Equations, Synchrony, Time, and Modes

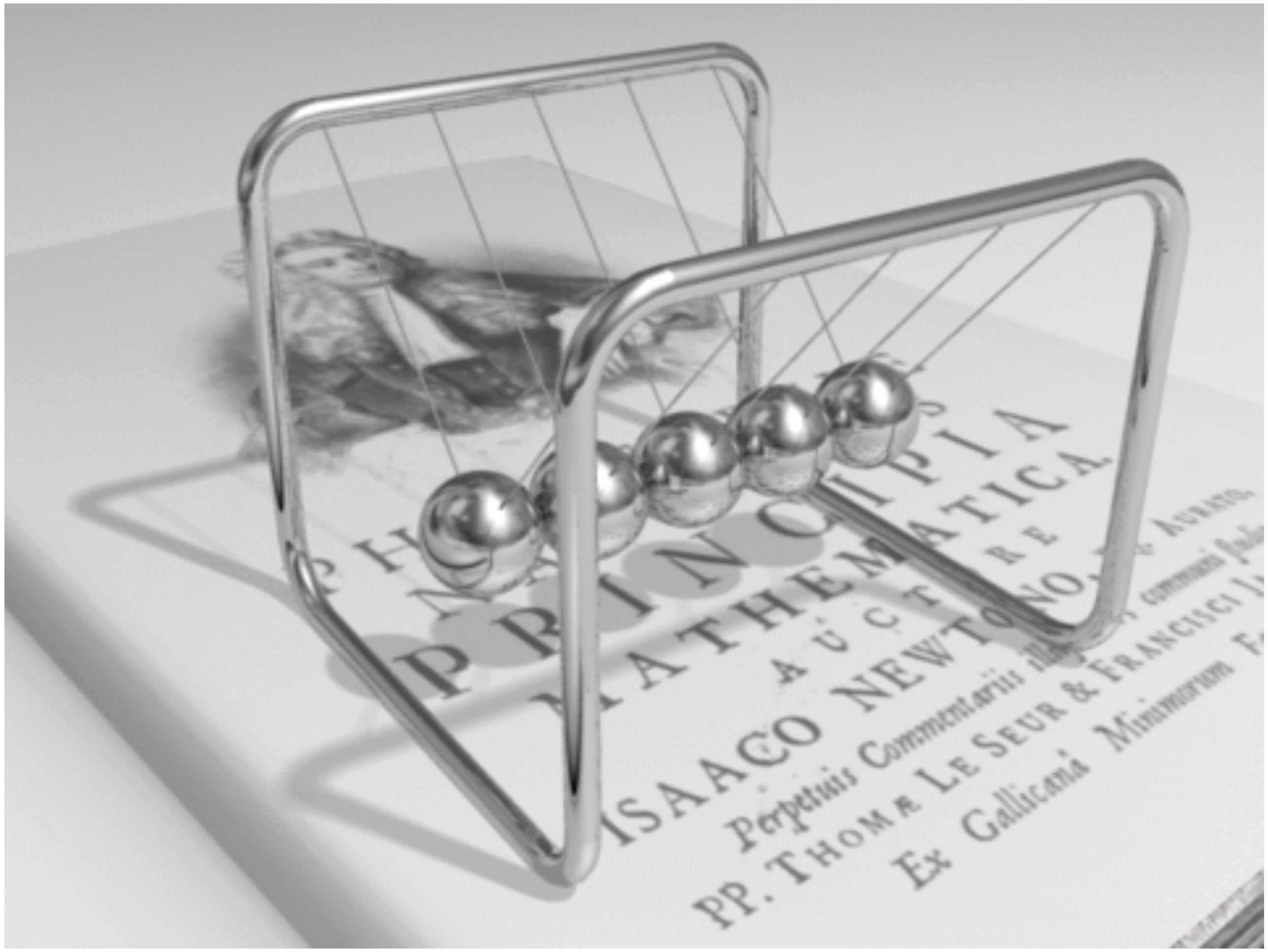
Edward A. Lee

Robert S. Pepper Distinguished Professor
UC Berkeley

Collaborative with:

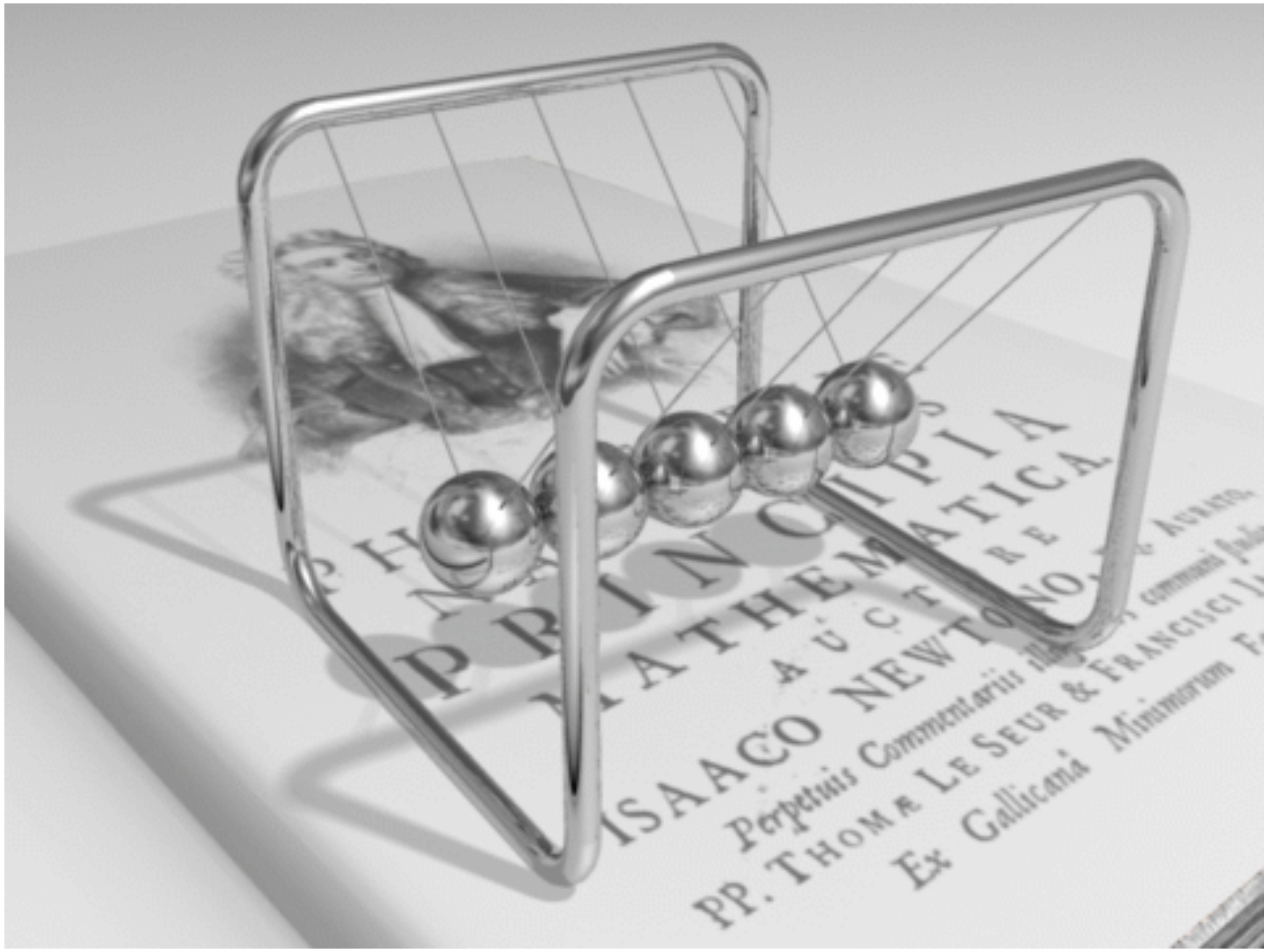
- ***Adam Cataldo***
- ***Patricia Derler***
- ***John Eidson***
- ***Xiaojun Liu***
- ***Eleftherios Matsikoudis***
- ***Haiyang Zheng***

Invited Talk at Workshop:
System Design meets Equation-based Languages: Workshop Program
Lunds, Sweden,
Sept. 18-21



What is the momentum of the middle ball as a function of time?

$$\mathbf{p}(t) = m\mathbf{v}(t)$$

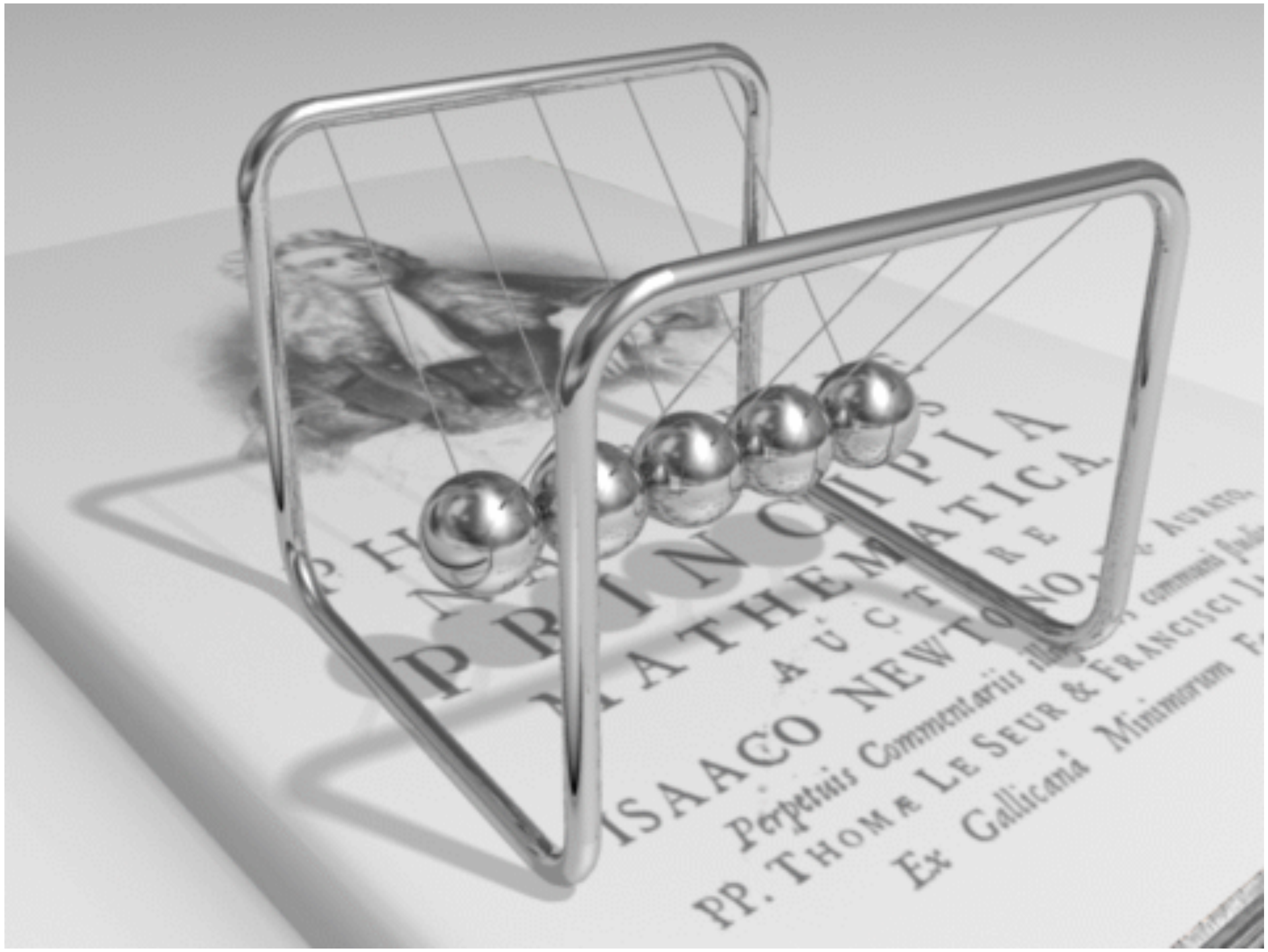


What is the momentum of the middle ball as a function of time?

