Modeling and Simulating Cyber-Physical Systems using CyPhySim

Invited Talk

Special Session on Design of Hybrid Systems

EMSOFT 2015

Amsterdam

Edward A. Lee

Oct. 6, 2015

What is CyPhySim?

CyPhySim is an open-source simulator for CPS based on Ptolemy II with:

- Mixed continuous and discrete dynamics
- Superdense time
- Modal models and hybrid systems
- Smooth tokens
- Five simulation engines:
  1. Discrete event simulation (DE)
  2. Quantized-state solvers (QSS)
  3. Runge-Kutta solvers (RK2-3, RK4-5)
  4. Algebraic loop solvers (Substitution, Newton, Homotopy)
  5. State machines
- Imports FMUs (functional mockup units)

What is CyPhySim?

CyPhySim is an open-source simulator for CPS based on Ptolemy II with:

- Mixed continuous and discrete dynamics
- Superdense time
- Modal models and hybrid systems
- Smooth tokens
- Five simulation engines:
    1. Discrete event simulation (DE)
    2. Quantized-state solvers (QSS)
    3. Runge-Kutta solvers (RK2-3, RK4-5)
    4. Algebraic loop solvers (Substitution, Newton, Homotopy)
    5. State machines
- Imports FMUs (functional mockup units)

This is too much to talk about. Read the paper (Lee et al., 2015).

Fundamental Limits of Modeling

Outline of This Talk:

- The Butterfly Effect
- Discretizing the Continuum
- Limits of Determinism

Determinism does not necessarily imply predictability.
(see e.g. [Thiele and Kumar, EMSOFT 2015])

Fundamental Limits of Modeling

Outline of This Talk:

- The Butterfly Effect
- Discretizing the Continuum
- Limits of Determinism

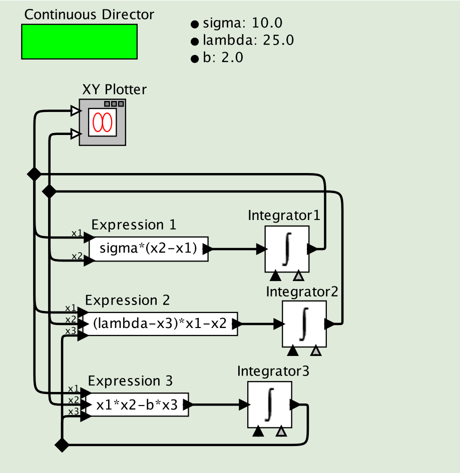
Determinism does not necessarily imply predictability.
(see e.g. [Thiele and Kumar, EMSOFT 2015])

Chaos and the Butterfly Effect

Lorenz attractor:

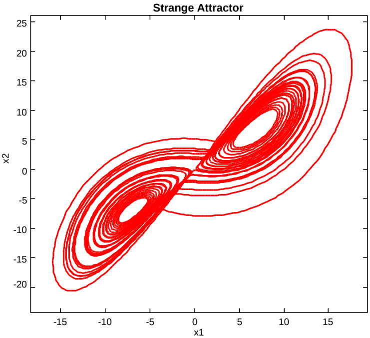
$$\begin{aligned}\dot{x}_1(t) &= \sigma(x_2(t) - x_1(t)) \\ \dot{x}_2(t) &= (\lambda - x_3(t))x_1(t) - x_2(t) \\ \dot{x}_3(t) &= x_1(t)x_2(t) - bx_3(t)\end{aligned}$$

This is a chaotic system, so arbitrarily small perturbations have arbitrarily large consequences.

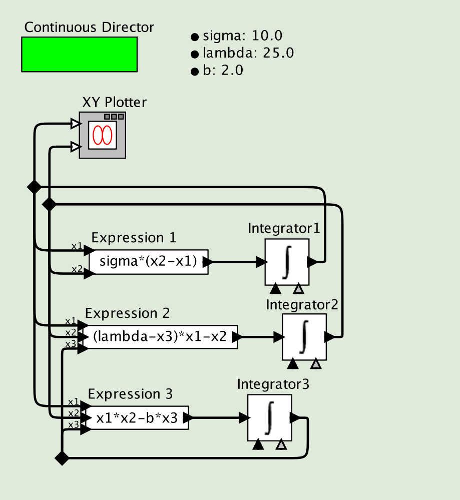


Chaos and the Butterfly Effect

Plot of x_1 vs. x_2:



The error in x_1 and x_2 due to numerical approximation is limited only by the stability of the system.

