Rigorous Simulation

Walid Taha

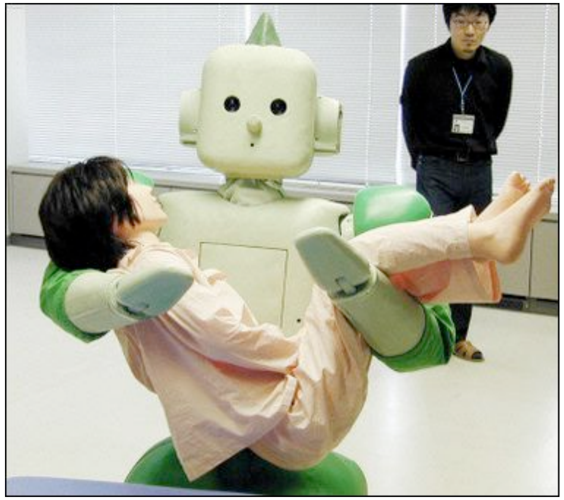
Halmstad University and Rice University

Rigorous Simulation

Aaron Ames, **Kevin Atkinson**, Jerker Bengtsson, Raktim Bhattacharya, Paul Brauner, Robert Cartwright, Alexandre Chapoutot, **Adam Duracz**, **Jan Duracz**, Henrik Eriksson, Veronica Gaspes, Christian Grante, Jun Inoue, **Michal Konecny**, Marcie O'Malley, Travis Martin,Jawad Masood, Marisa Peralta, Cherif Salama, **Walid Taha**, Edwin Westbrook, Fei Xu, **Yingfu Zeng**, Yun Zhu