$$\mathbf{p}(t) = m\mathbf{v}(t)$$

It might seem:

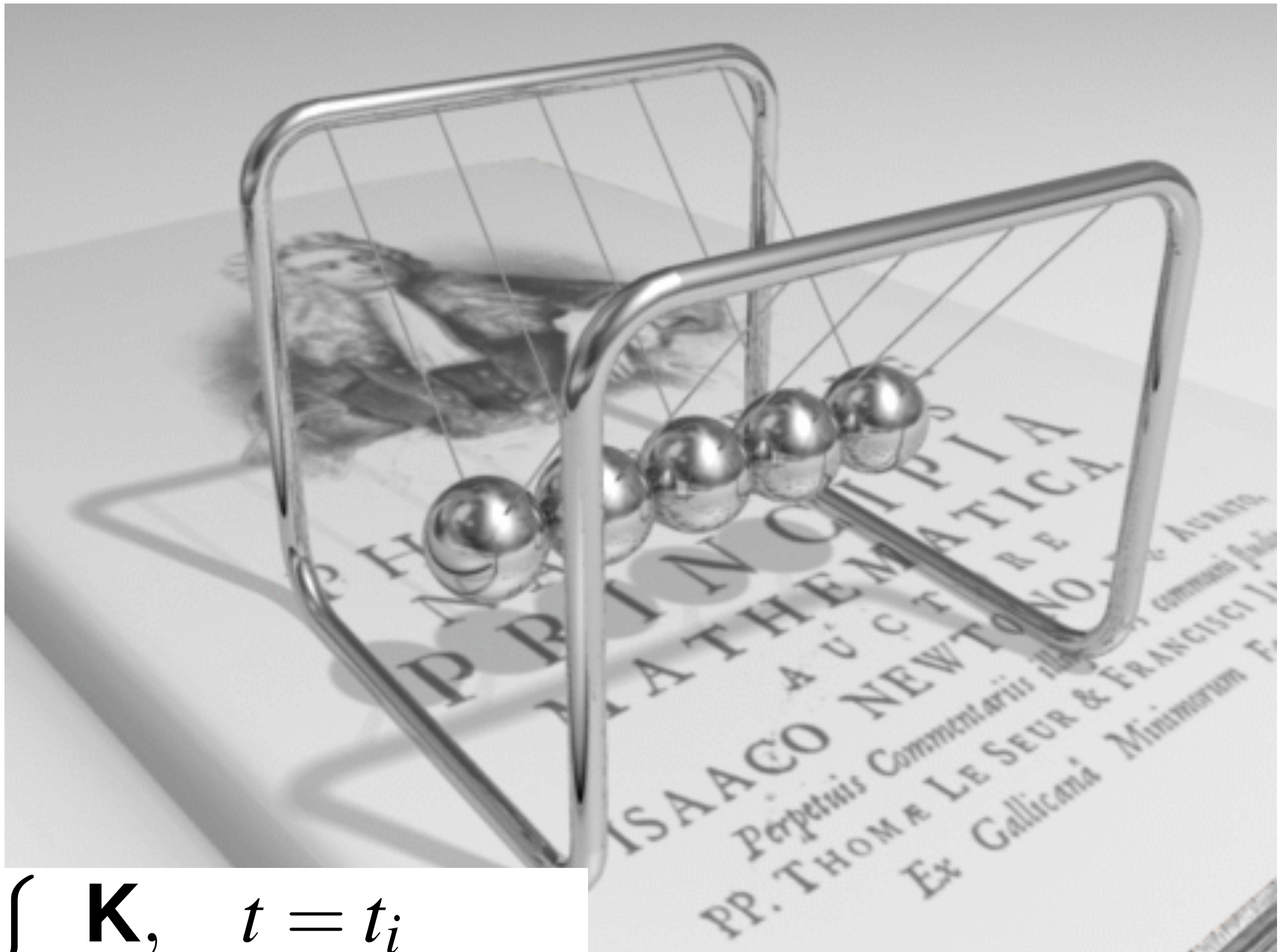
$$\mathbf{v}(t) = 0 \quad \Rightarrow \quad \mathbf{p}(t) = 0$$



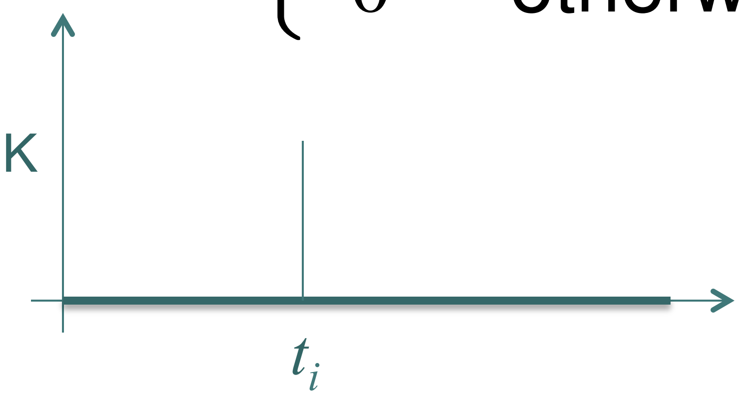
But no, it is:

$$\mathbf{v}(t) = \begin{cases} \mathbf{K}, & t = t_i \\ 0 & \text{otherwise} \end{cases}$$

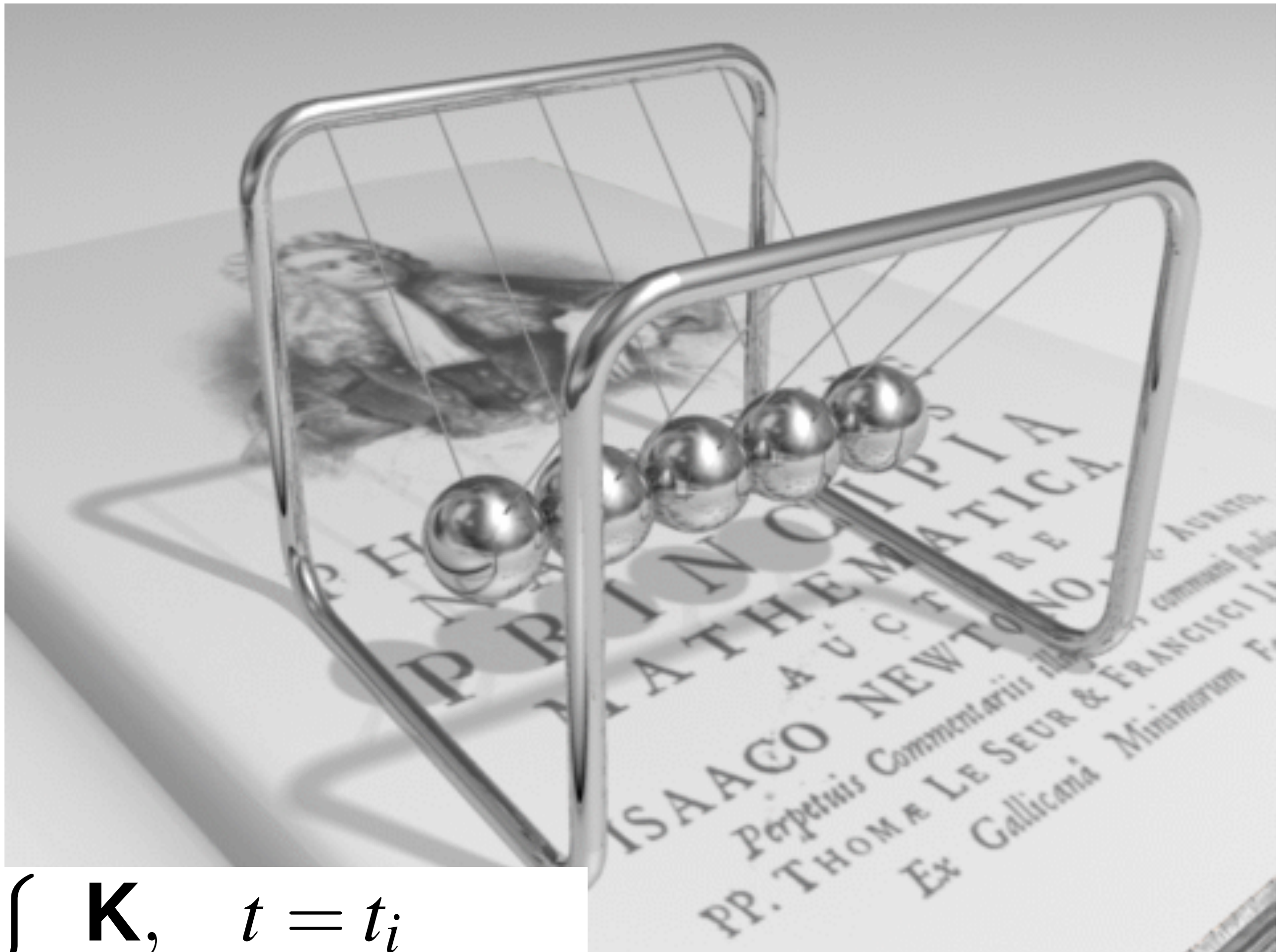
where t_i is the time of collision



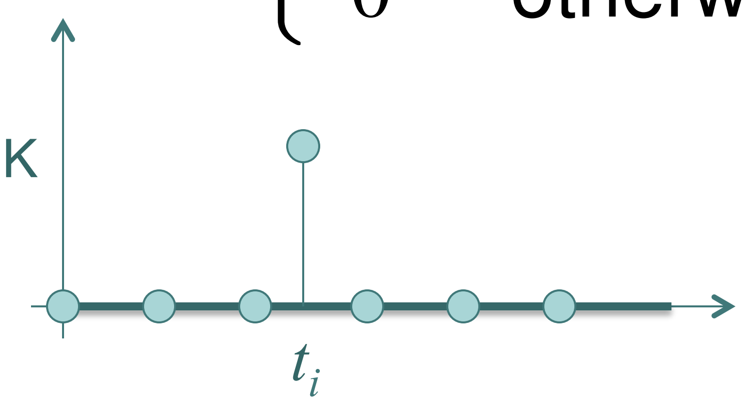
$$\mathbf{v}(t) = \begin{cases} \mathbf{K}, & t = t_i \\ 0 & \text{otherwise} \end{cases}$$



Since position is the integral of velocity, and the integral of **v** is zero, the ball does not move.