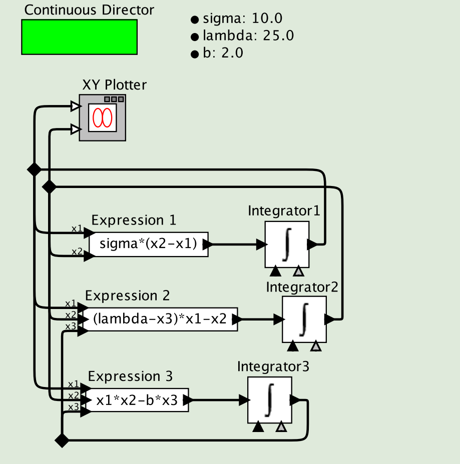


Models

Mathematical:

$$\begin{aligned}
\dot{x}_1(t) &= \sigma(x_2(t) - x_1(t)) \\
\dot{x}_2(t) &= (\lambda - x_3(t))x_1(t) - x_2(t) \\
\dot{x}_3(t) &= x_1(t)x_2(t) - bx_3(t)
\end{aligned}$$

Computational:

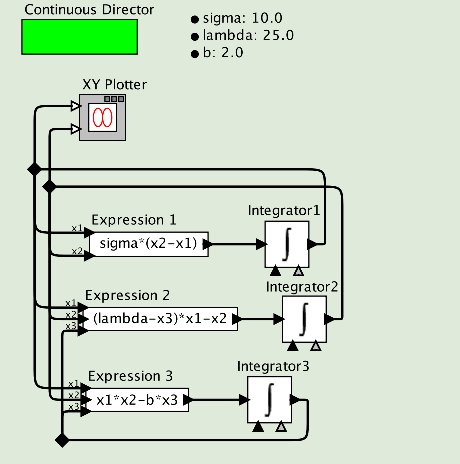


Mathematical:

$$\begin{aligned}\dot{x}_1(t) &= \sigma(x_2(t) - x_1(t))\\ \dot{x}_2(t) &= (\lambda - x_3(t))x_1(t) - x_2(t)\\ \dot{x}_3(t) &= x_1(t)x_2(t) - bx_3(t)\end{aligned}$$

Neither will match the behavior of a physical system being modeled.

Computational:



Newton's Cradle – The Model

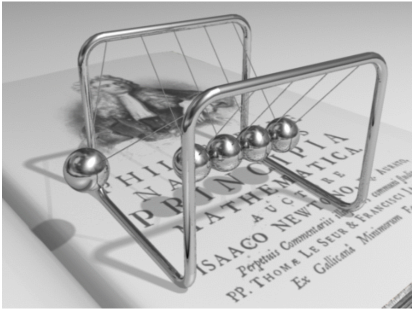
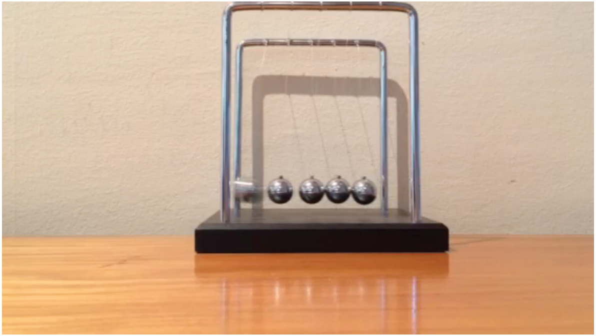


Image by Dominique Toussaint, GNU Free Documentation License, Version 1.2 or later.

Newton's Cradle – A Physical Realization



Model Fidelity

- In *science*, the value of a model lies in how well its behavior matches that of the physical system.
- In *engineering*, the value of the physical system lies in how well its behavior matches that of the model.

In *engineering*, model fidelity is a *two-way street.*

Model Fidelity

- In *science*, the value of a model lies in how well its behavior matches that of the physical system.
- In *engineering*, the value of the physical system lies in how well its behavior matches that of the model.

In *engineering*, model fidelity is a *two-way street.*

Model Fidelity

- In *science*, the value of a model lies in how well its behavior matches that of the physical system.
- In *engineering*, the value of the physical system lies in how well its behavior matches that of the model.

In *engineering*, model fidelity is a *two-way street*.

Cyber Models

Physical System:

Cyber Model:

```java
/** Reset the output receivers, which are the inside receivers of
 *  the output ports of the container.
 *  @exception IllegalActionException If getting the receivers fails.
 */
private void _resetOutputReceivers() throws IllegalActionException {
    List<IOPort> outputs = ((Actor) getContainer()).outputPortList();
    for (IOPort output : outputs) {
        if (_debugging) {
            _debug("Resetting inside receivers of output port: "
                    + output.getName());
        }
        Receiver[][] receivers = output.getInsideReceivers();
        if (receivers != null) {
            for (int i = 0; i < receivers.length; i++) {
                if (receivers[i] != null) {
                    for (int j = 0; j < receivers[i].length; j++) {
                        if (receivers[i][j] instanceof FSMReceiver) {
                            receivers[i][j].reset();
                        }
                    }
                }
            }
        }
    }
}
```

We have learned how to create physical systems whose behavior matches this model extremely well.