Halmstad University, Rice University and Texas A&M, SP, Volvo Trucks

Robot Design



NG-Test Project

Simulation in innovation

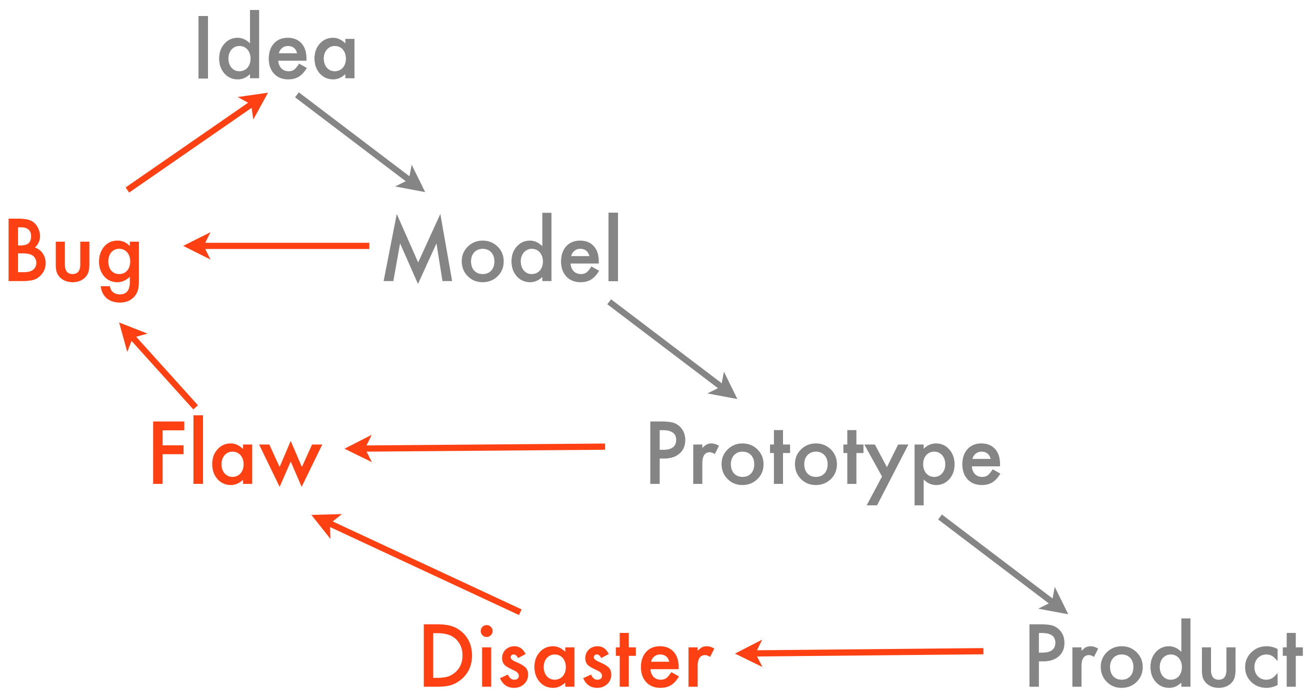
Idea

Model

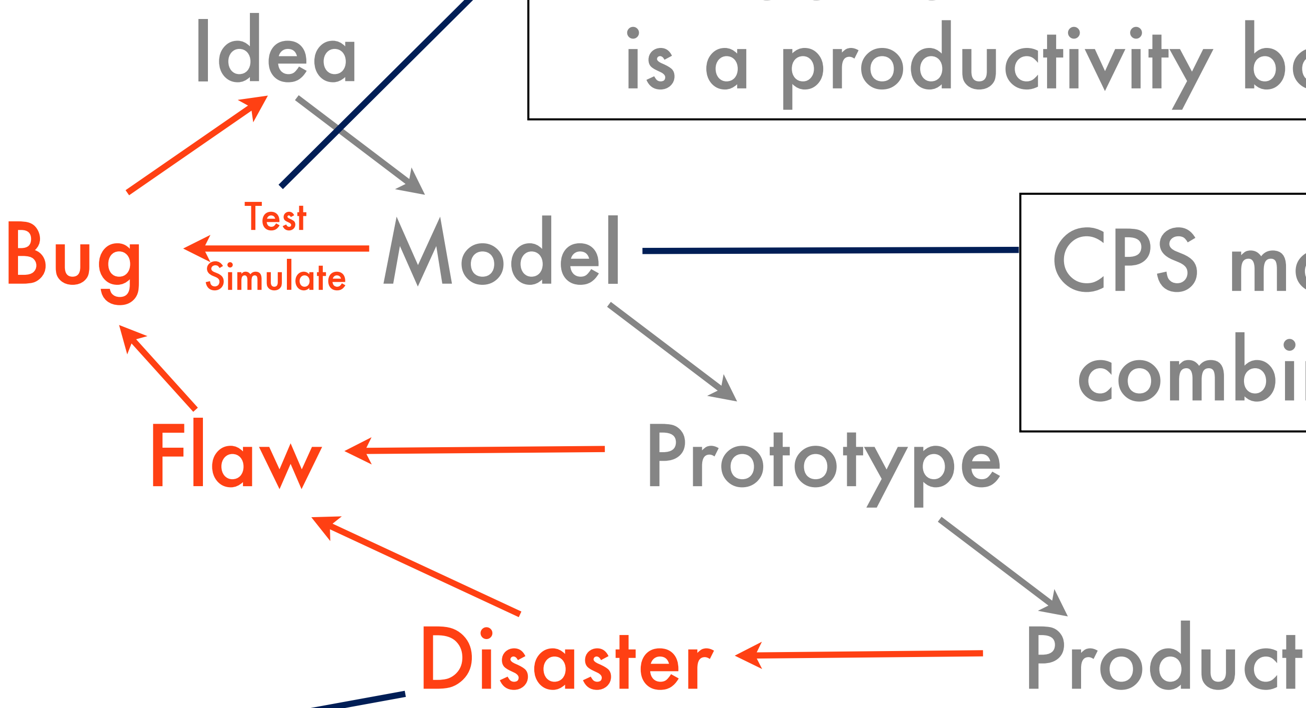
Prototype

Product

Simulation in innovation



Rigorous Simulation



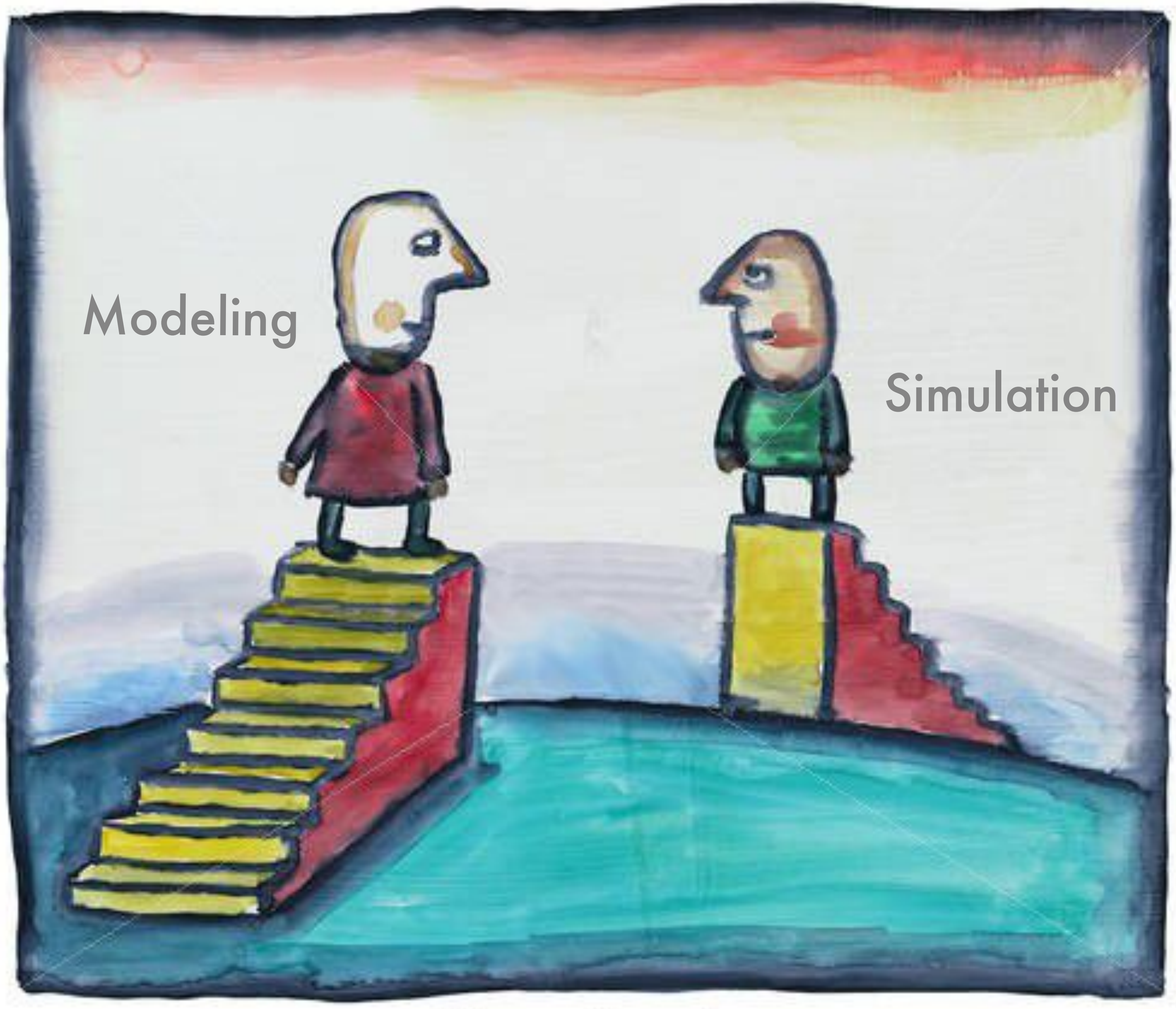
Debugging models (simulation) is a productivity bottleneck

CPS models must combine C & P!

Late "debugging" can be extremely expensive

This talk

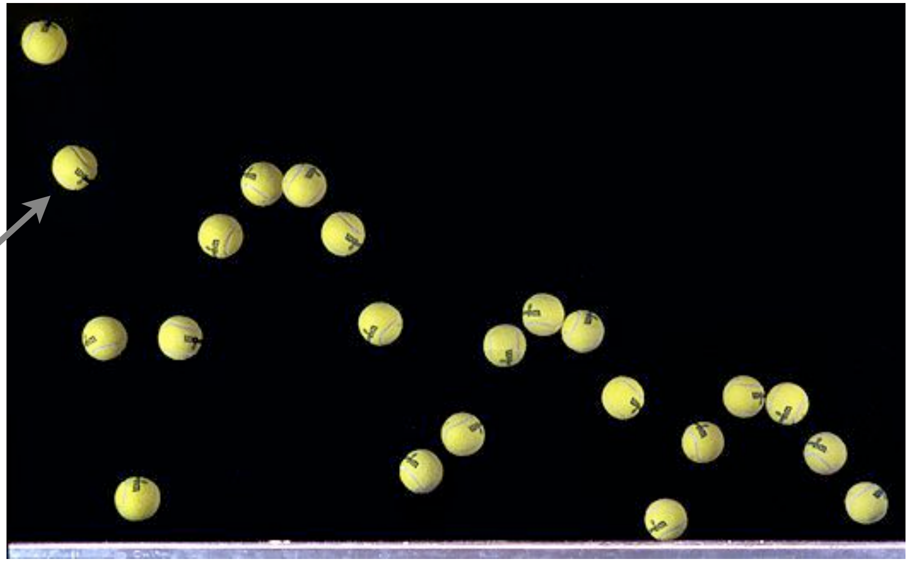
- The CPS simulation domain
- Available tools
- Rigorous simulation
- The staging connection
  - The E-L equation (previous work)
  - Binding time analysis (ongoing)


Modeling
Simulation

Why simulation is hard

- Solving continuous equations:
  - ODEs, DAEs, PDEs, IDEs, ...
  - IVP vs. BVP
- Hybrid aspects pervasive (both C & Ph)
- Dealing with precision *in models*
  - Uncertainty (intervals)

Hybrid systems basics



Derivatives

Interaction

Discontinuities

Computability

Specific problems

- Equality & zero crossing (if x=0 then ...)
- Zeno-behavior (bouncing ball)
- Static verification (e.g. solvable, stable)
- Numerical precision and validity
  - Pole detection (e.g. $x'=x^*x$, $x(0)=1$(?))
- Semantic treatment *desperately* needed

Current tools

- Simulink
- Mathematica, Maple
- ML, Haskell
- HA, e.g. CHARON
- Modelica, Verilog-AMS, ...