$$\mathbf{v}(t) = \begin{cases} \mathbf{K}, & t = t_i \\ 0 & \text{otherwise} \end{cases}$$

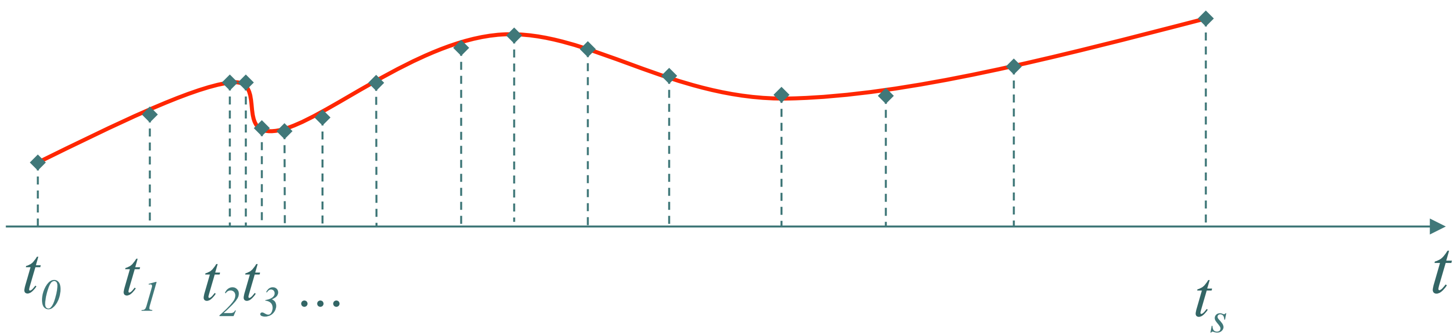


A *discrete* representation of this signal with *samples* is inadequate.

Samples yield discrete signals

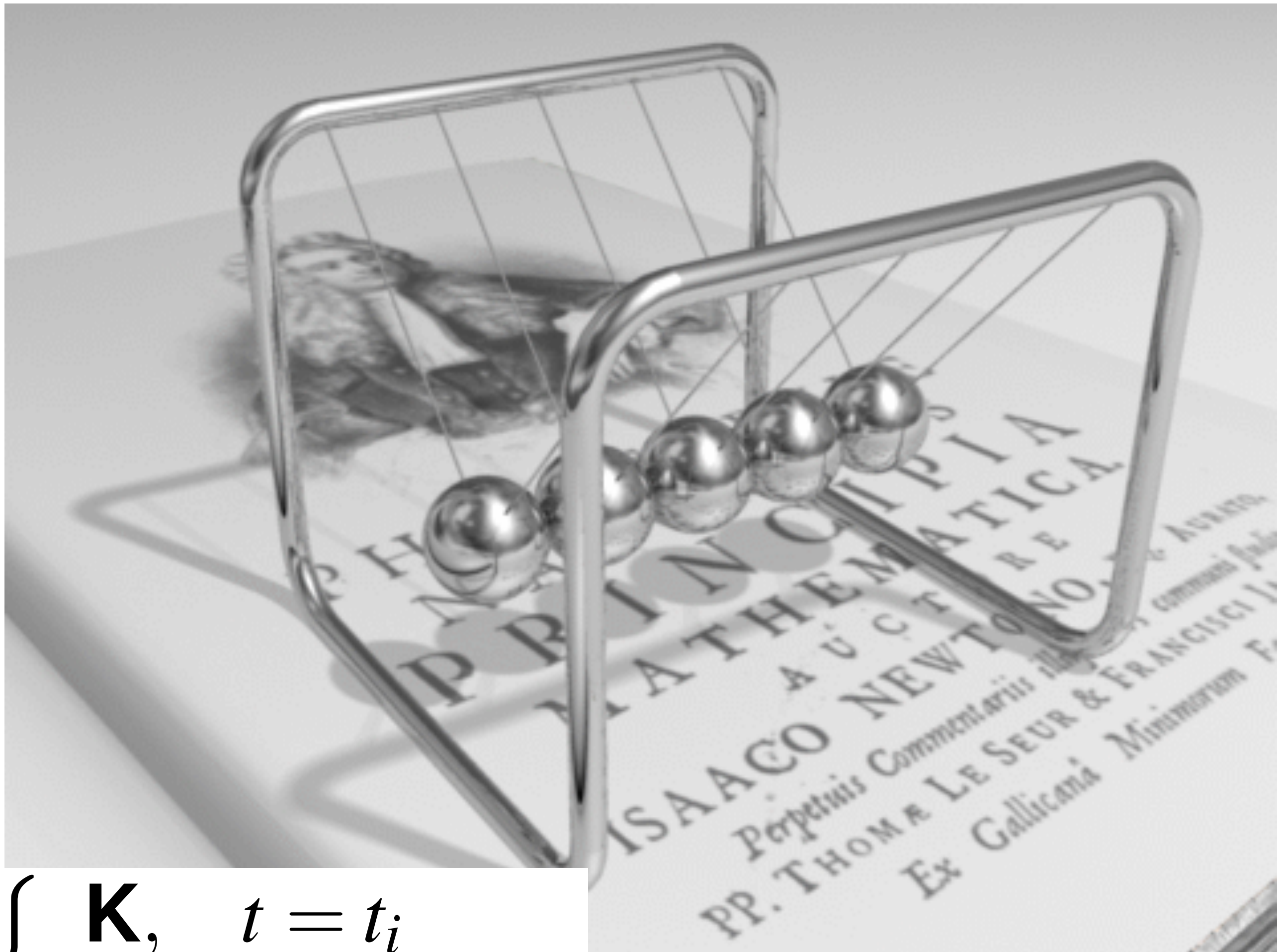
A signal $s: T \rightarrow D$ *is sampled at tags*

$$\pi(s) = \{t_0, t_1, \ldots\} \subset T$$

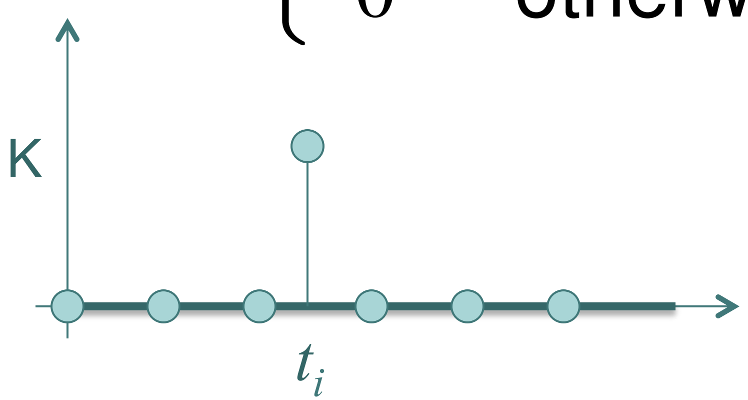


A signal s is *discrete* if there is an *order embedding* from its tag set $\pi(s)$ (the tags for which it is defined and not absent) to the natural numbers (under their usual order).

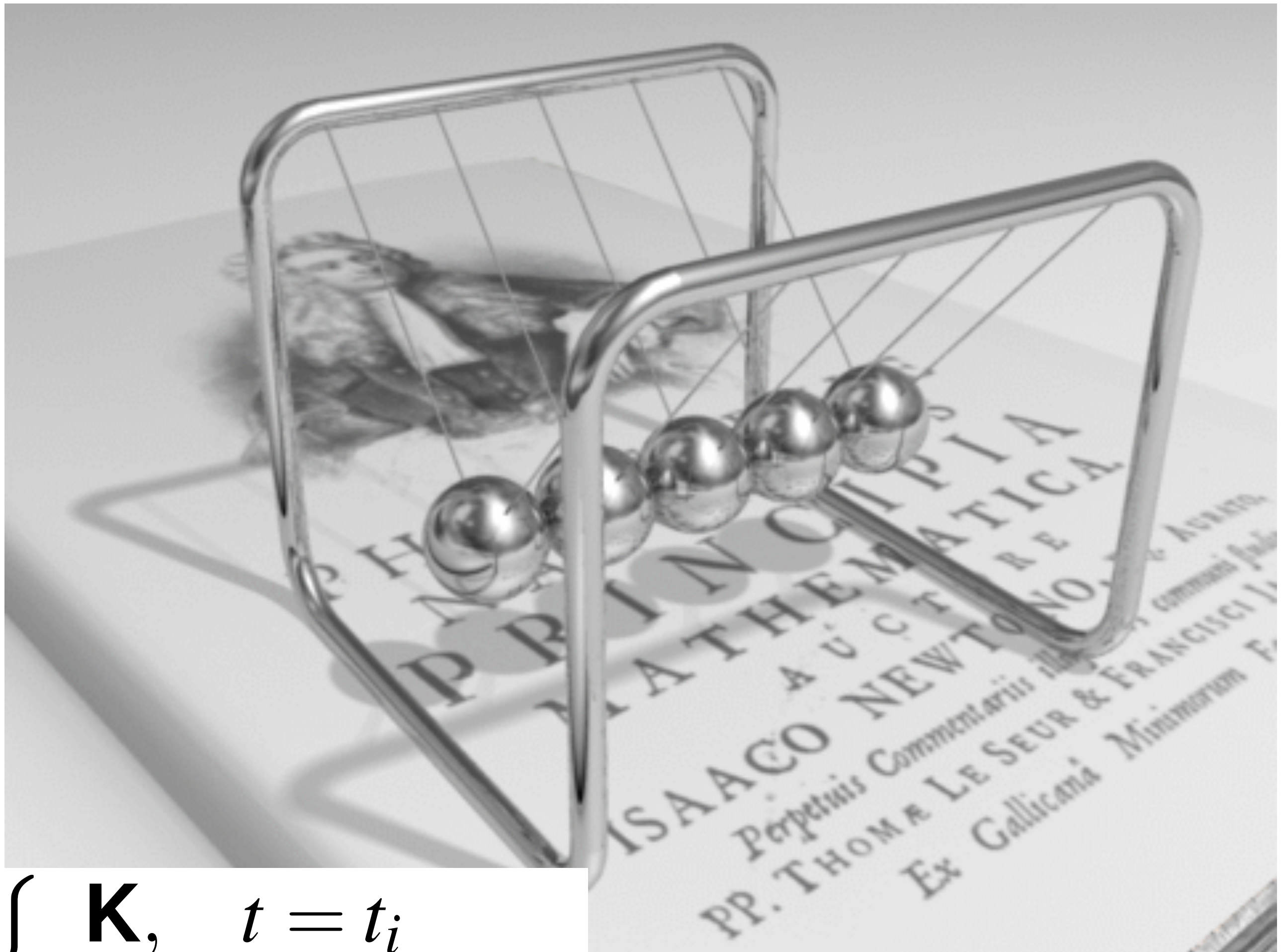
Note: Benveniste et al. use a different (and less useful?) notion of "discrete."