Faithful Physical Model of Newton's Cradle?

- localized plastic deformation
- viscous damping
- acoustic wave propagation

But:

- will it actually be more accurate?
- at what cost?

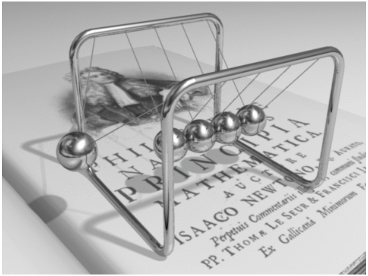


Image by Dominique Toussaint, GNU Free Documentation License, Version 1.2 or later.

I claim that an idealized model, with discrete collisions combined with simple continuous dynamics, is better for most engineering purposes than any more detailed model of the physics.

Faithful Physical Model of Newton's Cradle?

- localized plastic deformation
- viscous damping
- acoustic wave propagation

But:

- will it actually be more accurate?
- at what cost?

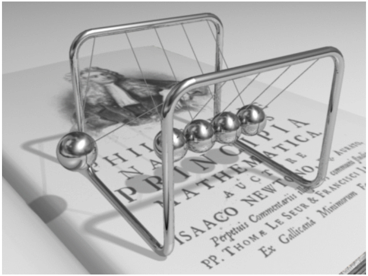


Image by Dominique Toussaint, GNU Free Documentation License, Version 1.2 or later.

I claim that an idealized model, with discrete collisions combined with simple continuous dynamics, is better for most engineering purposes than any more detailed model of the physics.

Fundamental Limits of Modeling

Outline of This Talk:

- The Butterfly Effect
- Discretizing the Continuum
- Limits of Determinism

We need deterministic models that are *also* not chaotic.
KISS.

Fundamental Limits of Modeling

Outline of This Talk:

- The Butterfly Effect
- Discretizing the Continuum
- Limits of Determinism

Newton's Second Law with Impulsive Forces

Consider modeling collisions of masses in motion. Simple $F = ma$ model:

$$\begin{aligned} x(t) &= x(0) + \int_0^t v(\tau) d\tau \\ v(t) &= v(0) + \frac{1}{m} \int_0^t F(\tau) d\tau \end{aligned}$$

With an impulsive force at time T of magnitude F_i:

$$v(t) = v(0) + \frac{1}{m} \int_0^t (F(\tau) + F_i \delta(\tau - T)) d\tau$$

where δ is the Dirac delta function.

Newton's Second Law with Impulsive Forces

Consider modeling collisions of masses in motion. Simple $F = ma$ model:

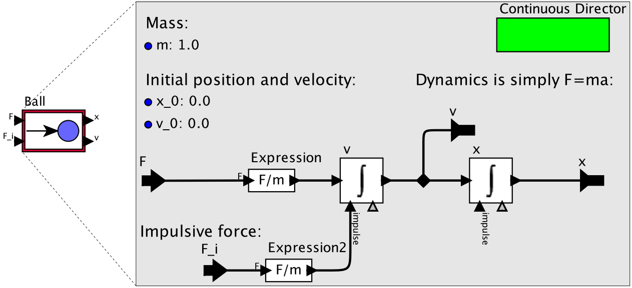
$$
\begin{aligned}
x(t) &= x(0) + \int_0^t v(\tau) d\tau \\
v(t) &= v(0) + \frac{1}{m} \int_0^t F(\tau) d\tau
\end{aligned}
$$

With an impulsive force at time T of magnitude F_i:

$$v(t) = v(0) + \frac{1}{m} \int_0^t (F(\tau) + F_i \delta(\tau - T)) d\tau$$

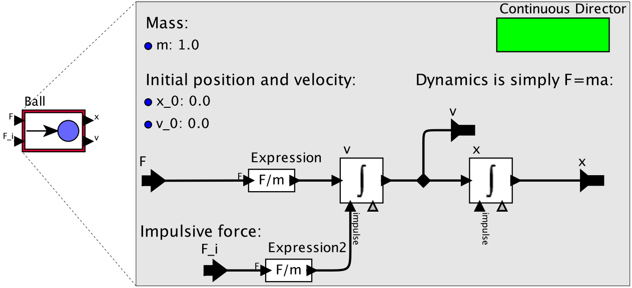
where δ is the Dirac delta function.

