CoasterX: A Case Study in Component-Driven Hybrid Systems Proof Automation

Brandon Bohrer
(joint work with Adriel Luo, Xuean Chuang, André Platzer)

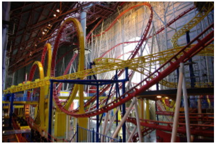
Logical Systems Lab
Computer Science Department
Carnegie Mellon University

ADHS, Jul 7 2018

Roller Coasters are Safety-Critical Systems

Top Thrill

Steel Phantom



Mindbender

[BLCP18]

Joker's Jinx

Phantom's Revenge

Fujin Raijin II

Rollback **Head Injury** Derailment

Formal Proofs in d$\mathcal{L}$ Ensure Safe Designs

Top Thrill

Steel Phantom

Mindbender

[BLCP18]

Rollback **Head Injury** Derailment

$$\Downarrow$$

$$\mathrm{Pre} \rightarrow [phys]\mathrm{Post}$$

Identify:

- Notion of safety *Post* $(acc < acc_{\mathrm{hi}})$

Formal Proofs in d$\mathcal{L}$ Ensure Safe Designs

Top Thrill

Rollback

Steel Phantom

Head Injury

Mindbender

Derailment

[BLCP18]

$$\Downarrow$$

$$\mathrm{Pre} \rightarrow [\mathit{phys}]\mathrm{Post}$$

Identify:

- Notion of safety *Post* ($acc < acc_{\mathrm{hi}}$)
- Safe conditions *Pre* ($v = v_0$)

Formal Proofs in d$\mathcal{L}$ Ensure Safe Designs

Top Thrill

Steel Phantom

Mindbender

[BLCP18]

Rollback **Head Injury** Derailment

$$\Downarrow$$

$$\mathrm{Pre} \rightarrow [\mathit{phys}]\mathrm{Post}$$

Identify:

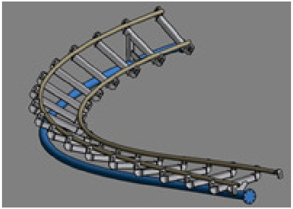
- Notion of safety *Post* ($acc < acc_{\mathrm{hi}}$)
- Safe conditions *Pre* ($v = v_0$)

Verify physical *plant* ($\{x' = \ldots, y' = \ldots\}$)

Design Verification Supplements
Simulation

Simulations typically used today [XXLY12, Wei15]

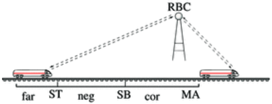
Approach	Pro	Con
Simulate	Rich dynamics, easy	Low rigor+precision
Verify	High rigor+precision	Simple dynamics, **hard**

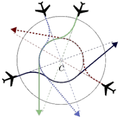


Verifying Physical Designs is a Challenge

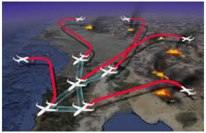
- How do we verify models at scale?
- How do we make verification accessible to non-experts?

Verifying Plant Designs is Important

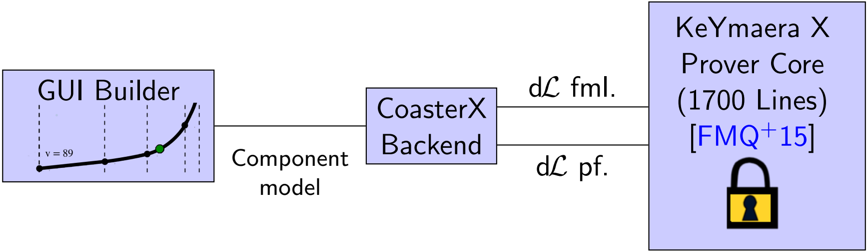




$$\Downarrow$$

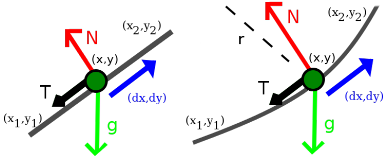


Component-Driven Proof Automation Enables Design Verification



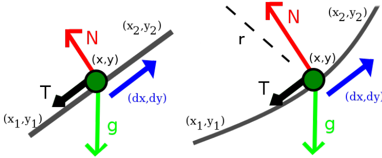
Goal	Solution
Accessible	High-level graphical modeling
Rigorous	Formal proof checked by small prover core
Scalable	Proof scales by exploiting component structure