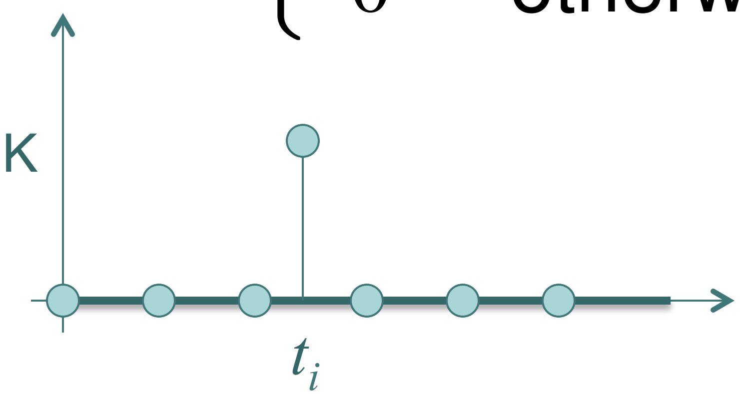
$$\mathbf{v}(t) = \begin{cases} \mathbf{K}, & t = t_i \\ 0 & \text{otherwise} \end{cases}$$



No discrete subset of real-valued times is adequate to unambiguously represent this signal.



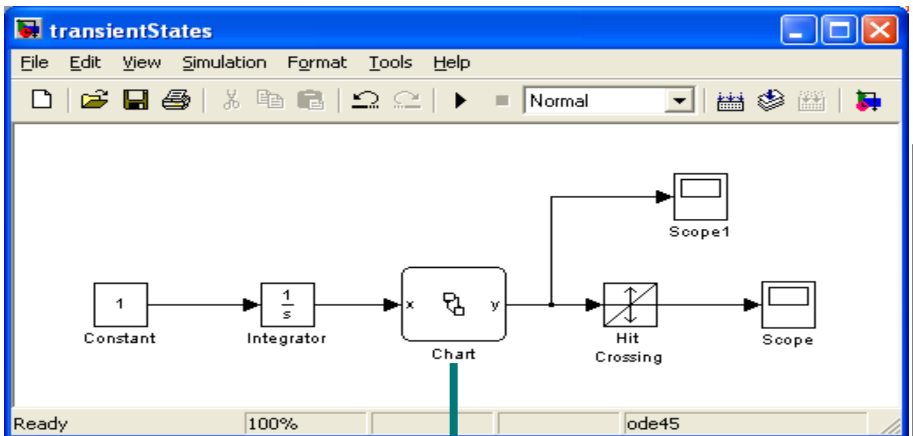
$$\mathbf{v}(t) = \begin{cases} \mathbf{K}, & t = t_i \\ 0 & \text{otherwise} \end{cases}$$



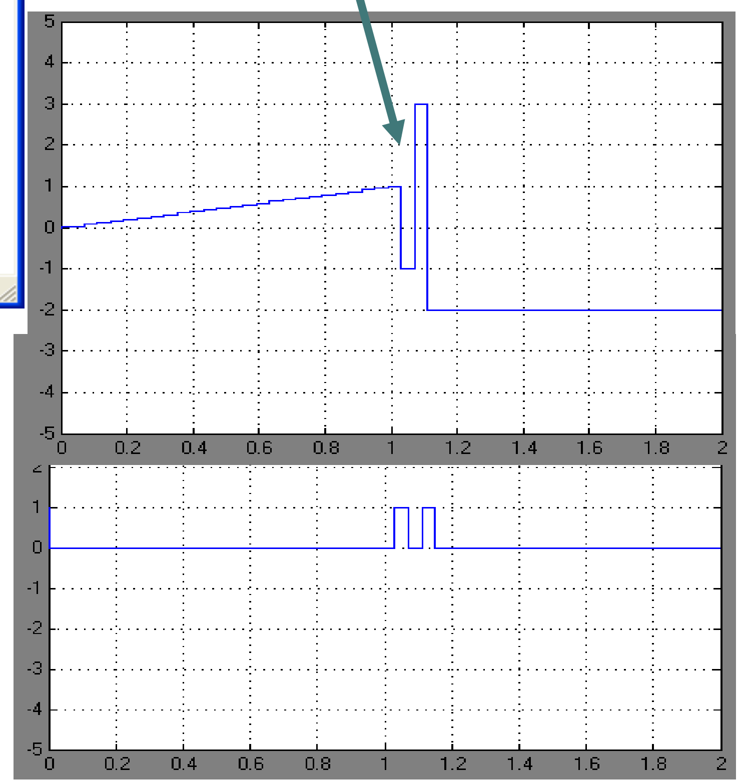
There is no *semantic distinction* between a discrete event and a rapidly varying continuous signal.

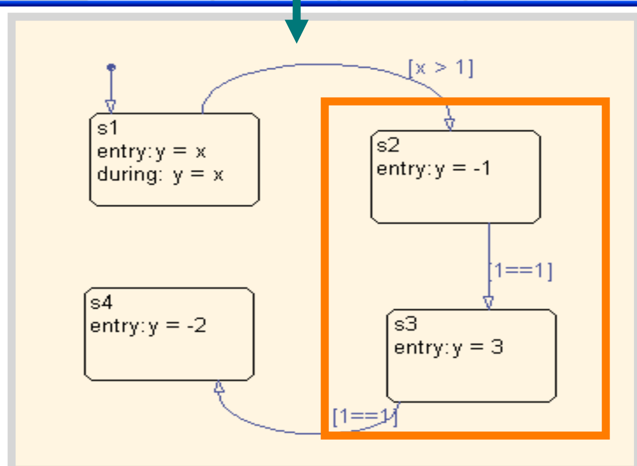
Simulink/Stateflow cannot accurately model such events.

In Simulink, a signal can only have one value at a given time. Hence Simulink introduces solver-dependent behavior.



The simulator engine of Simulink introduces a non-zero delay to consecutive transitions.





Transient States

Ptolemy II uses *Superdense Time*

[Maler, Manna, Pnuelli, 92]

for Continuous-Time Signals

$$\mathbf{v} : (\mathbb{R} \times \mathbb{N}) \to \mathbb{R}^3$$

Initial value: $$\mathbf{v}(t_i, 0) = 0$$

Intermediate value: $$\mathbf{v}(t_i, 1) = \mathbf{K}$$

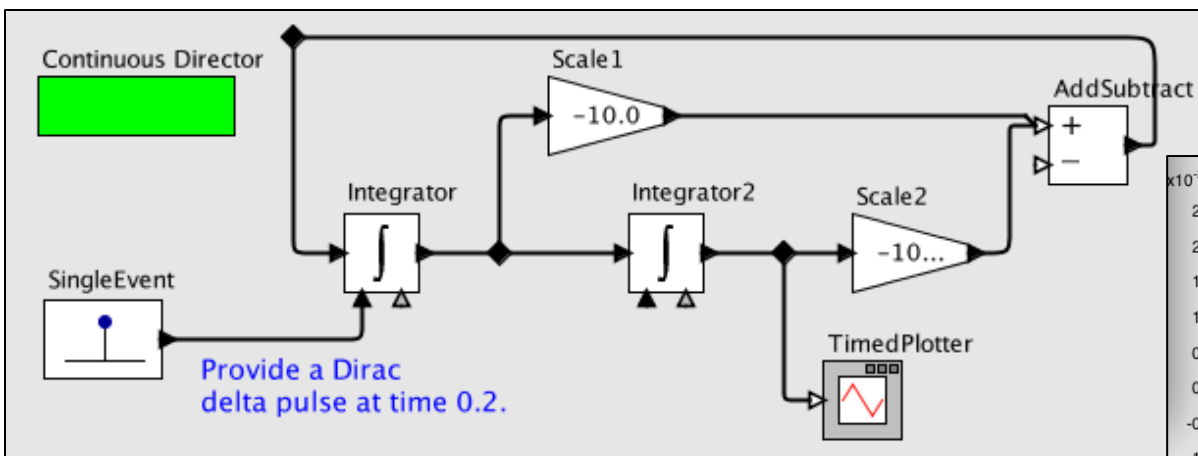
Final value: $$\mathbf{v}(t_i, n) = 0, \quad n \geq 2$$

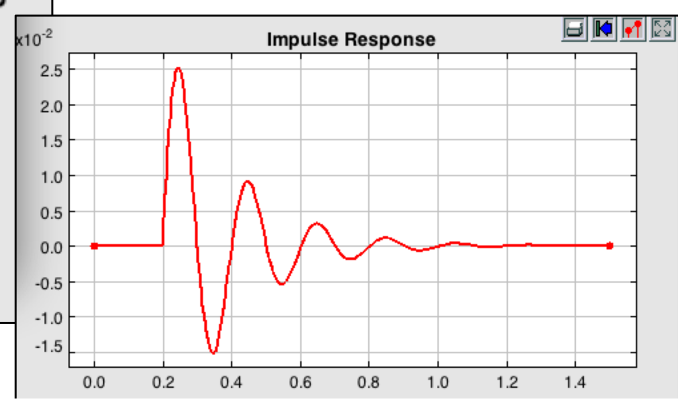
At each **tag**, the signal has *exactly one value*. At each time point, the signal has *a sequence of values*. Signals are *piecewise continuous*, in a well-defined technical sense, a property that makes ODE solvers work well.

Consequences of using Superdense Time

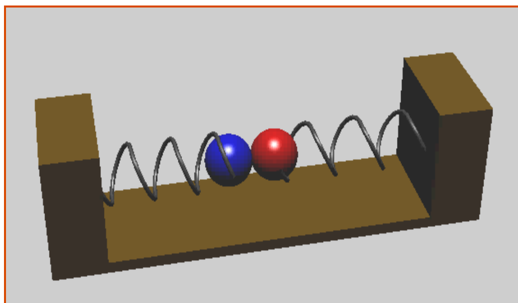
- Transient states are well represented:

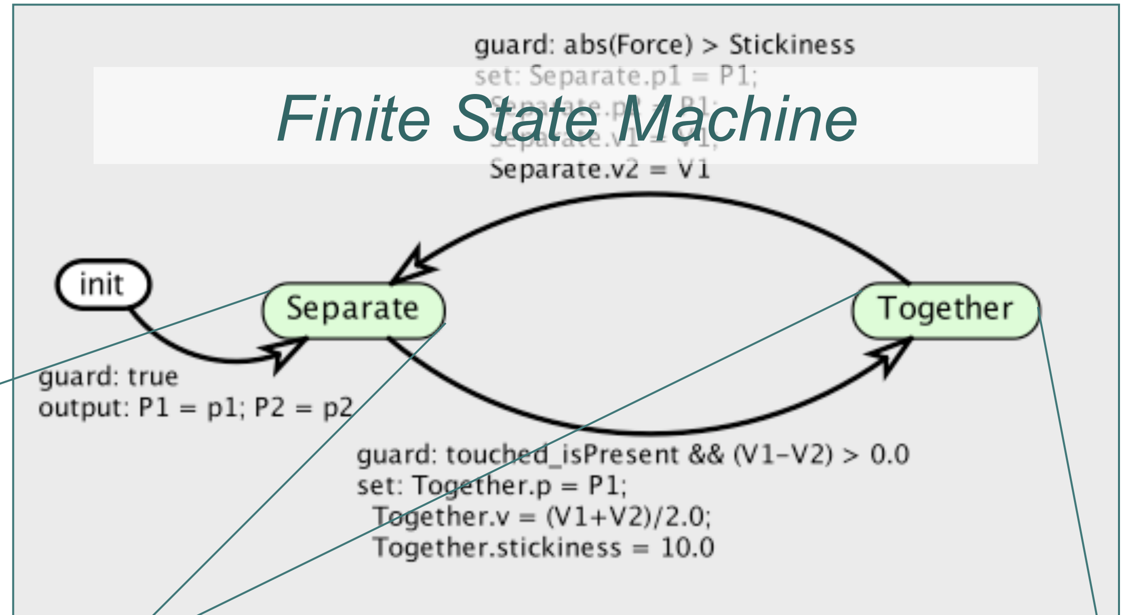
- Infinitessimals (even Dirac delta functions):