Strengths

- Simulink: **Lots of models (IP)**
- Mathematica, Maple: **Symbolics**
- ML, Haskell: **Expressivity, concurrency**
- HA, e.g. CHARON: **Formal proof**
- Modelica, Verilog-AMS: **Equations**

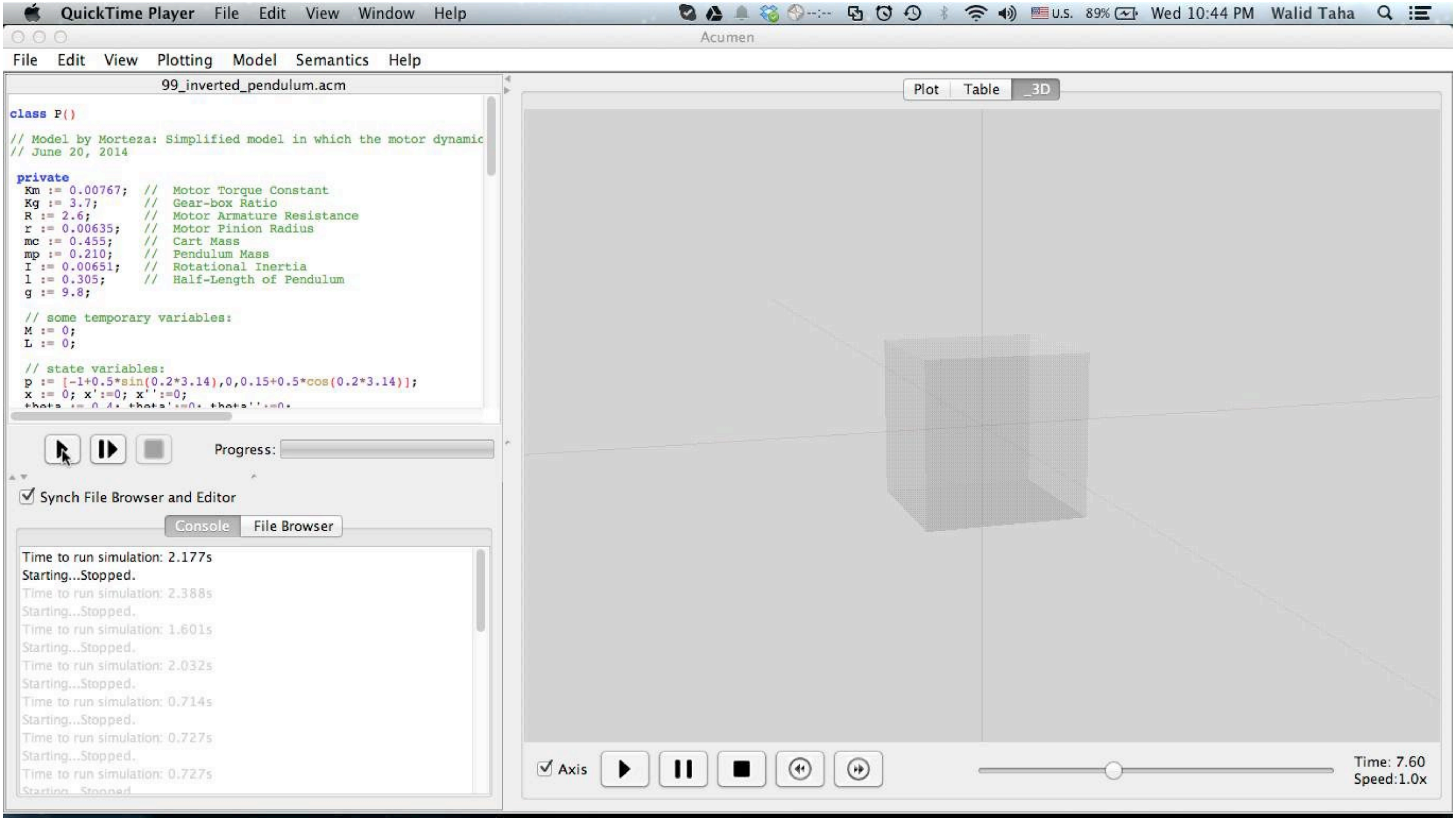
Weaknesses

- Simulink: **Numeric semantics...**
- Mathematica: **Executability**
- ML, Haskell: **"Indeterminism"**
- HA, e.g. CHARON: **Scalability**
- Modelica, Verilog-AMS: **Semantics...**

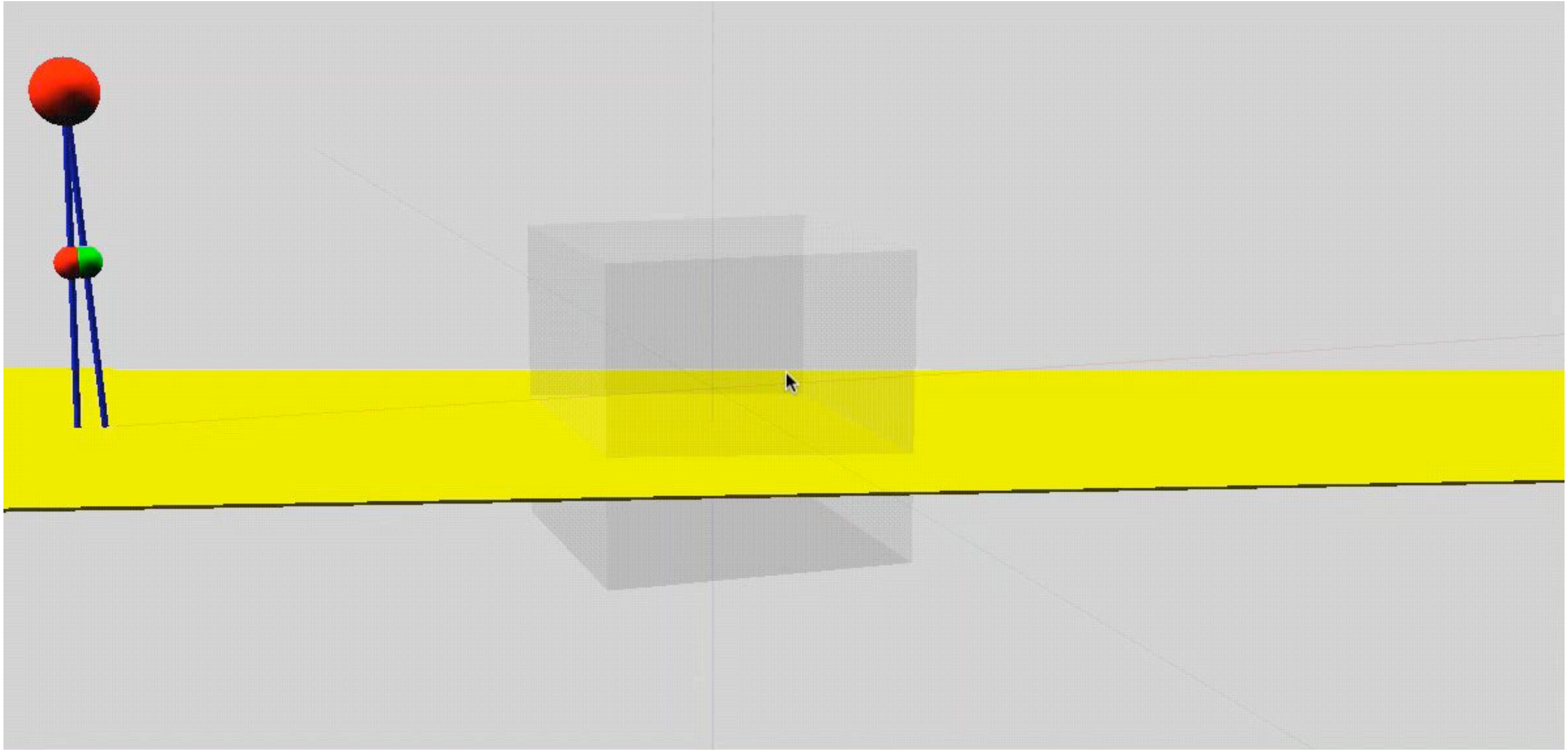
Acumen

- Acumen ’09: continuous language
  - “Math as a programming language”
- Acumen ’10 (- now): hybrid language
  - Hierarchical hybrid systems
  - IDE (automatic plotting & 3D view)
  - Rigorous semantics