Track Sections are Components for Coasters



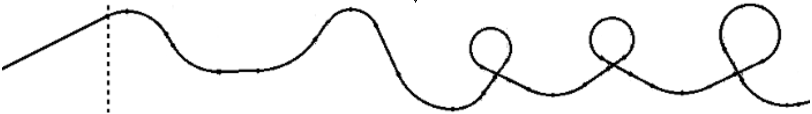
Generic Component

Track Sections are Components for Coasters



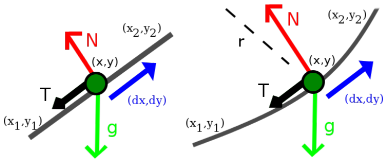
Generic Component

$\Downarrow$



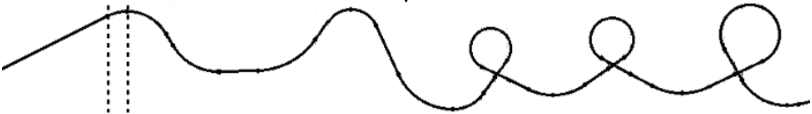
Automatic Composition

Track Sections are Components for Coasters



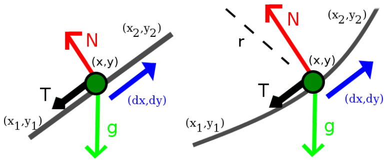
Generic Component

$\Downarrow$



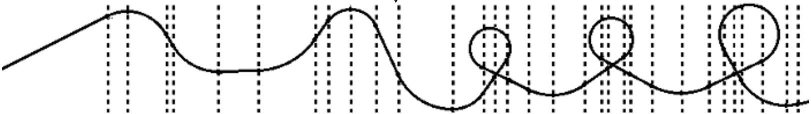
Automatic Composition

Track Sections are Components for Coasters



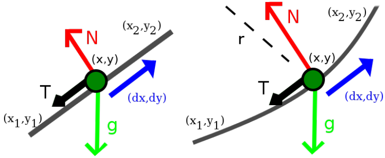
Generic Component

$\Downarrow$



Automatic Composition

Track Sections are Components for Coasters



Generic Component

$\Downarrow$

Automatic Composition

Background: d$\mathcal{L}$ Formulas

$$P, Q \ ::= \ P \wedge Q \mid \neg P \mid \forall x P \mid \theta_1 \geq \theta_2 \mid [\alpha]P$$

Example: $\mathrm{Pre} \rightarrow [\mathit{plant}]\mathrm{Post}$

Construct	Meaning
$P \wedge Q,\ \neg P, \forall x P$	First-order Logic
$\theta_1 \geq \theta_2$	Real arithmetic comparisons
$[\alpha]P$	Safety: After α runs, P always holds

Background: Hybrid Programs

$$\alpha, \beta \ ::= \ \cdots \mid \{x' = \theta \ \& \ P\} \mid \alpha \cup \beta \mid \alpha^*$$

Construct	Meaning
$\{x' = \theta \ \& \ P\}$	Evolve x at continuous rate θ
	Evolution domain constraint P asserted continuously
$\alpha \cup \beta$	Choose either α or β nondeterministically
α^*	Loop α nondeterministically $n \geq 0$ times

Velocity and Acceleration Bounds are Fundamental

Rollback

$0 < v_{lo} \leq v$

Head Injury

$|a| \leq a_{hi}$

Derailment

$|a| \leq a_{hi}$

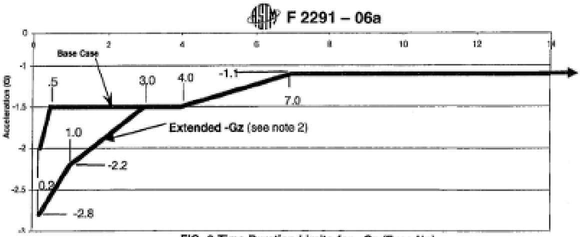
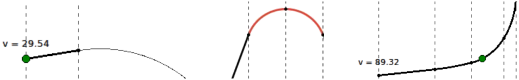