Computational Model



NOTE: The output v depends immediately on the input F_i, if it is present.

Computational Model



NOTE: The output v depends immediately on the input F_i, if it is present.

Direct Feedthrough

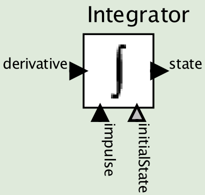
At time t, the *state* output is

$$v(t) = v(0) + \int_{t_0}^{t} \dot{v}(\tau)d\tau,$$

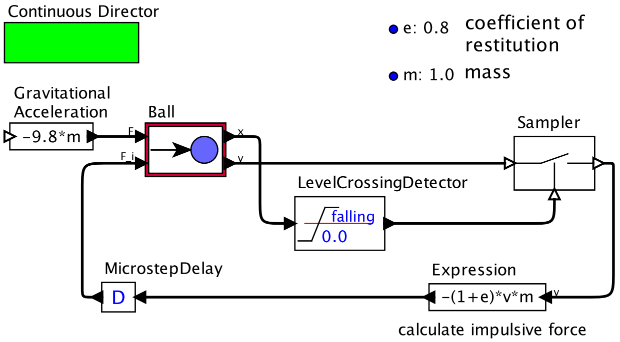
If the *impulse* input is present, then it adds immediately to $v(t)$.

> The output at time t depends on the *impulse* input at time t, but not on the *derivative* input.

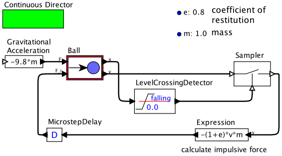
CyPhySim Integrator has "impulse" input:



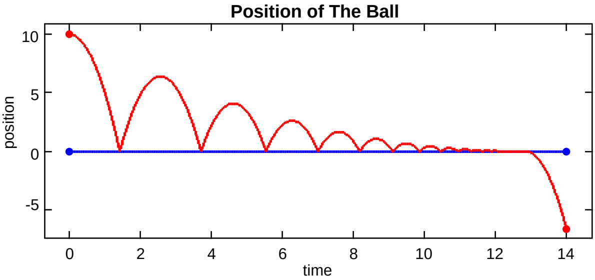
Bouncing Ball Model



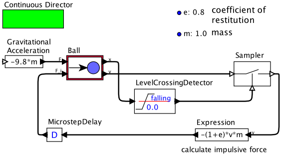
Bouncing Ball Execution



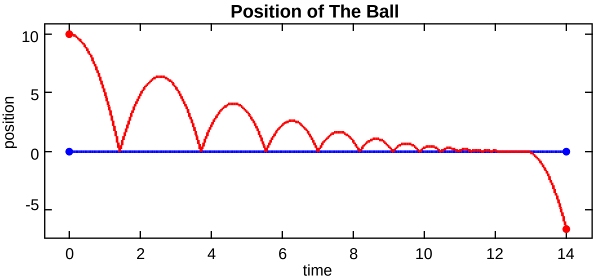
The velocity and position of the ball lie in a continuum. The surface is modeled as discrete.



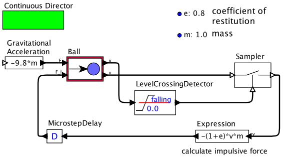
Bouncing Ball Execution



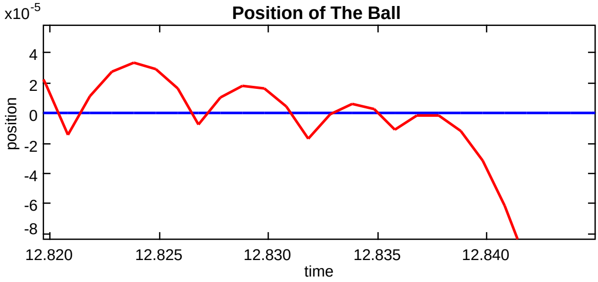
The velocity and position of the ball lie in a continuum. The surface is modeled as discrete.



Bouncing Ball Execution

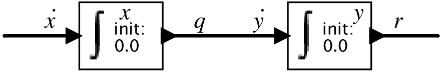


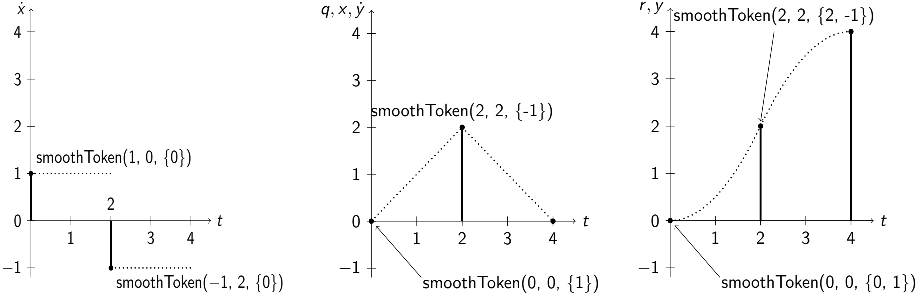
Level-crossing can only be done up to some precision, and the resulting error will inevitably be large enough that the ball tunnels through the surface.