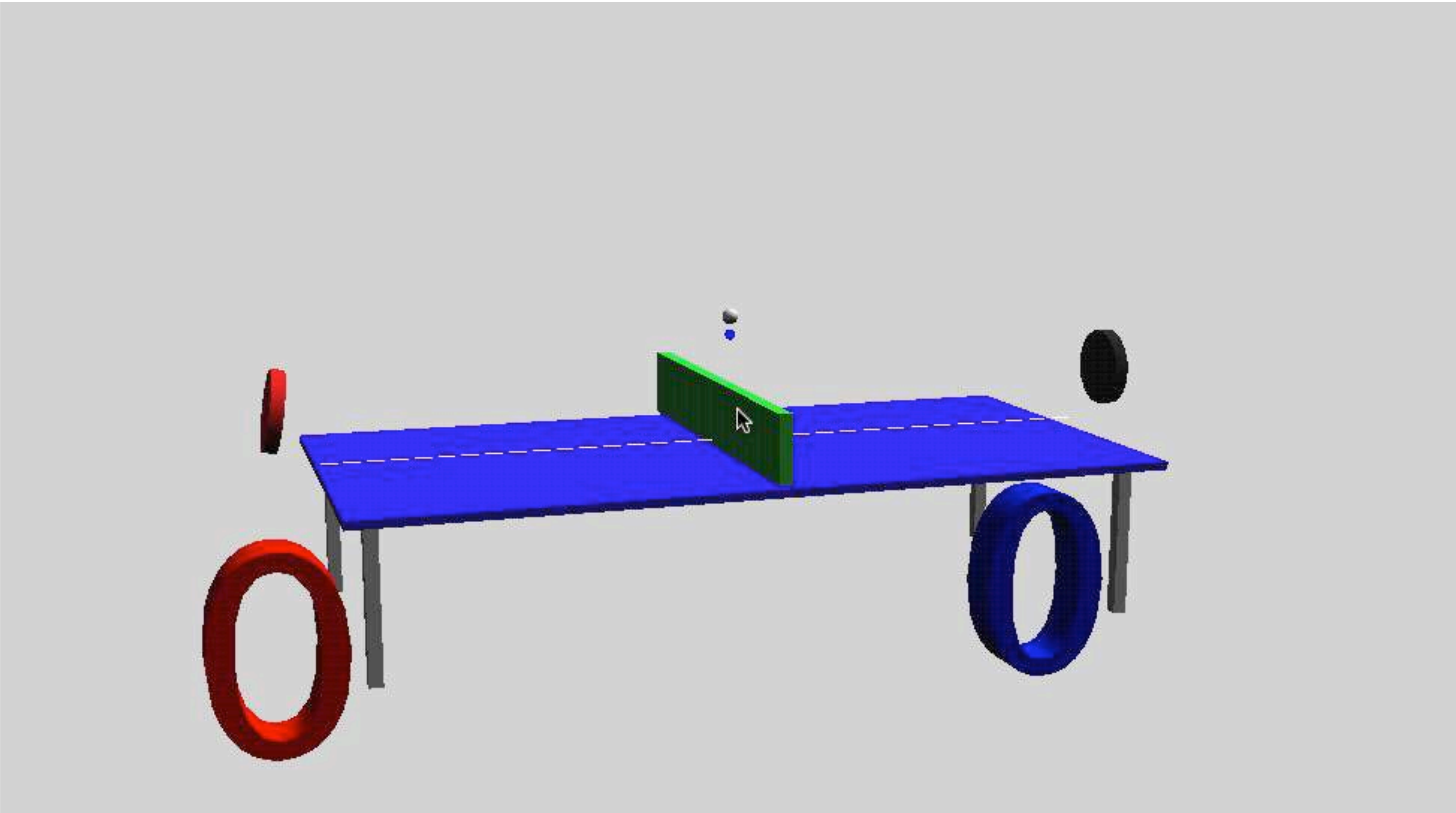
Basic Look and Feel



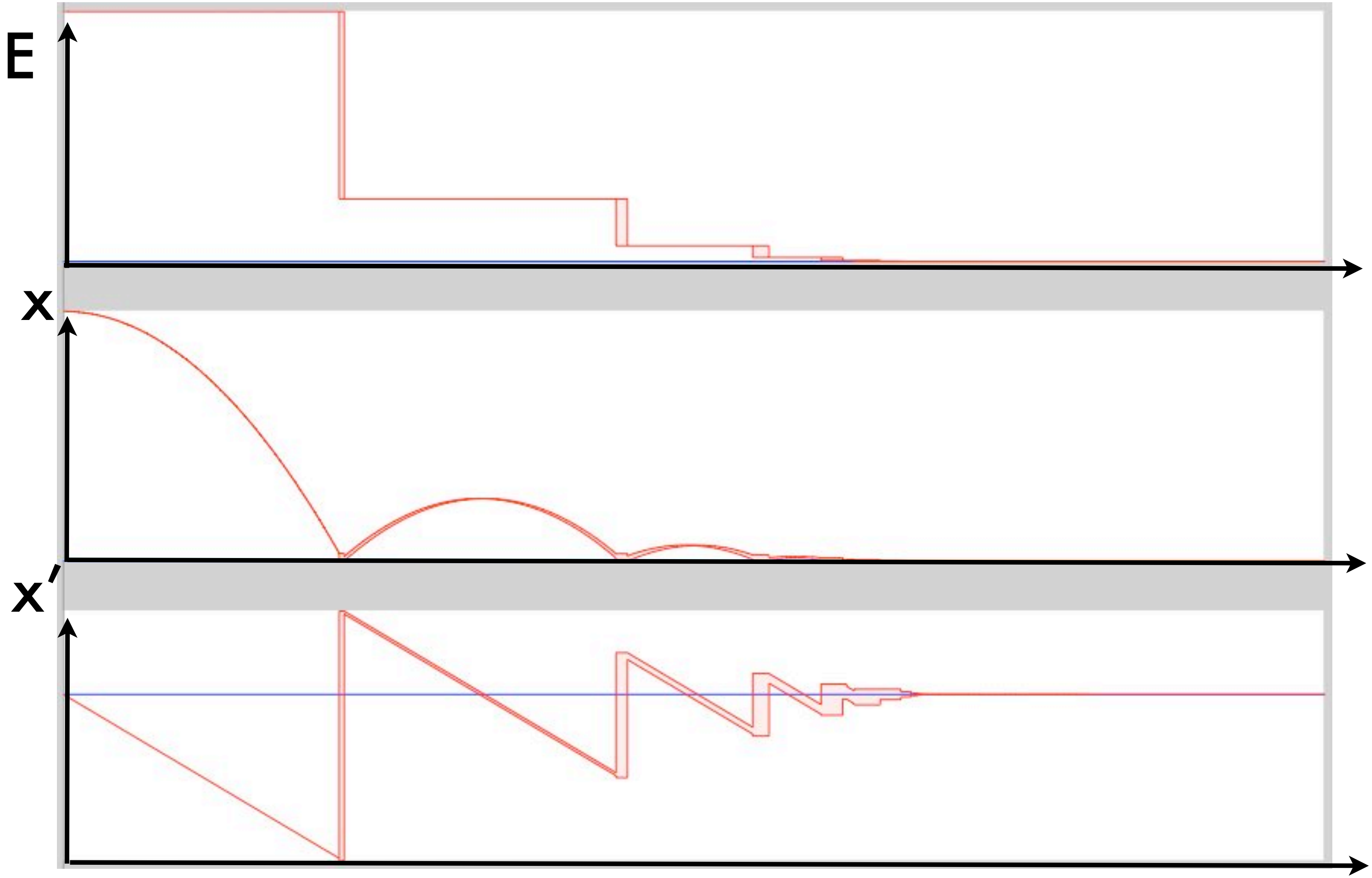
Hybrid Dynamics Example



Example for CPS course



Rigorous Semantics



Rigorous Semantics

CPSNA 2013
The 1st IEEE International Conference on Cyber-Physical Systems, Networks, and Applications