FIG. 9 Time Duration Limits for –Gz (Eyes Up)

[AST17]

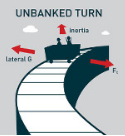
Tracks are 2D

- 2D modeling greatly simplifies GUI



- Vertical and horizontal bounds only (no lateral bound)
- Ignores *banking, wind, roll resistance (1-2%)*

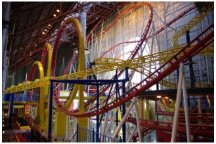
$\Rightarrow$



Conservative Bound Suffices for Phantom

Top Thrill

Steel Phantom



Mindbender

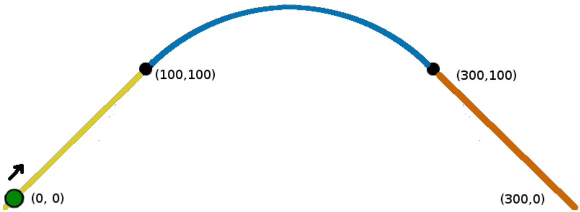
Joker's Jinx

Phantom's Revenge

Fujin Raijin II

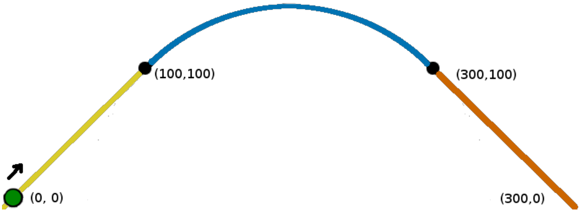
Rollback	Head Injury	Derailment
⚠ (>)	✓ (<)	✓ (<)

Example



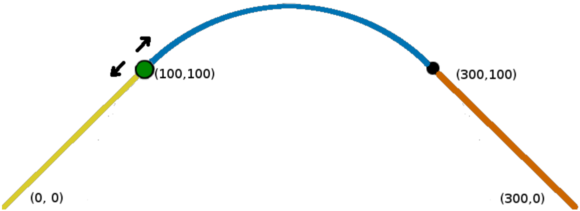
$$\begin{aligned}\text{plant} \equiv \{\{&\mathbf{x}' = \sqrt{\mathbf{2}}/\mathbf{2}\ \mathbf{v}, \mathbf{y}' = \sqrt{\mathbf{2}}/\mathbf{2}\ \mathbf{v}, \mathbf{v}' = -\sqrt{\mathbf{2}}/\mathbf{2}\ \mathbf{g}\ \&\ \mathbf{0} \le \mathbf{x} \le \mathbf{100}\} \\ \cup \{&x' = dx\ \ v, y' = dy, v' = -dy\ \ g, dx' = -dy\ \ v/100\sqrt{2}, \\ &dy' = dx\ \ v/100\sqrt{2}\ \&\ 100 \le x \le 200\} \\ \cup \{&x' = \sqrt{2}/2\ v, y' = -\sqrt{2}/2\ v, v' = \sqrt{2}/2\ g\ \&\ 200 \le x \le 300\}\}^*\end{aligned}$$

Example



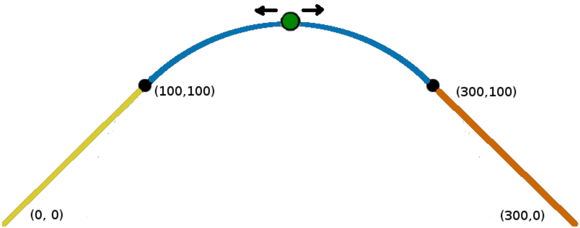
$$\begin{aligned}\text{plant} \equiv \{&\{\text{Line}(\ldots)\ \&\ \mathbf{0 \leq x \leq 100}\}\\ &\cup \{\text{Arc}(\ldots)\ \&\ 100 \leq x \leq 200\}\\ &\cup \{\text{Line}(\ldots)\ \&\ 200 \leq x \leq 300\}\}^*\end{aligned}$$