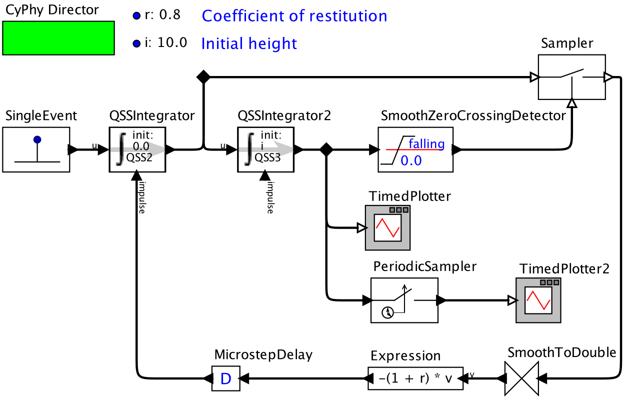
Aside: CyPhySim Innovation: Smooth Tokens

Samples carry derivative information, not just sample value.

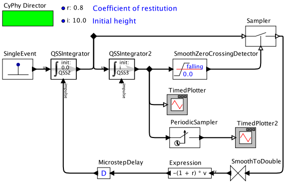




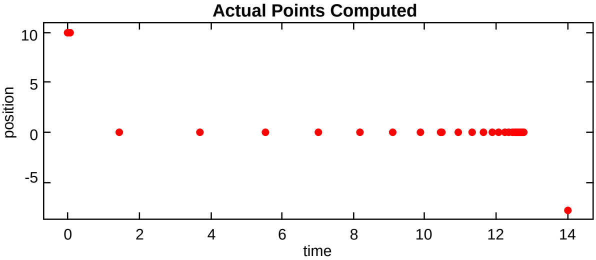
QSS Bouncing Ball Model



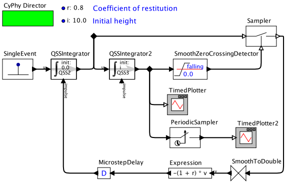
Bouncing Ball Execution



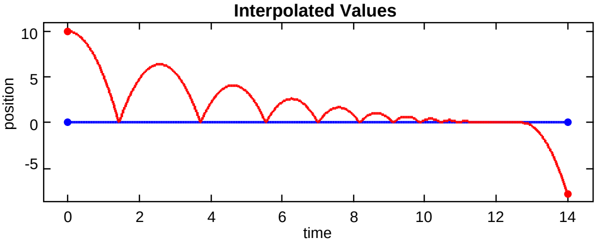
Only compute at times where extrapolation becomes invalid, which in this case is the times of the zero crossings.



Bouncing Ball Execution



Extrapolation from given samples gives an accurate picture of the trajectory.



Properties of QSS

Favoring QSS:

- Zero crossings are *predictable*. No iteration is required to find them.
- Step sizes are *predictable*. No need to reject step sizes and backtrack.
- For some models, QSS is *computationally exact*.
- For linear systems, the error is bounded and controllable.

Favoring classical ODE solvers:

- Inputs may not be quantized.
- Input derivatives may not be known.
- Feedback systems require truncating derivatives, reducing accuracy.
- Feedback systems may oscillate around a steady state.

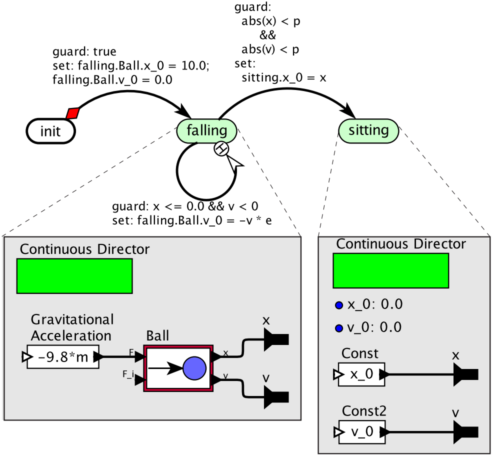
CyPhySim provides both integration strategies.

Regimes of Validity of a Model

Both QSS and classical models tunnel through the surface.