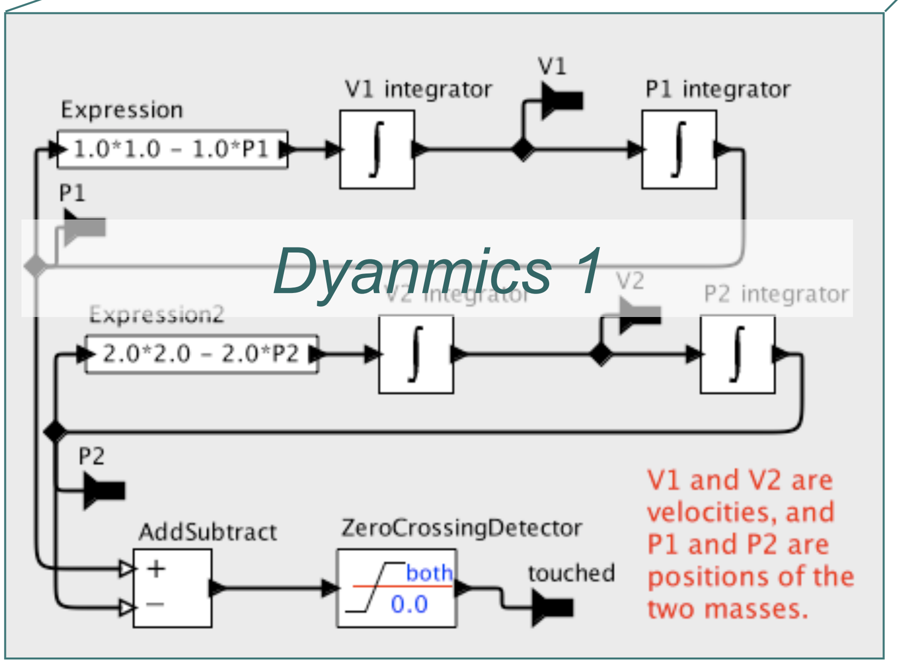


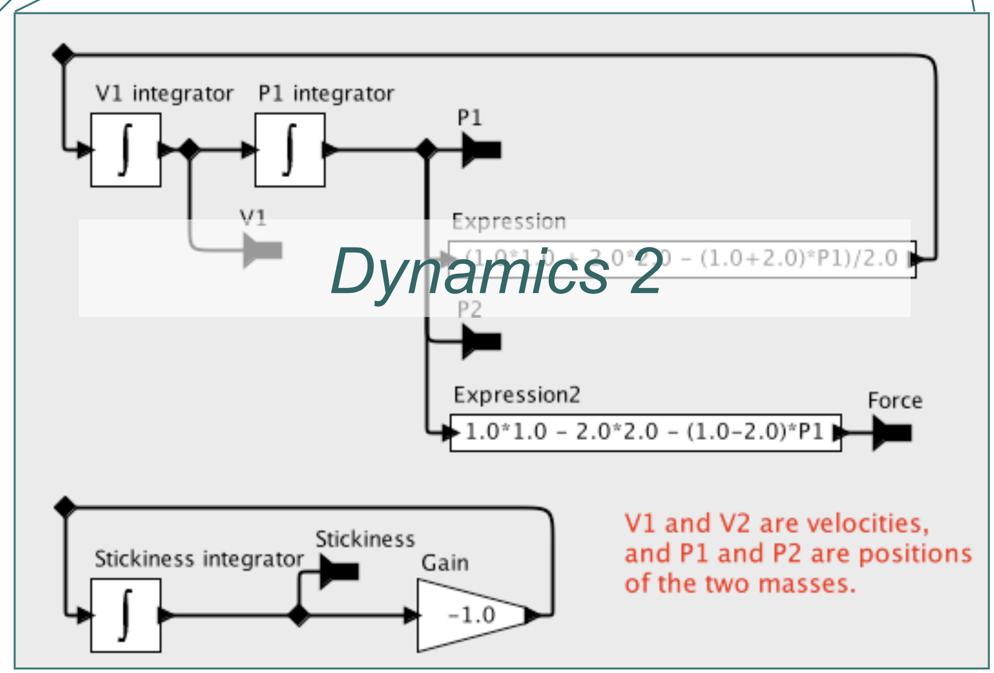


More Consequences: Hybrid System

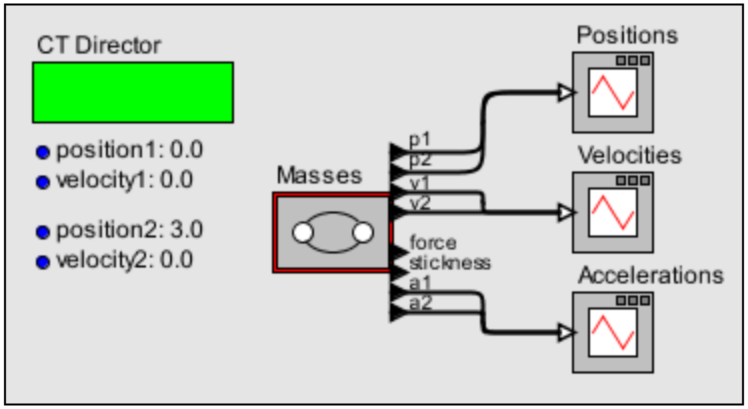


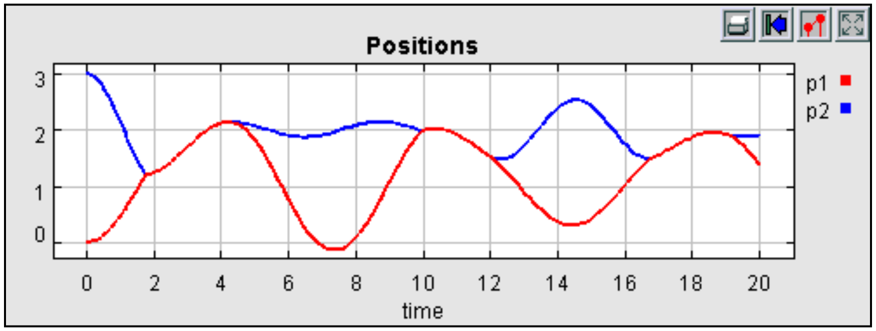


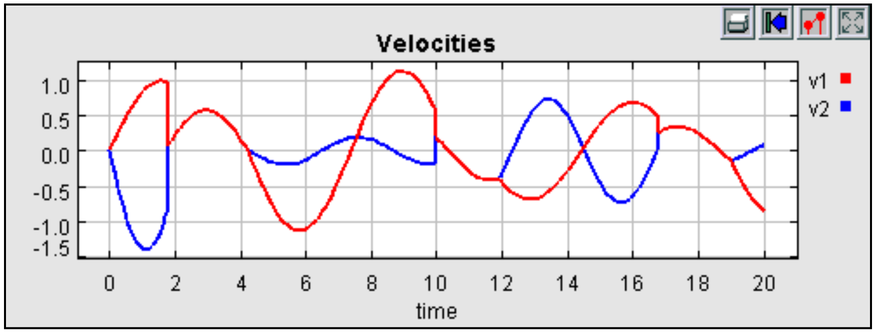




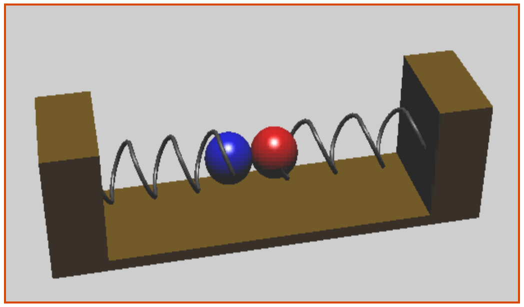
Transitions between modes are instantaneous

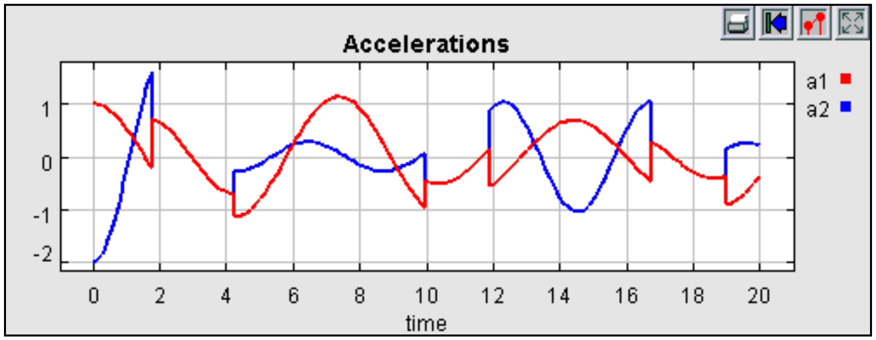




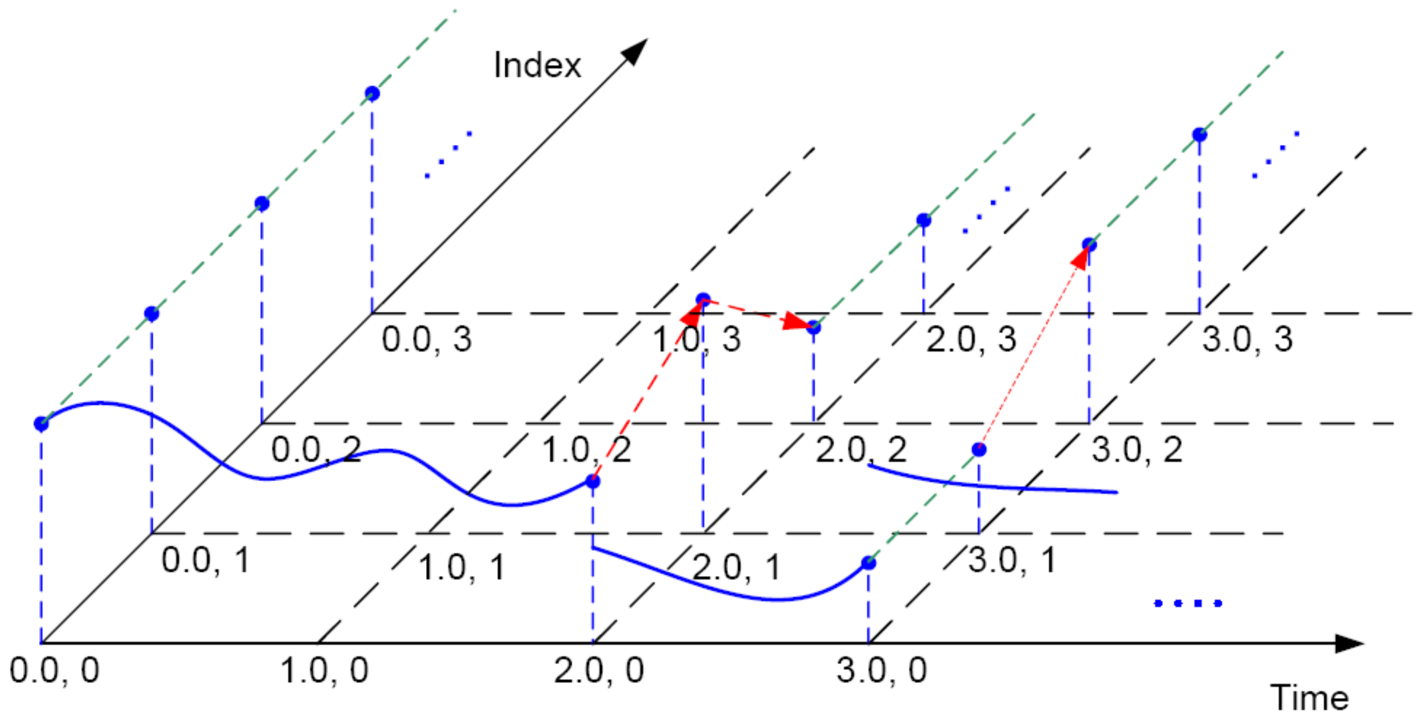


In the signals at the right, the velocities and accelerations proceed through a sequence of values at the times of the collisions and separations.