Taipei, Taiwan **August 19 - 20, 2013**

Enclosing the Behavior of a Hybrid System up to and Beyond a Zeno Point

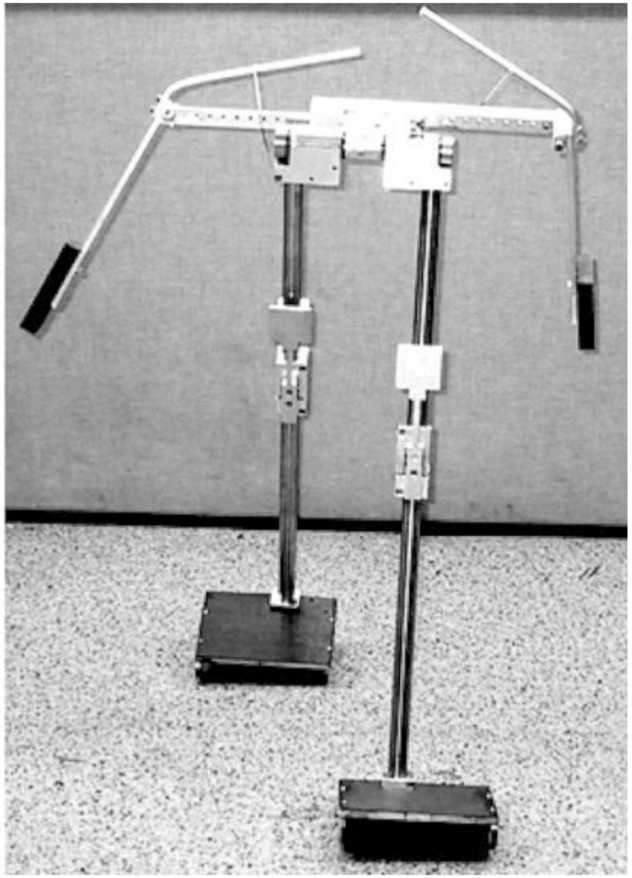
Michal Konečný*, Walid Taha†‡, Jan Duracz†, Adam Duracz† and Aar
*School of Engineering and Applied Science, Aston University, Birmingham, UK, Email:
†Halmstad University, Halmstad, Sweden, Email: Walid.Taha@hh.se, Jan.Duracz@hh.s
‡Computer Science Department, Rice University, Texas, USA
§Department of Electrical & Computer Engineering, Texas A&M University, USA

Abstract—**Even simple hybrid systems like the classic bouncing ball can exhibit Zeno behaviors. The existence of this type of behavior has so far forced simulators to either ignore some events or risk looping indefinitely. This in turn forces modelers to either insert ad hoc restrictions to circumvent Zeno behavior or to abandon hybrid modeling. To address this problem, we take a fresh look at event detection and localization. A key insight that emerges from this investigation is that an *enclosure* for a given time interval can be valid independently of the occurrence of a given event. Such an event can then even occur an unbounded number of times, thus making it possible to handle certain types of Zeno behavior.**

systems over a decad
interesting patholo
lar behavior turn
between tradition
its existence can
tions and/or enteri
consideration in de
systems simulator i

This phenomenon
systems. It can oc
with impact constr

The Staging Connection



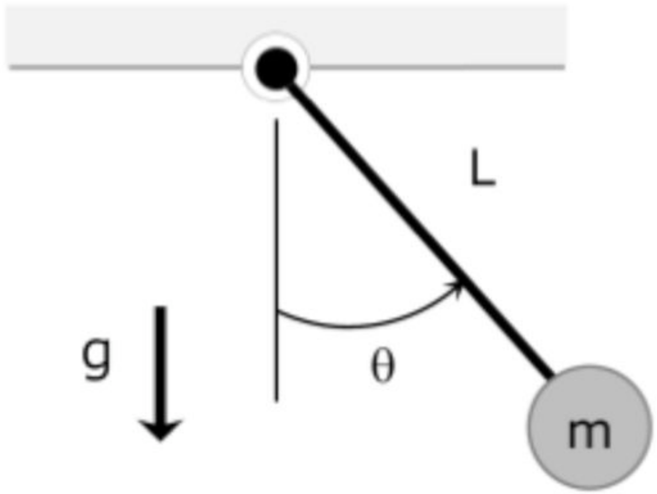
A passive robot walks naturally down *inclines*

Can this be generalized?

The Staging Connection

- Effort by one of our three *domain experts*
- Robot model uses 8x8 Lagrangian
  - Must convert to “executable” math
  - Mathematica gave a 13MB derivative!
- It’s very hard for robotics experts to find the right tools for simulation

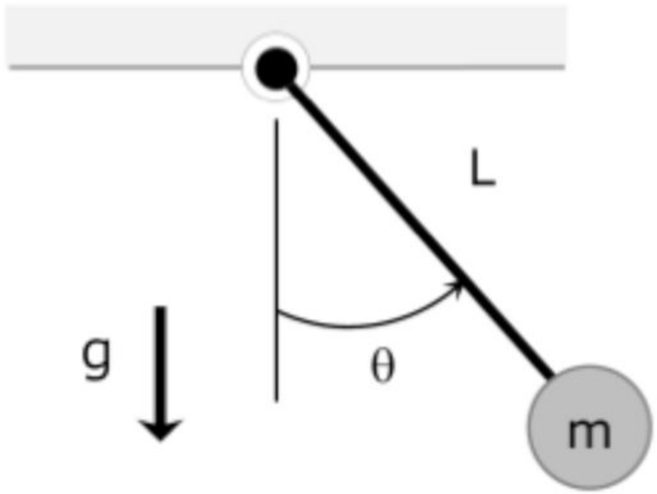
Equational models



$$m = 5 \quad g = 9.81 \quad \ell = 3 \quad I = m\ell^2$$
$$F\ell\cos(\theta) - mg\ell\sin(\theta) = I\ddot{\theta}$$