All models are wrong, some are useful.
[Box and Draper, 1987]

Modal models (generalized hybrid systems) split models into *modes*, and a transition system ensures that a mode is active only when the model in that mode is valid.



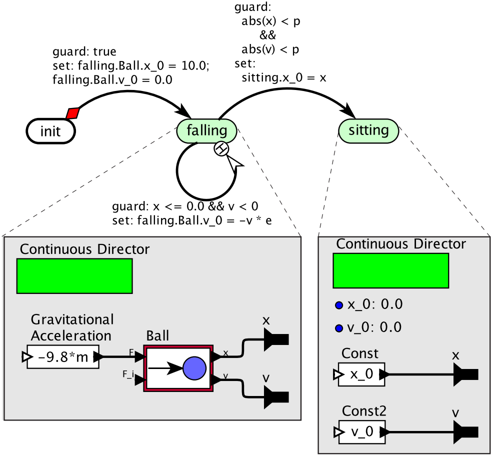
Modal model of the bouncing ball that does not tunnel.

Regimes of Validity of a Model

Both QSS and classical models tunnel through the surface.

All models are wrong, some are useful.
[Box and Draper, 1987]

Modal models (generalized hybrid systems) split models into *modes*, and a transition system ensures that a mode is active only when the model in that mode is valid.



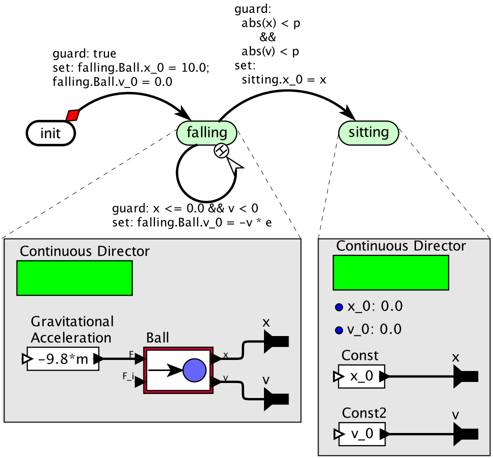
Modal model of the bouncing ball that does not tunnel.

Regimes of Validity of a Model

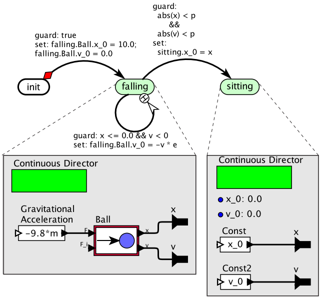
Both QSS and classical models tunnel through the surface.

All models are wrong, some are useful.
[Box and Draper, 1987]

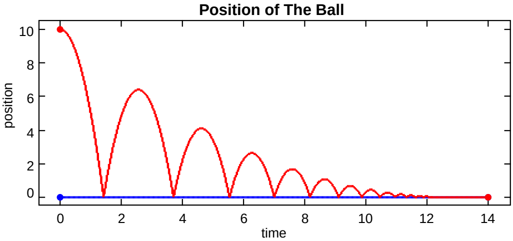
Modal models (generalized hybrid systems) split models into *modes*, and a transition system ensures that a mode is active only when the model in that mode is valid.



Modal model of the bouncing ball that does not tunnel.



Switch out of the free-fall mode when that model is no longer valid.



Fundamental Limits of Modeling

Outline of This Talk:

- The Butterfly Effect
- Discretizing the Continuum
- Limits of Determinism

Mixing continuums and discrete models requires approximation.

1. Use symbolic computation where possible.
2. Be explicit about the regime of validity of a model.

Fundamental Limits of Modeling

Outline of This Talk:

- The Butterfly Effect
- Discretizing the Continuum
- Limits of Determinism

Collisions + Continuous Dynamics

Discrete and continuous, cyber and physical.

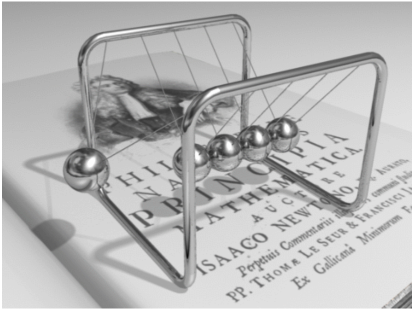
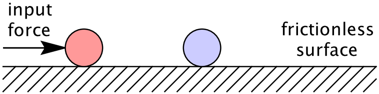


Image by Dominique Toussaint, GNU Free Documentation License, Version 1.2 or later.

Interestingly, even purely physical models illustrate the subtleties.