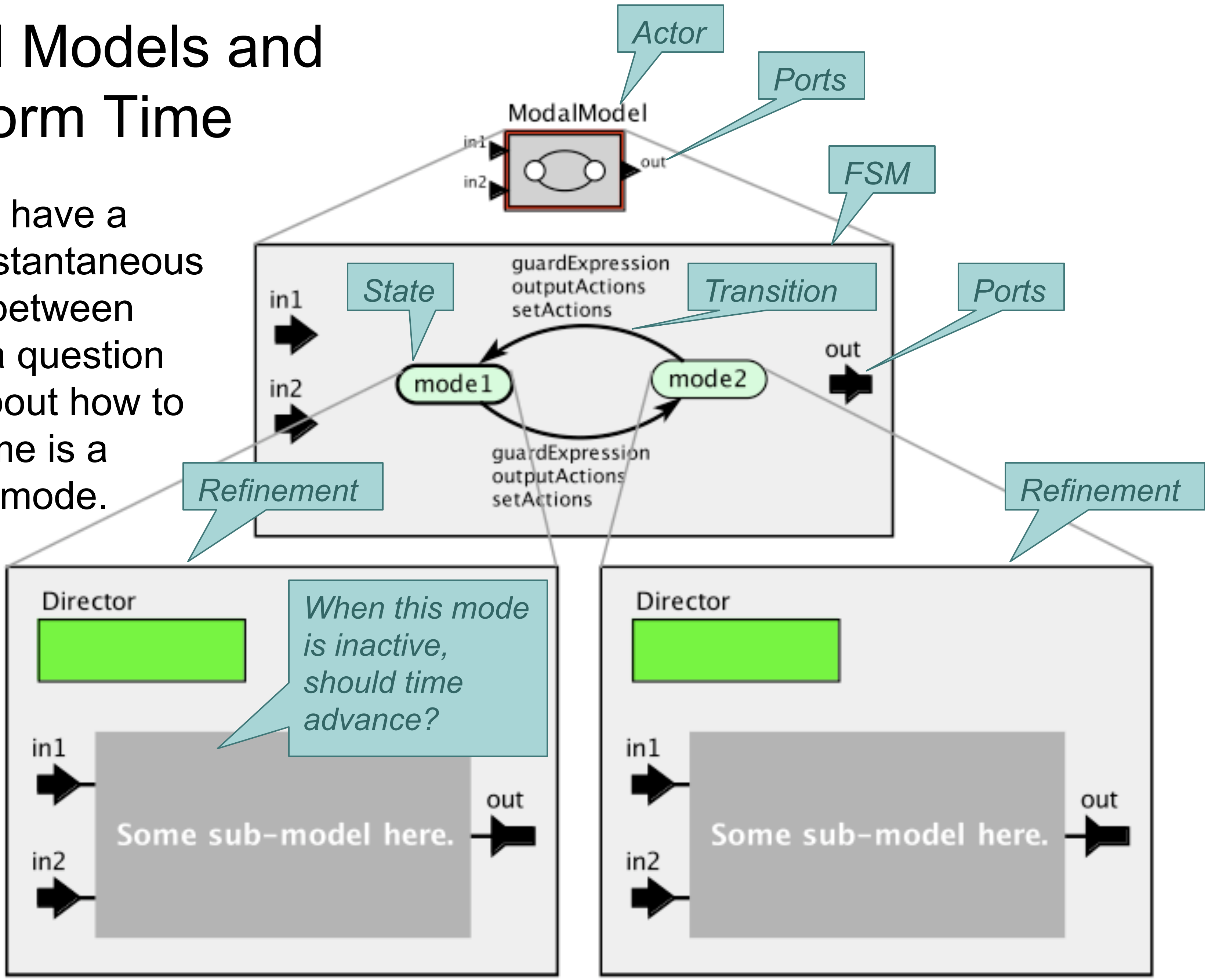
Superdense Time



The red arrows indicate value changes between tags, which correspond to discontinuities. Signals are continuous from the left *and* continuous from the right at points of discontinuity.

Modal Models and Multiform Time

Once we have a clean, instantaneous handoff between modes, a question arises about how to model time is a *dormant* mode.



The Modal Model Muddle

It's about time

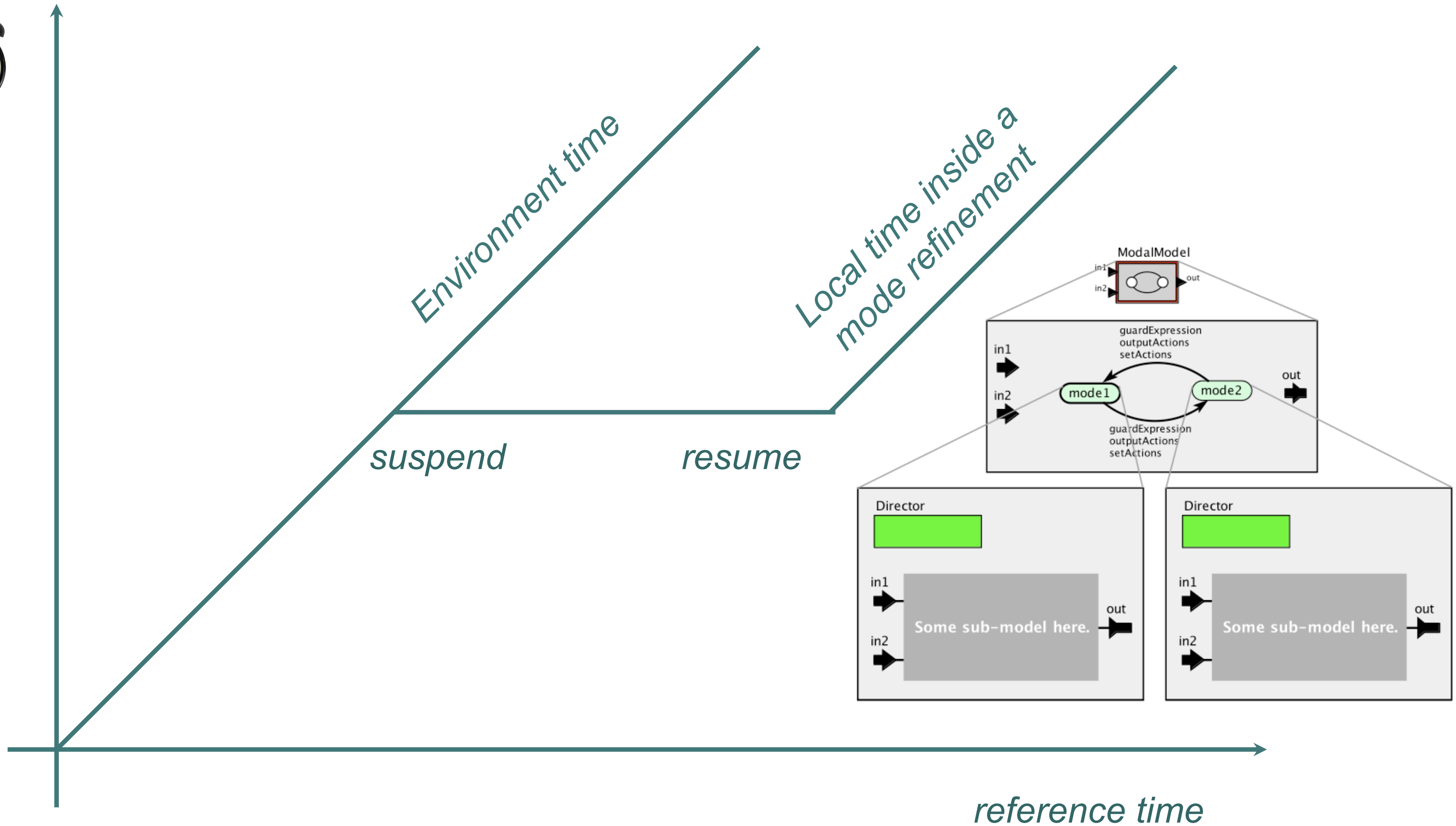
After trying several variants on the semantics of modal time, we settled on this:

A mode refinement has a *local* notion of time. When the mode refinement is inactive, local time does not advance. Local time has a monotonically increasing gap relative to environment time.

MultiForm Time in Ptolemy II

In Ptolemy II Modal Models,
Time is suspended and resumed

local time



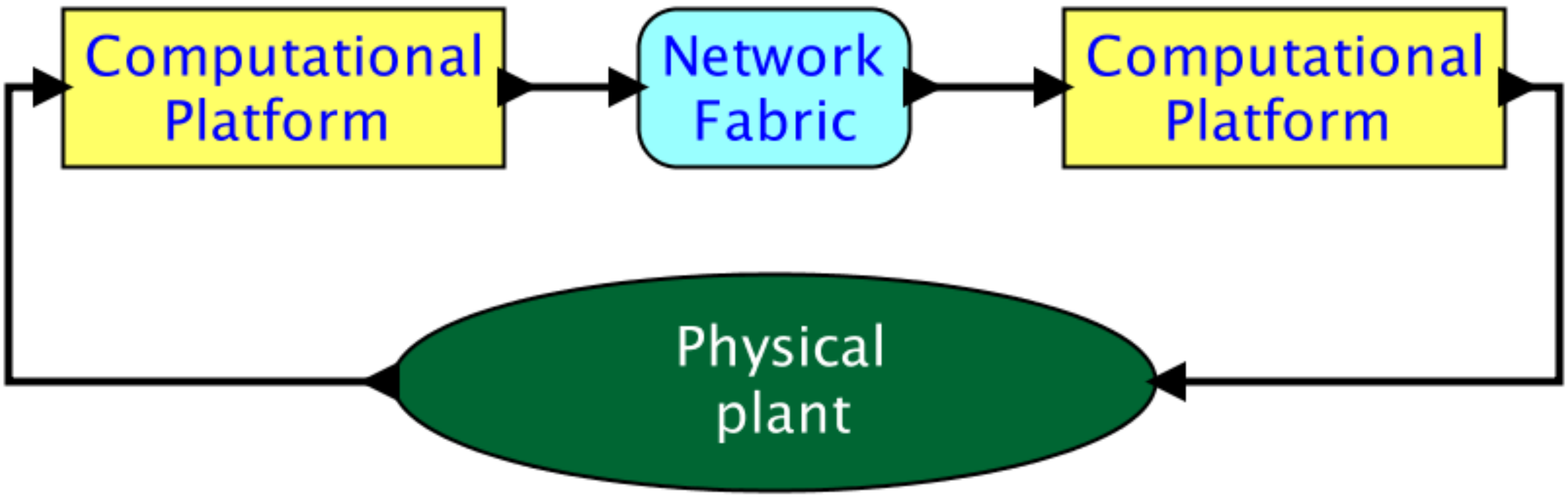
Variants for the Semantics of Modal Time that we Tried or Considered, but that Failed

- Mode refinement executes while "inactive" but inputs are not provided and outputs are not observed.
- Time advances while mode is inactive, and mode refinement is responsible for "catching up."
- Mode refinement is "notified" when it has requested time increments that are not met because it is inactive.
- When a mode refinement is re-activated, it resumes from its first missed event.