(a) Newtonian Formulation of a Pendulum

Equational models

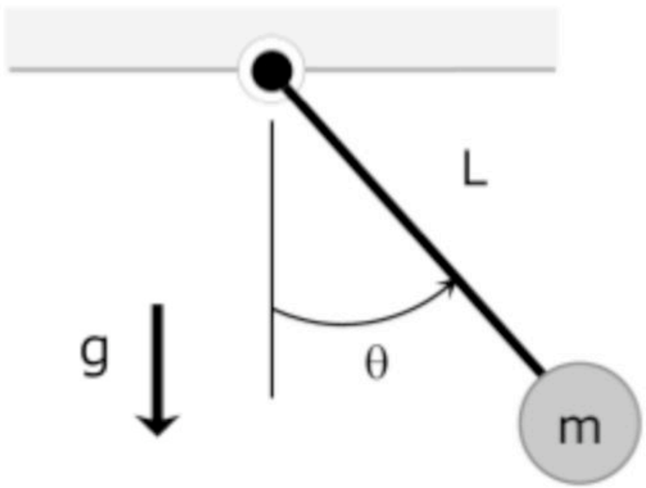


$$m = 5 \quad g = 9.81 \quad \ell = 3 \quad I = m\ell^2$$

$$F\ell\cos\boxed{(\theta)} - mg\ell\sin(\theta) = I\boxed{\ddot{\theta}}$$

(a) Newtonian Formulation of a Pendulum

Equational models



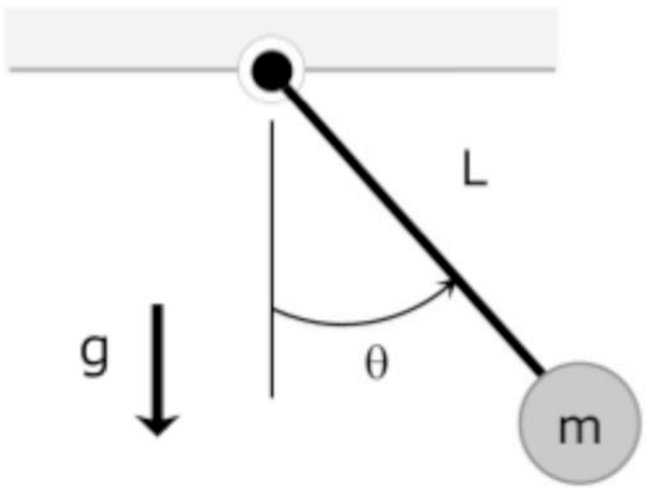
$$m = 5 \quad g = 9.81 \quad \ell = 3 \quad I = m\ell^2$$

$$T = \frac{1}{2} I \dot{\theta}^2 \quad V = mg\ell\,(1 - \cos(\theta))$$

$$L = T - V \qquad \frac{d}{dt}\left(\frac{\partial L}{\partial \dot{\theta}}\right) - \frac{\partial L}{\partial \theta} = 0$$

(b) Lagrangian Formulation of a Pendulum

Equational models



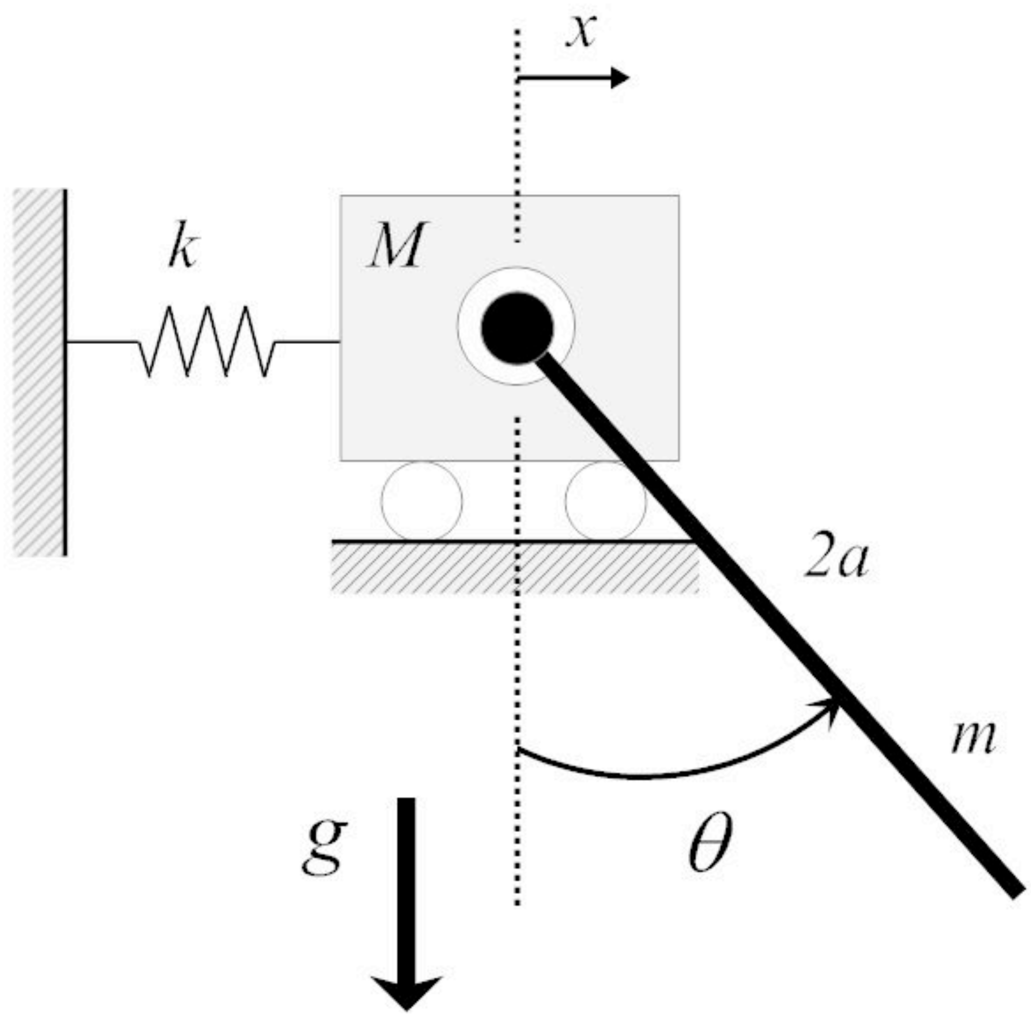
$$m = 5 \quad g = 9.81 \quad \ell = 3 \quad I = m\ell^2$$

$$T = \frac{1}{2} I \dot{\theta}^2 \quad V = mg\ell\,(1 - \cos(\theta))$$

$$L = T - V \qquad \frac{d}{dt}\left(\boxed{\frac{\partial L}{\partial \dot{\theta}}}\right) - \boxed{\frac{\partial L}{\partial \theta}} = 0$$

(b) Lagrangian Formulation of a Pendulum

Equational models



Equational models

$$q = [x, \theta] \quad a = 1 \quad m = 2 \quad M = 5$$

$$g = \frac{49}{5} \quad k = 2 \quad I = \frac{4}{3} m a^2$$

$$T = \frac{1}{2}(M + m)\dot{x}^2 + m a \dot{x}\dot{\theta}\cos(\theta) + \frac{2}{3} m a^2 \dot{\theta}^2$$

$$V = \frac{1}{2} k x^2 + m g a (1 - \cos(\theta)) \quad L = T - V$$

$$\forall i \in \dim(q) \quad \frac{d}{dt}\left(\frac{\partial L}{\partial \dot{q}_i}\right) - \frac{\partial L}{\partial q_i} = 0$$

(d) Pendulum/Mass

Equational models

$$\boxed{q} = \boxed{[x, \theta]} \quad a = 1 \quad m = 2 \quad M = 5$$

$$g = \frac{49}{5} \quad k = 2 \quad I = \frac{4}{3} m a^2$$

$$T = \frac{1}{2}(M + m)\dot{x}^2 + ma\dot{x}\dot{\theta}\cos(\theta) + \frac{2}{3} m a^2 \dot{\theta}^2$$

$$V = \frac{1}{2} k x^2 + mga(1 - \cos(\theta)) \quad L = T - V$$

$$\boxed{\forall i \in \dim(q)} \quad \frac{d}{dt}\left(\frac{\partial L}{\partial \boxed{\dot{q}_i}}\right) - \frac{\partial L}{\partial \boxed{q_i}} = 0$$

(d) Pendulum/Mass

From Acumen to DAEs

$$q = [x, \theta] \quad a = 1 \quad m = 2 \quad M = 5 \quad g = 9.8 \quad k = 2$$

$$I = \frac{4}{3}ma^2 \quad T = \frac{1}{2}(M+m)\dot{x}^2 + ma\dot{x}\dot{\theta}\cos(\theta) + \frac{2}{3}ma^2\dot{\theta}^2$$

$$V = \frac{1}{2}kx^2 + mga(1-\cos(\theta)) \quad L = T - V$$

$$\forall i \in \dim(q) \quad \frac{d}{dt}\left(\frac{\partial L}{\partial \dot{q}_i}\right) - \frac{\partial L}{\partial q_i} = 0$$