Example



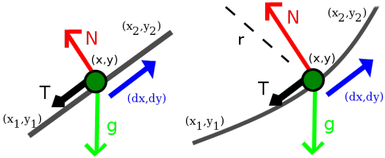
$$
\begin{aligned}
\text{plant} \equiv \{ & \{\text{Line}(\ldots)\ \&\ \mathbf{0 \le x \le 100}\} \\
& \cup \{\text{Arc}(\ldots)\ \&\ \mathbf{100 \le x \le 200}\} \\
& \cup \{\text{Line}(\ldots)\ \&\ 200 \le x \le 300\}\}^*
\end{aligned}
$$

Example



$$\begin{aligned}\text{plant} \equiv \{&\{\text{Line}(\ldots)\ \&\ 0 \le x \le 100\}\\ &\cup \{\text{Arc}(\ldots)\ \&\ \mathbf{100 \le x \le 200}\}\\ &\cup \{\text{Line}(\ldots)\ \&\ 200 \le x \le 300\}\}^*\end{aligned}$$

Individual Components are Modeled as ODEs



Arc Segment:

$$\text{Arc} \overset{\text{def}}{\equiv} \{x' = v \cdot dx, y' = v \cdot dy, v' = -dy \cdot g,$$
$$dx' = -dy \cdot v/r, dy' = dx \cdot v/r$$
$$\& \ InBounds(x_1, x_2, y_1, y_2)\}$$

Concrete Parameters are Plugged in From GUI

Line Segment:

$$\text{Line} \overset{\text{def}}{\equiv} \{x' = v \cdot dx, y' = v \cdot dy, v' = -dy \cdot g \;\&\; \mathit{InBounds}(x_1, x_2, y_1, y_2)\}$$

$$\Downarrow_{\text{Subst}}$$

$$\text{Line}(1, 0, \ldots) \overset{\text{def}}{\equiv} \{x' = v \cdot 1, y' = v \cdot 0, v' = -0 \cdot g \;\&\; \mathit{InBounds}(0, 100, 200, 200)\}$$

Composition is Modeled with Discrete Programs

Let track sections sec_i be component instances:

$$\mathsf{sec}_i \overset{\text{def}}{\equiv} \mathsf{Line}(\mathit{args}_i) \text{ or } \mathsf{Arc}(\mathit{args}_i)$$

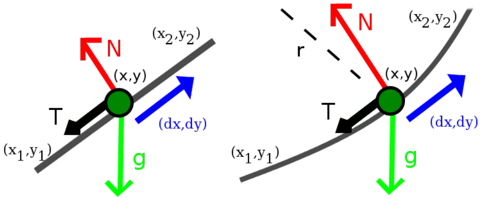
and system model α:

$$\mathsf{plant} \overset{\text{def}}{\equiv} (\mathsf{sec}_1 \cup \cdots \cup \mathsf{sec}_n)^*$$

Components Verified with Invariants and Solving

- Straight line is solvable, thus decidable.
- Arc needs invariant (energy conservation), proved manually:

$$E = E_0 \wedge \mathrm{OnTrack} \rightarrow [\mathsf{Arc}]\,(E = E_0 \wedge \mathrm{OnTrack})$$



Instantiation is Verified by Substitution

- Conceptually simple step
- Greatly improves performance (20x in some cases)

$$\text{Line} \stackrel{\text{def}}{\equiv} \{x' = v \cdot dx, y' = v \cdot dy, v' = -dy \cdot g \\ \& \ \mathit{InBounds}(x_1, x_2, y_1, y_2)\}$$

$$\Downarrow_{\text{Subst}}$$

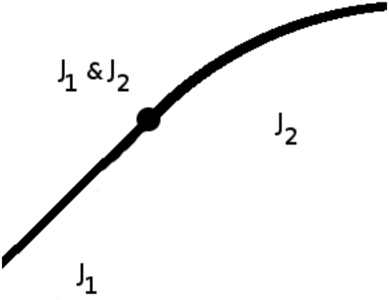
$$\text{Line}(1, 0, \ldots) \stackrel{\text{def}}{\equiv} \{x' = v \cdot 1, y' = v \cdot 0, v' = -0 \cdot g \\ \& \ \mathit{InBounds}(0, 100, 200, 200)\}$$

Composition is Verified by Contract-Checking

- At boundary, invariants for both sections hold
- Checked with arithmetic solving + custom automation