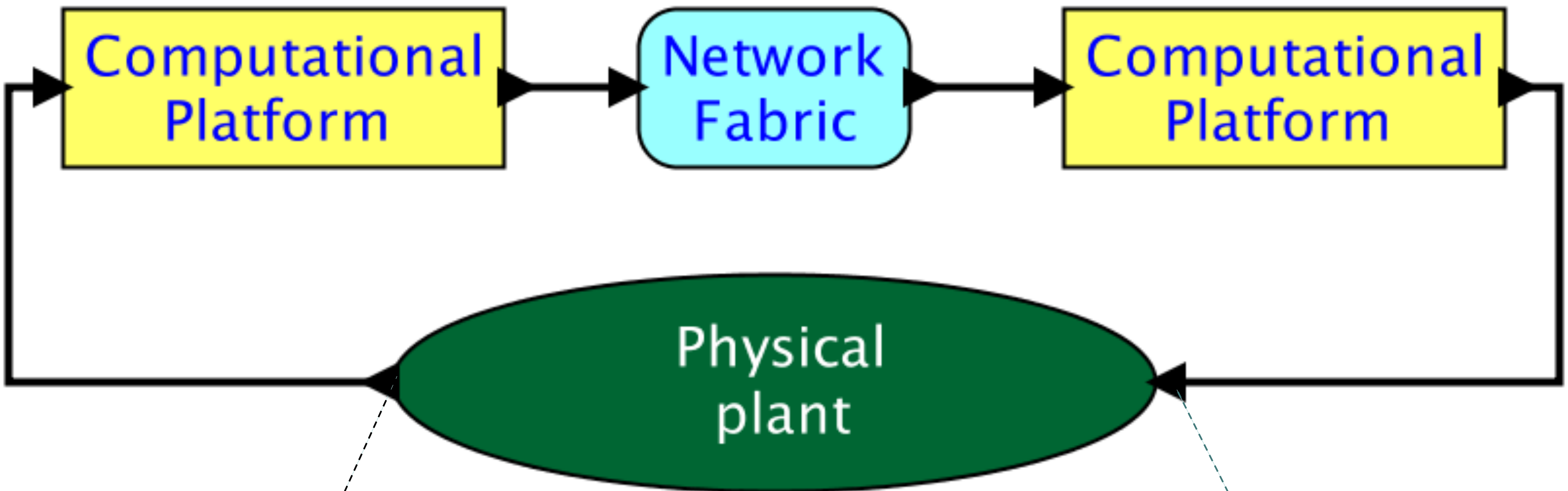
All of these led to some very strange models…

Final solution: Local time does not advance while a mode is inactive. Monotonically growing gap between local time and environment time.

Once we have multiform time, we can build accurate models of cyber-physical systems



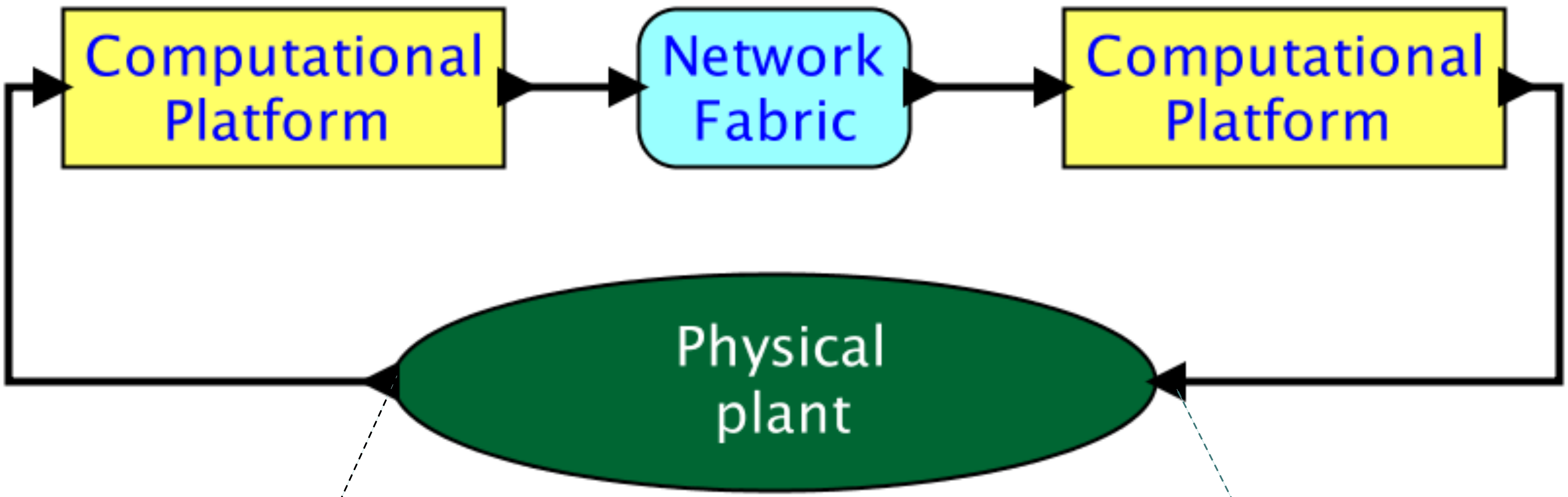
Engineers model physical dynamics using differential-algebraic equations.



The variable t represents an idealized Newtonian notion of time.

$$\dot{\theta}(t) = \dot{\theta}(0) + \frac{1}{I}\int_0^t \mathbf{T}(\tau)\,d\tau$$

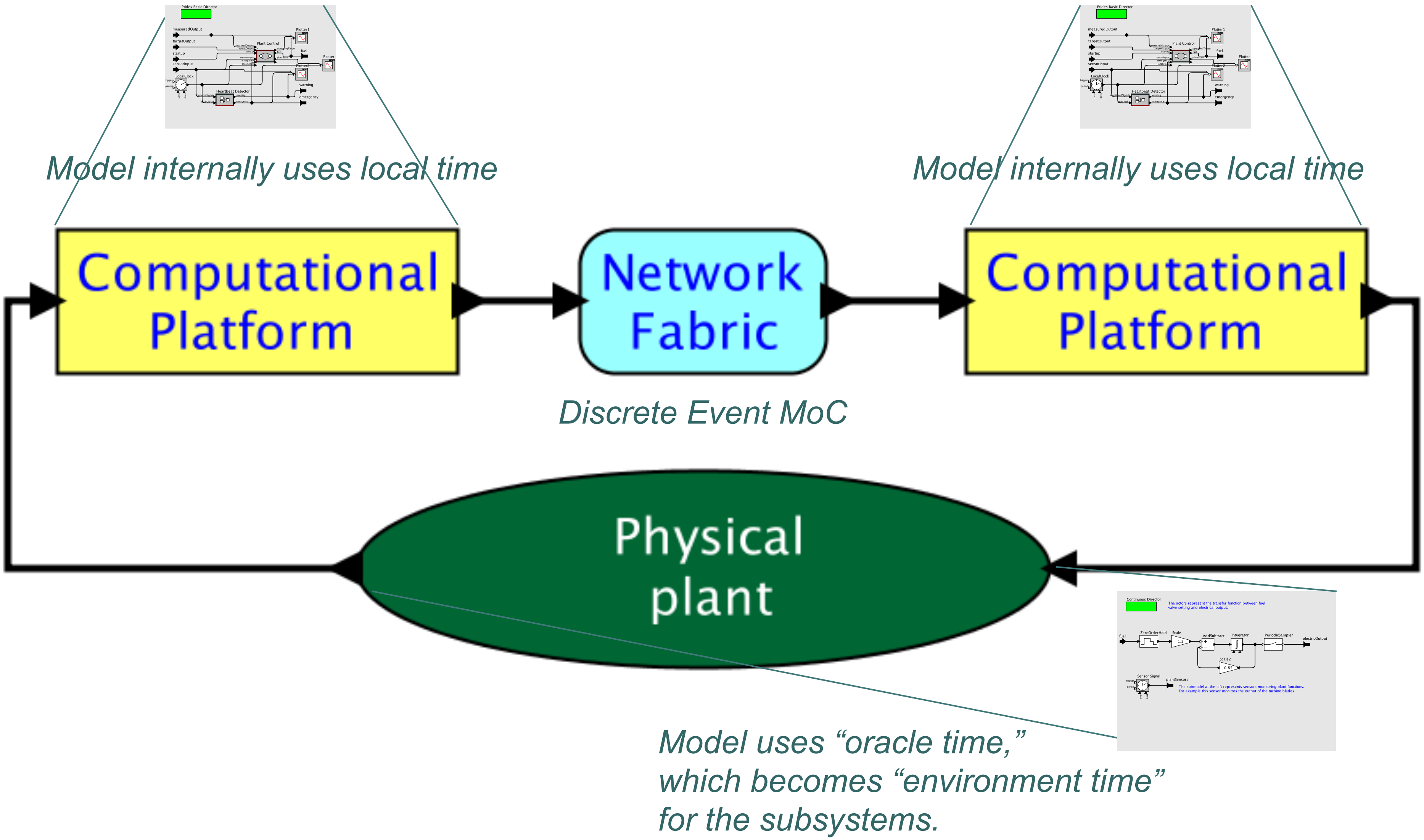
But computational platforms have no access to *t.*
Instead, local measurements of time are used.