(a) Acumen Source for Pendulum/Mass Example

Defined: $q := [x, \theta] \quad a := 1 \;\ldots\; I := \frac{4}{3}ma^2 \;\ldots$

Computed: $\forall i \in \dim(q) \quad \frac{d}{dt}\left(\frac{\partial L}{\partial \dot{q}_i}\right) - \frac{\partial L}{\partial q_i} = 0$

(b) After Defined Variable Analysis

From Acumen to DAEs

Defined: $q := [x, \theta] \quad a := 1 \ \ldots \quad I := \frac{4}{3} m a^2 \ \ldots$

Computed: $\forall i \in \dim(q) \quad \frac{d}{dt}\left(\frac{\partial L}{\partial \dot{q}_i}\right) - \frac{\partial L}{\partial q_i} = 0$

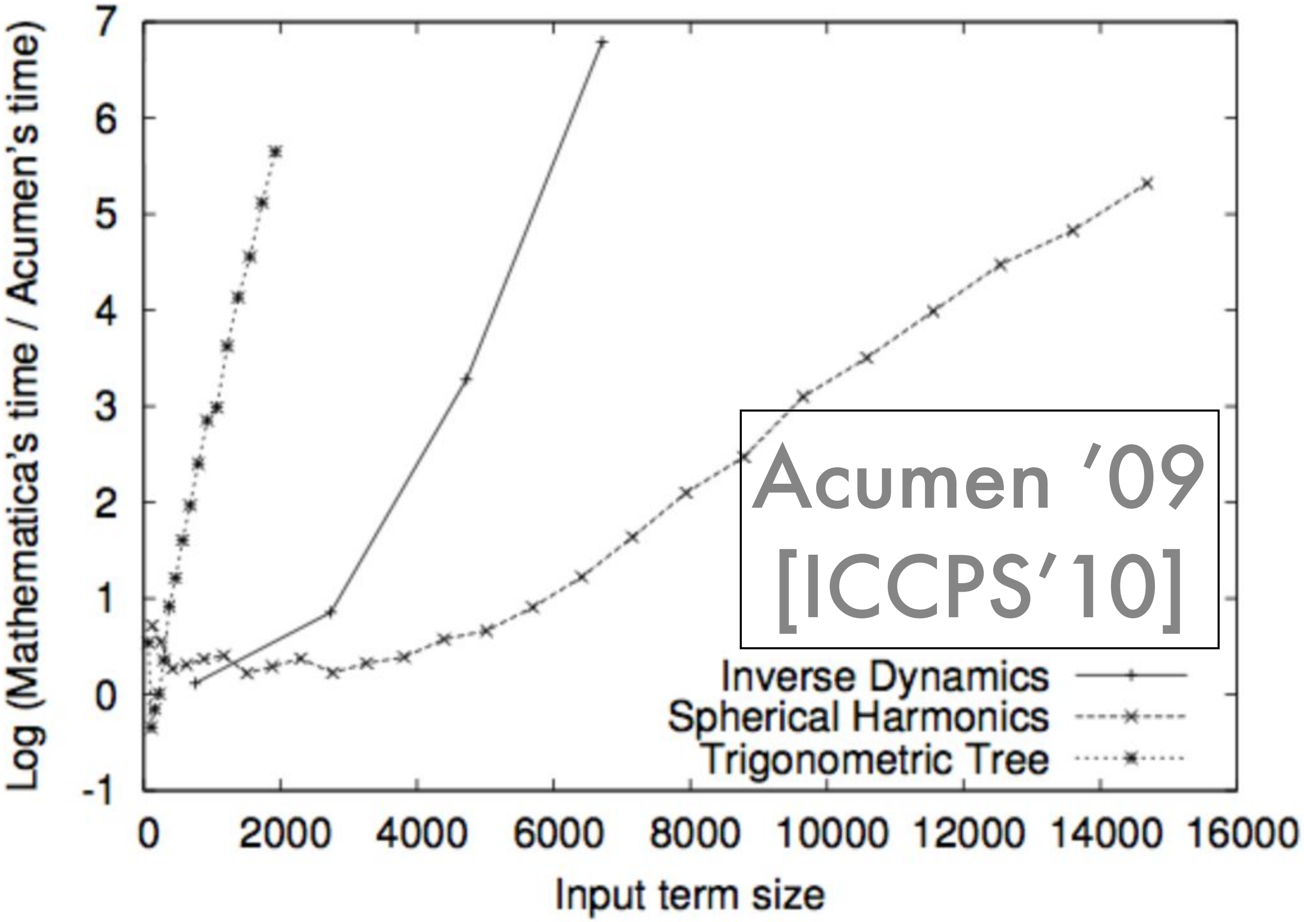
(b) After Defined Variable Analysis

Defined: $q := \boxed{[\boxed{x}, \boxed{\theta}]} \quad a := \boxed{1} \ \ldots \quad I := \boxed{\frac{4}{3} m a^2} \ \ldots$

Computed: $\boxed{\forall i \in \dim(q) \quad \boxed{\frac{d}{dt}\left(\frac{\partial L}{\partial \boxed{\dot{q}_i}}\right) - \frac{\partial L}{\partial \boxed{q_i}} = 0}}$

(c) After Binding-Time Analysis (BTA)

Effect on Performance



(f) Acumen is exponentially faster than Mathematica.

BTA: Syntax

$$
\begin{array}{lcl}
e \in \textsf{Exp} & ::= & c, x, \langle e_i \rangle, f(\langle e_i \rangle) \\
s \in \textsf{Cmd} & ::= & x = e;\ x' = e;\ \textsf{if}(e_0)\ s_1\ ;\ s_2 \\
c \in \textsf{Constraints} & ::= & \mathit{Stat};\ \mathit{Dyn};\ l_1 \sqsubseteq l_2
\end{array}
$$

$$
\rho : \textsf{Exp} \rightarrow l\ ;\ [\![\cdot]\!]_{\rho,l} : \langle c_i \rangle
$$

BTA: Collecting constraints

$$
\begin{aligned}
\llbracket c \rrbracket_{\rho,l} &= l \sqsubseteq \mathit{Stat} \\
\llbracket x \rrbracket_{\rho,l} &= \rho(x) \sqsubseteq l \\
\llbracket f(\langle e_i \rangle^{l_i}) \rrbracket_{\rho,l} &= \{\cup_i \llbracket e_i \rrbracket_{\rho,l_i}\} \cup \{\cup_i l_i \sqsubseteq l\} \\
\llbracket \langle e_i \rangle^{l_i} \rrbracket_{\rho,l} &= \{l \sqsubseteq \mathit{Stat}\} \cup \{\cup_i \llbracket e_i \rrbracket_{\rho,l_i}\} \\
\llbracket x = e^{l_1} \rrbracket_{\rho,l} &= l \sqsubseteq \rho(x) \cup l_1 \sqsubseteq l \cup \llbracket e \rrbracket_{\rho,l_1} \\
\llbracket x' = e^{l_1} \rrbracket_{\rho,l} &= l \sqsubseteq \rho(x') \cup \mathit{Dyn} \sqsubseteq \rho(x) \cup l_1 \sqsubseteq l \cup \llbracket e \rrbracket_{\rho,l_1} \\
\llbracket e_1^{l_1} = e_2^{l_2} \rrbracket_{\rho,l} &= l_1 \sqsubseteq l \cup l_2 \sqsubseteq l \cup \llbracket e_1 \rrbracket_{\rho,l_1} \cup \llbracket e_2 \rrbracket_{\rho,l_2} \\
\llbracket \text{if } e^{l_e} \; s_1^{l_1} \; s_2^{l_2} \rrbracket_{\rho,l} &= l_e \sqsubseteq l \cup l_1 \sqsubseteq l \cup l_2 \sqsubseteq l \cup \llbracket e \rrbracket_{\rho,l_e} \cup \llbracket s_1 \rrbracket_{\rho,l_1} \cup \llbracket s_2 \rrbracket_{\rho,l_2}
\end{aligned}
$$

