}{\text{car}}$, empty: $F_{pb} = 10\,575 \frac{\text{lbf}}{\text{car}}$ | | | | | | | |
|  Delay brakes | | 10,817 | 13,123 | 18,111 | 9,277 | 11,583 | 16,571 |
|  Air brakes | | 9,743 | 10,859 | 13,188 | 8,200 | 9,309 | 11,602 |
| Difference | | 1,074 | 2,264 | 4,923 | 1,077 | 2,274 | 4,969 |

Difference exceeds FRA requirement of at most 1000ft undershoot!

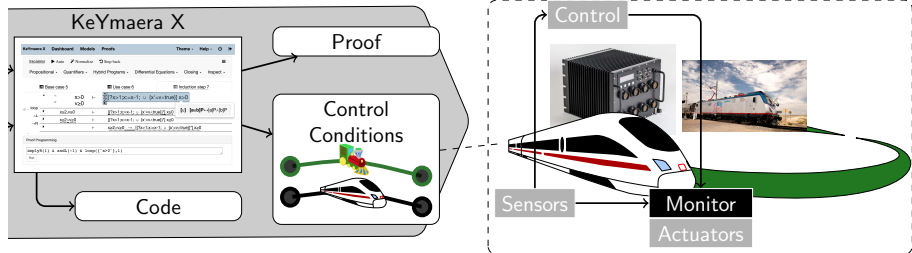
Theorem proving ensures correct model



No overshoot ✓ + Limited undershoot ✓

- ▶ **Proofs** for certification
- ▶ **Models** and **code** for system architecture and implementation
- ▶ **Control conditions** for runtime monitoring and testing

Transfer safety of model to controller implementation



Monitor desired effect + safe environment

- ▶ ModelPlex synthesizes and proves monitors for model compliance
- ▶ Runtime: ensure safety and detect anomalies
- ▶ Testing: generate and analyze test cases

Mitsch, S., Platzer, A.: ModelPlex: Verified runtime validation of verified cyber-physical system models. *Formal Methods in System Design*, 49(1), 2016.

KeYmaera X Dashboard Models Proofs Theme Help

Escalator Auto Normalize Step back

Propositional Quantifiers Hybrid Programs Differential Equations Closing Inspect

Base case 5 Use case 6 Induction step 7

| | | | |
|-----------------|----------------------------|----------|--|
| | -1: $x > 0$ | \vdash | $[[?x > 1; x := x - 1; \cup \{x' = v \wedge \text{true}\}]] x > 0$ |
| | -2: $v \geq 0$ | | |
| loop | $x \geq 2, v \geq 0$ | \vdash | $[[?x > 1; x := x - 1; \cup \{x' = v \wedge \text{true}\}]]^* x \geq 0$ |
| $\wedge L$ | $x \geq 2 \wedge v \geq 0$ | \vdash | $[[?x > 1; x := x - 1; \cup \{x' = v \wedge \text{true}\}]]^* x \geq 0$ |
| $\rightarrow R$ | | \vdash | $x \geq 2 \wedge v \geq 0 \rightarrow [[?x > 1; x := x - 1; \cup \{x' = v \wedge \text{true}\}]]^* x \geq 0$ |

Proof Programming

```

implyR(1) & andL(-1) & loop({'x>0'}, 1)
  
```

Run

[u] [a ub]P ↔ [a]P ∧ [b]P

www.keymaeraX.org

Stefan Mitsch

Computer Science Department, Carnegie Mellon University
 smitsch@cs.cmu.edu