Collisions: Mixing Discrete and Continuous



Conservation of momentum:

$$m_1 v_1' + m_2 v_2' = m_1 v_1 + m_2 v_2.$$

Conservation of kinetic energy:

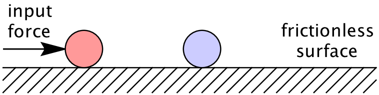
$$\frac{m_1 (v_1')^2}{2} + \frac{m_2 (v_2')^2}{2} = \frac{m_1 (v_1)^2}{2} + \frac{m_2 (v_2)^2}{2}.$$

We have two equations and two unknowns, v_1' and v_2'.

After a Collision

Quadratic problem has two solutions.

Solution 1: $v_1' = v_1$, $v_2' = v_2$
(ignore collision).



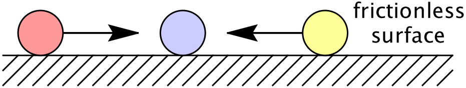
Solution 2:

$$\begin{aligned} v_1' &= \frac{v_1(m_1 - m_2) + 2m_2 v_2}{m_1 + m_2} \\ v_2' &= \frac{v_2(m_2 - m_1) + 2m_1 v_1}{m_1 + m_2}. \end{aligned}$$

Note that if $m_1 = m_2$, then the two masses simply exchange velocities (Newton's cradle).

Simultaneous, Noncausal Collisions

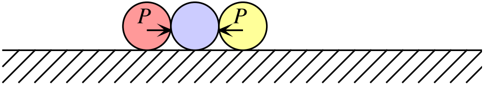
Consider this scenario:



Simultaneous collisions where one collision does not cause the other.

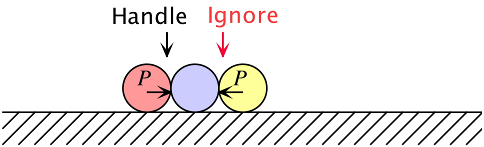
Simultaneous, Noncausal Collisions

One solution: nondeterministic interleaving of the collisions:



At superdense time $(\tau, 0)$, we have two simultaneous collisions.

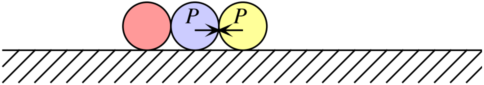
One solution: nondeterministic interleaving of the collisions:



At superdense time $(\tau, 1)$, choose arbitrarily to handle the left collision.

Simultaneous, Noncausal Collisions

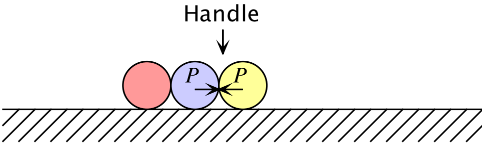
One solution: nondeterministic interleaving of the collisions:



After superdense time $(\tau, 1)$, the momentums are as shown.

Simultaneous, Noncausal Collisions

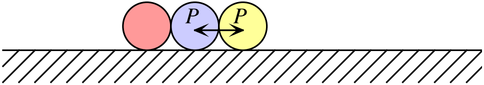
One solution: nondeterministic interleaving of the collisions:



At superdense time $(\tau, 2)$, handle the new collision.

Simultaneous, Noncausal Collisions

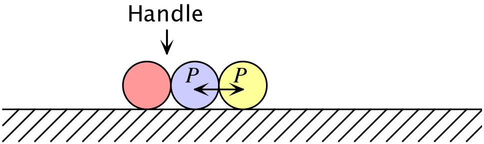
One solution: nondeterministic interleaving of the collisions:



After superdense time $(\tau, 2)$, the momentums are as shown.

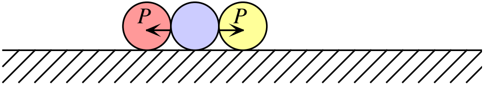
Simultaneous, Noncausal Collisions

One solution: nondeterministic interleaving of the collisions:



At superdense time $(\tau, 3)$, handle the new collision.

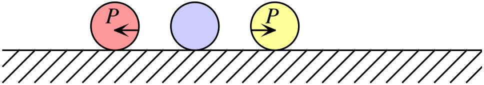
One solution: nondeterministic interleaving of the collisions:



After superdense time $(\tau, 3)$, the momentums are as shown.

Simultaneous, Noncausal Collisions

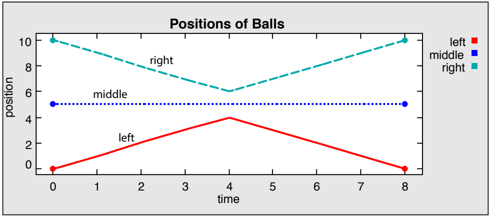
One solution: nondeterministic interleaving of the collisions:



The balls move away at equal speed (if their masses are the same!)