BTA: Solving constraints

$$
\begin{aligned}
C \cup Stat^k \cup x \sqsubseteq k &\rightarrow C \cup Stat^k \cup Stat^x \\
C \cup Dyn^x \cup x \sqsubseteq k &\rightarrow C \cup Dyn^x \cup Dyn^k \\
C \cup Stat^x \cup Dyn^x &\rightarrow contradiction
\end{aligned}
$$

Example: Pendulum


```
// Source Model

model Main(simulator)=
 initially
   t=pi/4, t'=0, t''= 0,
   g= 0,  L=0, T = 0, V = 0
 always
 g = 9.8,
 T = (1/2) * t'^2,
 V = -g*cos(t),
 L = T - V,
 L'[t']' - L'[t]=0
```

// Labeled model

model Main(simulator)=

 initially

   t^{0} =pi/4,t'^{1} =0,t''^{2} =0,

   g^{3}=0, L^{4}=0, T^{5}=0,V^{6}=0

 always

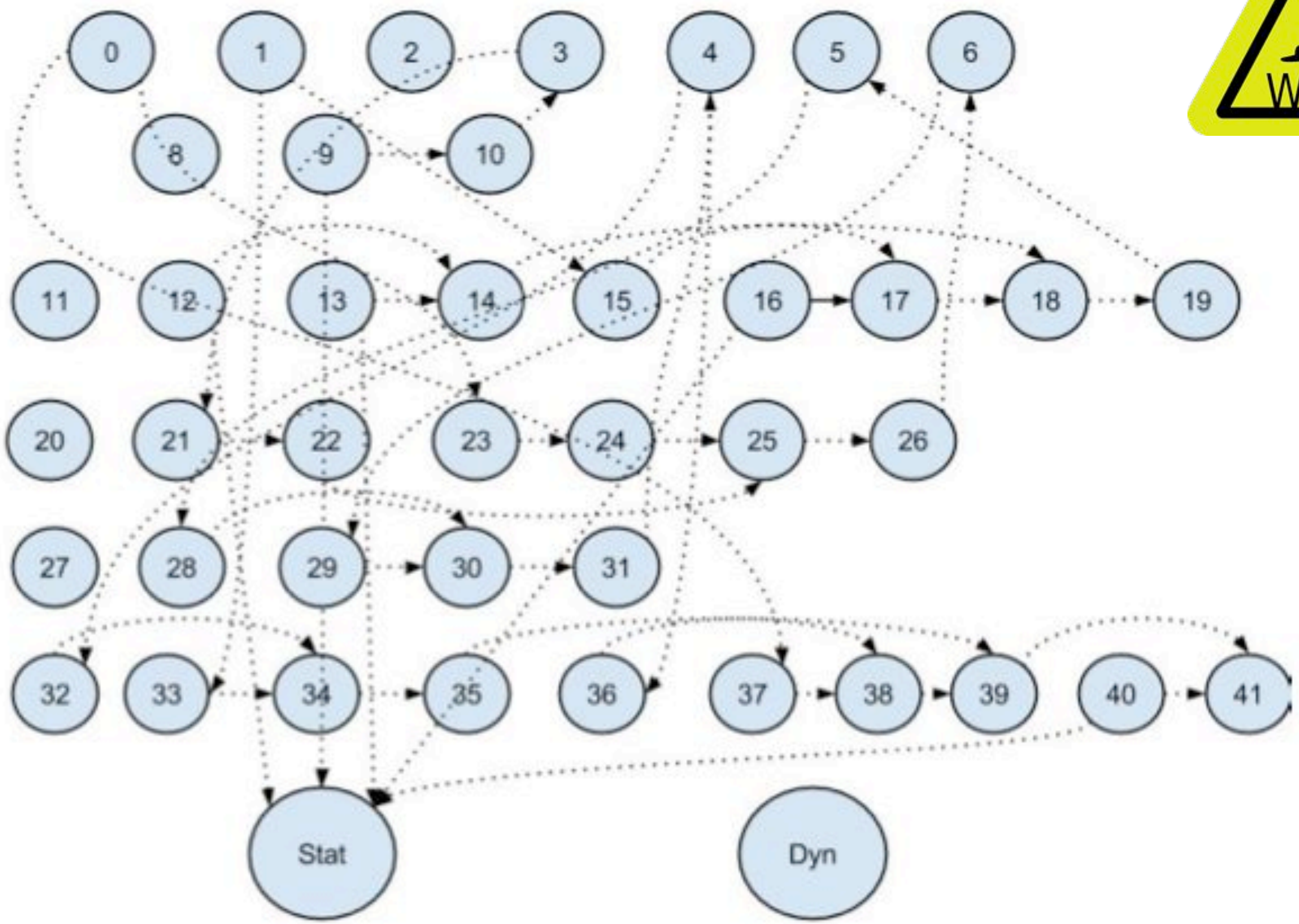
 $(g^{8}$ = $9.8^{9})^{10}$,

 $(T^{11}$ = $((1^{12}$ / $2^{13})^{14}$*$(t'^{15}$^$2^{16})^{17})^{18})^{19}$,

 $(V^{20}$ = $((-g^{21})^{22}$*$\cos(t^{23})^{24})^{25})^{26}$,

 $((L^{32}{}'[t'^{33}]^{34}{}'^{35}$ - $L^{36}{}'[t^{37}]^{38})^{39}$ = $0^{40})^{41}$

Example: Pendulum



Example: Pendulum


```
// Specializtion

model Main(simulator)=
 initially
   t=pi/4, t'=0, t''= 0,
   g= 0,  L=0, T = 0, V = 0
 always
 g = 9.8,
 T = (1/2) * t'^2,
 V = -g*cos(t),
 L = 0.5 * t' ^ 2 - (-9.8 * cos(t)),
 0.5 * 2 * t'' + 9.8 * sin(t)= 0
```

```
// Gaussian Elimination

model Main(simulator)=
 initially
   t=pi/4, t'=0, t''= 0,
   g= 0,  L=0, T = 0, V = 0
 always
 g = 9.8,
 T = (1/2) * t'^2,
 V = -g*cos(t),
 L = 0.5 * t' ^ 2 - (-9.8 * cos(t)),
 t'' = - 9.8 * sin(t)
```

BTA: Next Steps

- Formalizing correctness criteria
  - "Well-annotated programs don't go wrong"
- Establishing sufficiency for an interesting class of models
  - Example: Rigid body dynamics

Conclusions (1/2)

- Rigorous simulation is a powerful tool
- Being based on simulation makes intuitive
- Being rigorous makes it a verification tool
  - Semantics makes the tool rigorous
  - Staging implements the semantics efficiently

Conclusions (2/2)

- Rigorous simulation naturally accommodates parametric uncertainty
  - Modeling uncertainty makes simulations much more informative
- Using rigorous simulation during early-stage design has a distinctive flavor that promotes robust design

Thank you!

- To download our papers:
  - http://effective-modeling.org
- To download Acumen
  - http://acumen-language.org
- For CPS Lecture Notes
  - http://bit.ly/LNCPS-2014