Example:

$$J_1 \equiv (x = y)$$
$$J_2 \equiv \left(y^2 + (x - 200)^2 = 100^2\right)$$



Analysis Distinguished 6 Safe/Unsafe Real Coasters

Top Thrill

Steel Phantom **(6.5g)**

Backyard

El Toro

Phantom's Revenge **(3.5g)**

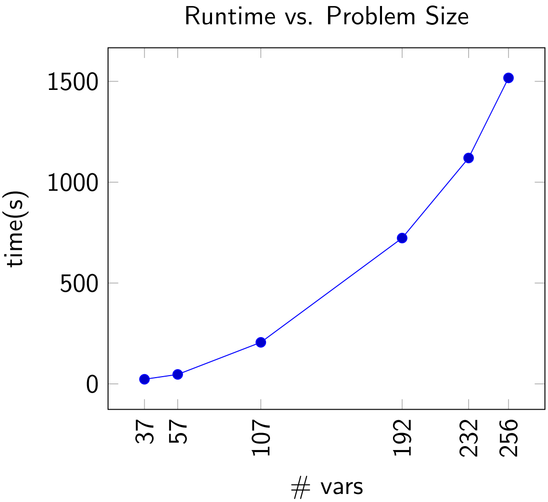
Lil' Phantom

This is the Largest d$\mathcal{L}$ Model Ever

Stats:

	CoasterX Max		Previous Max (Est.)
Components	56	$>$	3
Fml size	52KB	$>$	6.5KB
Proof Steps	20M (29K w/ reuse)	$>$	100K

Scalability is Quadratic



(on a recent workstation)

Component Verification Cost Sometimes Matters

Component	Time	# Steps
Line	**140s**	**900K**
Arc	$\approx$4.5s	$\approx$12.5K

Automatic proof (Line) vastly slower than manual proof (Arcs)

Future Work

Advanced Dynamical Models Answer Deeper Questions

Acceleration

$|a| \leq a_{hi}$

$\Rightarrow$

Rollback

$0 < v_{lo} \leq v$

Stuck

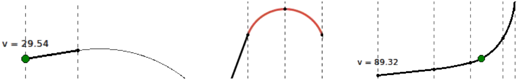
$0 < v_{lo} \leq v$

Friction

Wind

Advanced 3D Design

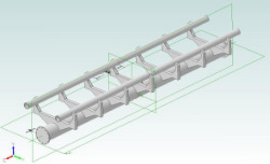
2D



Build Detect Simulate

⇓

3D



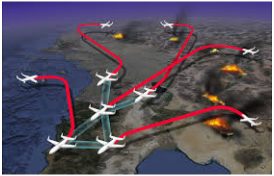
3D Modeling support enables lateral bounds and banking support

Rich Contracts Enable High-Impact Domains

- Transit networks: Contracts at intersections/switches
- Flight plans: Contracts at crossing points

Rail

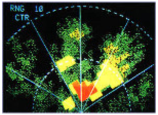
Road



UAV

Coasters Support Pedagogical Mission

- 15-424 CPS Foundations: Fun applications motivate students
- Course feeds into undergraduate research
- Initial stages were Adriel + Xuean's 15-424 course project



GPWS

Chute

Pong

Coaster

Chess

Baseball

Questions?

Top Thrill

Steel Phantom

Backyard

El Toro

Phantom's Revenge

Lil' Phantom

References I

- ASTM, *Standard Practice for Design of Amusement Rides and Devices*, Standard, ASTM Intl., Sep 2017.
- Brandon Bohrer, Adriel Luo, Xuean Chuang, and André Platzer, *CoasterX: A case study in component-driven hybrid systems proof automation*, IFAC, 2018.
- Nathan Fulton, Stefan Mitsch, Jan-David Quesel, Marcus Völp, and André Platzer, *KeYmaera X: An axiomatic tactical theorem prover for hybrid systems*, CADE (Berlin) (Amy Felty and Aart Middeldorp, eds.), LNCS, vol. 9195, Springer, 2015, pp. 527–538.
- Nick Weisenberger, *Coasters 101: An engineer's guide to roller coaster design*, 2015.

References II

Gening Xu, Hujun Xin, Fengyi Lu, and Mingliang Yang, *Kinematics and dynamics simulation research for roller coaster multi-body system*, Advanced Materials Research, vol. 421, Trans Tech Publications, 2012, pp. 276–280.