Arbitrary Interleaving

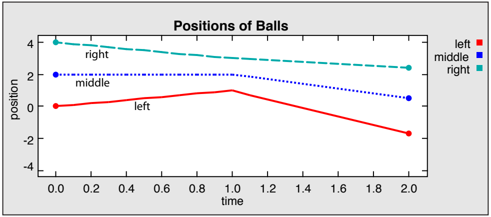
Arbitrary interleaving of the collisions yields the right result (for any choice of interleaving), but only if the masses are the same!



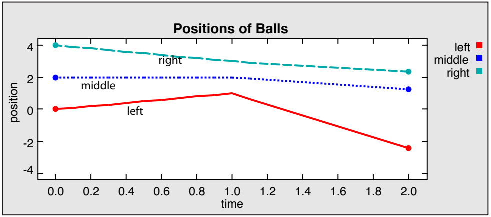
CollisionSimultaneous.xml

Arbitrary Interleaving Yields Nondeterminism

If the masses are different, the behavior depends on which collision is handled first!



(a)

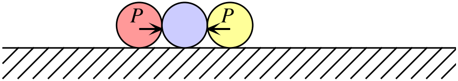


(b)

Recall the Heisenberg Uncertainty Principle

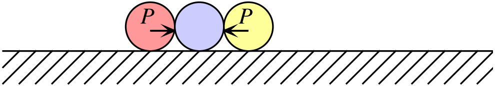
We cannot simultaneously know the position and momentum of an object to arbitrary precision.

But the reaction to these collisions depends on knowing position and momentum precisely.



Heisenberg Uncertainty Principle

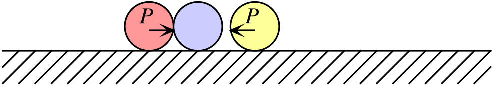
Arbitrary interleaving and nondeterministic results appear to be defensible on physical grounds.



Is Determinism Incomplete?

Let τ be the time between collisions. Consider a sequence of models for $\tau > 0$ where $\tau \to 0$.

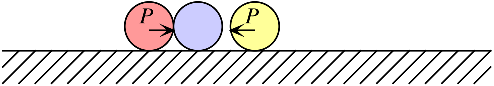
Every model in the sequence is deterministic, but the limit model is not.



In Lee (2014), I show that a direct description of this scenario results in a *non-constructive* model. The nondeterminism arises in making this model constructive.

Is Determinism Incomplete?

Let τ be the time between collisions. Consider a sequence of models for $\tau > 0$ where $\tau \to 0$.



Every model in the sequence is deterministic, but the limit model is not.

In Lee (2014), I show that a direct description of this scenario results in a *non-constructive* model. The nondeterminism arises in making this model constructive.

Fundamental Limits of Modeling

Outline of This Talk:

- The Butterfly Effect
- Discretizing the Continuum
- Limits of Determinism

Deterministic models may become nondeterministic at the limits.

Concluding Remarks

- The discrete, computable world of Cyber Systems is a subset with measure zero of Physical Systems.
- Intuitively, this means that the probability that a randomly chosen physical process can be replicated in the Cyber world is identically zero.
- Engineers, therefore, have it much better than scientists. Our goal is to create physical systems that replicate models that we construct in the Cyber world. Our probabilities are much better!!

Concluding Remarks

- The discrete, computable world of Cyber Systems is a subset with measure zero of Physical Systems.
- Intuitively, this means that the probability that a randomly chosen physical process can be replicated in the Cyber world is identically zero.
- Engineers, therefore, have it much better than scientists. Our goal is to create physical systems that replicate models that we construct in the Cyber world. Our probabilities are much better!!

Concluding Remarks

- The discrete, computable world of Cyber Systems is a subset with measure zero of Physical Systems.
- Intuitively, this means that the probability that a randomly chosen physical process can be replicated in the Cyber world is identically zero.
- Engineers, therefore, have it much better than scientists. Our goal is to create physical systems that replicate models that we construct in the Cyber world. Our probabilities are much better!!

Reading

Lee, E. A., 2014: Constructive models of discrete and continuous physical phenomena. *IEEE Access*, **2(1)**, 1–25. doi:10.1109/ACCESS.2014.2345759.

—, 2015: The past, present, and future of cyber-physical systems: A focus on models. *Sensor*, **15(3)**, 4837–4869. doi:10.3390/s150304837.

Lee, E. A., M. Niknami, T. S. Nouidui, and M. Wetter, 2015: Modeling and simulating cyber-physical systems using CyPhySim. In *EMSOFT*.