A superdense Newtonian notion of time becomes environment time

$$\dot{\theta}(t) = \dot{\theta}(0) + \frac{1}{I}\int_0^t \mathbf{T}(\tau)\,d\tau$$

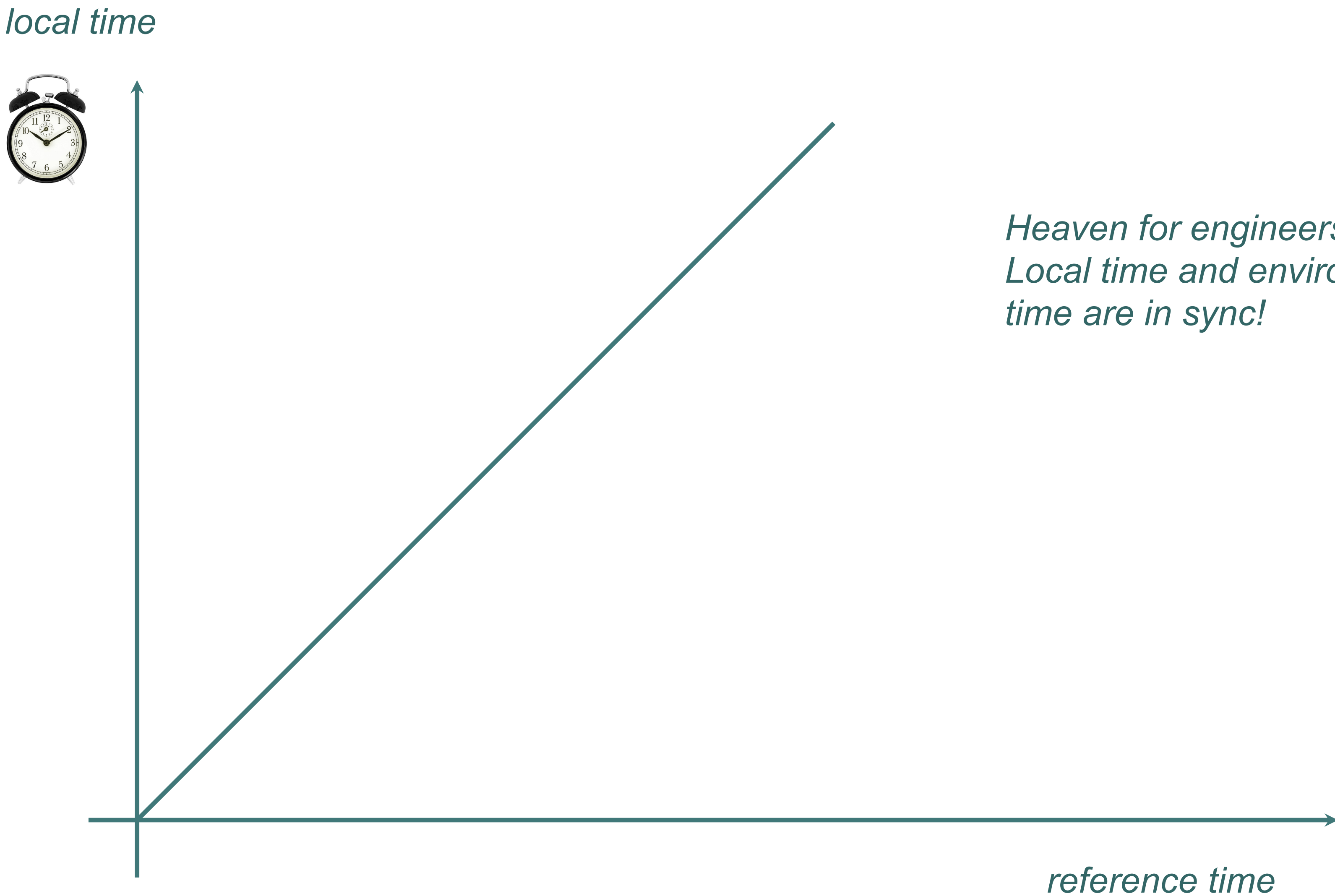
Local time within a hierarchy can advance at different rates.



Clocks drift

- Fabrication tolerance
- Aging
- Temperature
- Humidity
- Vibrations
- Quality of the quartz.
- Clock drifts measured in “parts per million” or ppm
  1 ppm corresponds to a deviation of 1µs every second

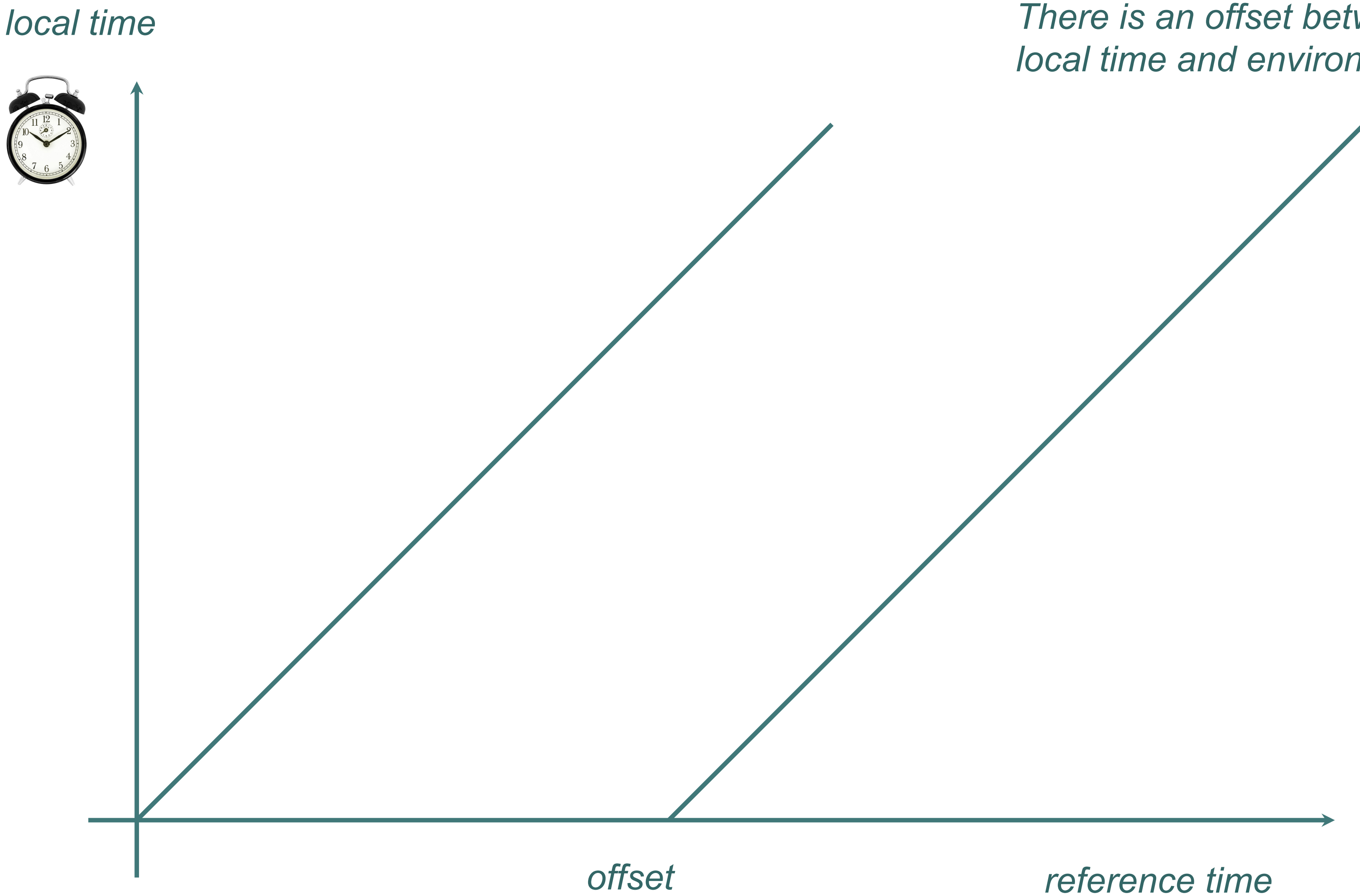
MultiForm Time in Ptolemy



Heaven for engineers.
Local time and environment time are in sync!

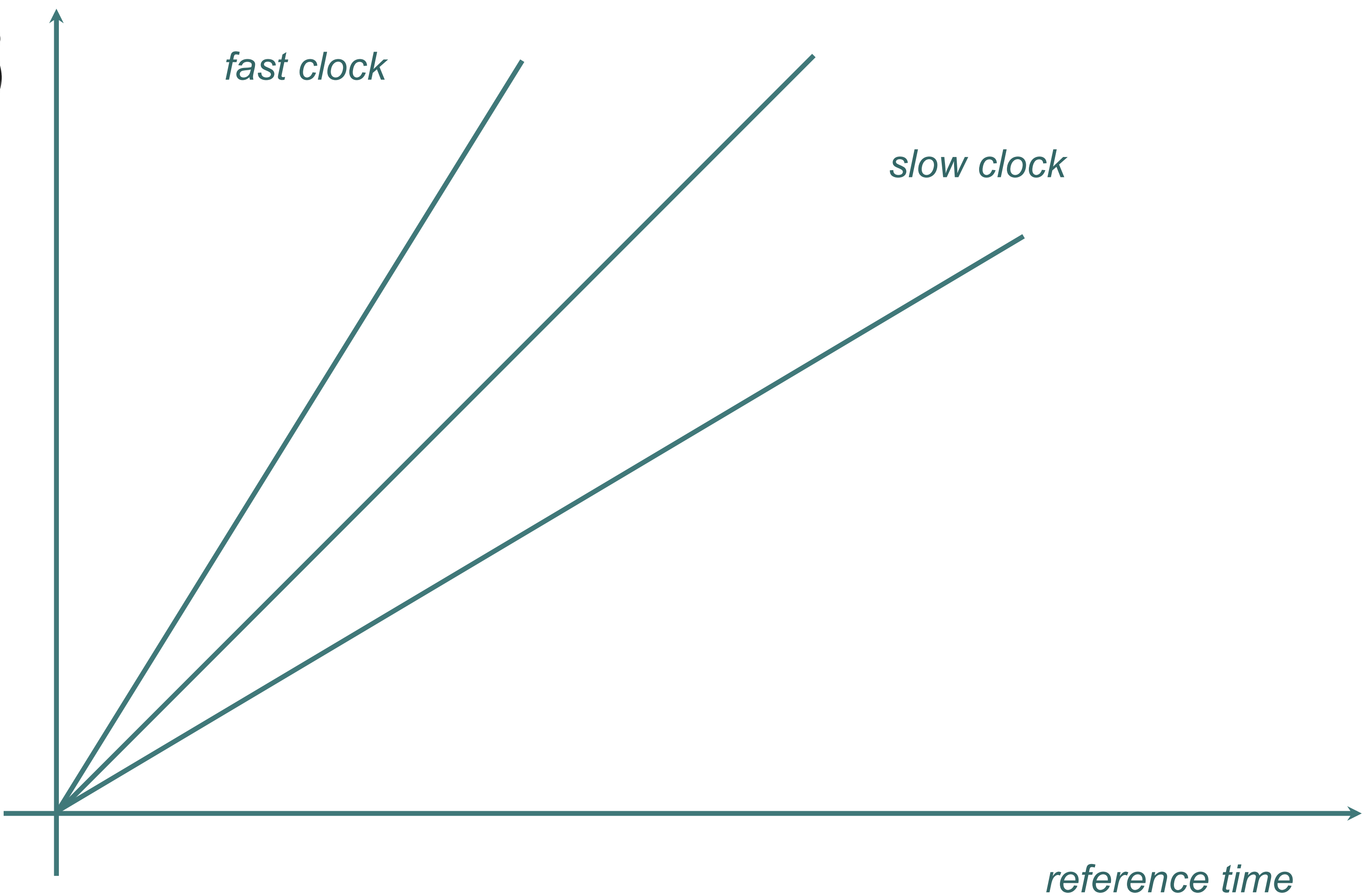
Multiform Time in the Real World

Reality:
There is an offset between local time and environment time



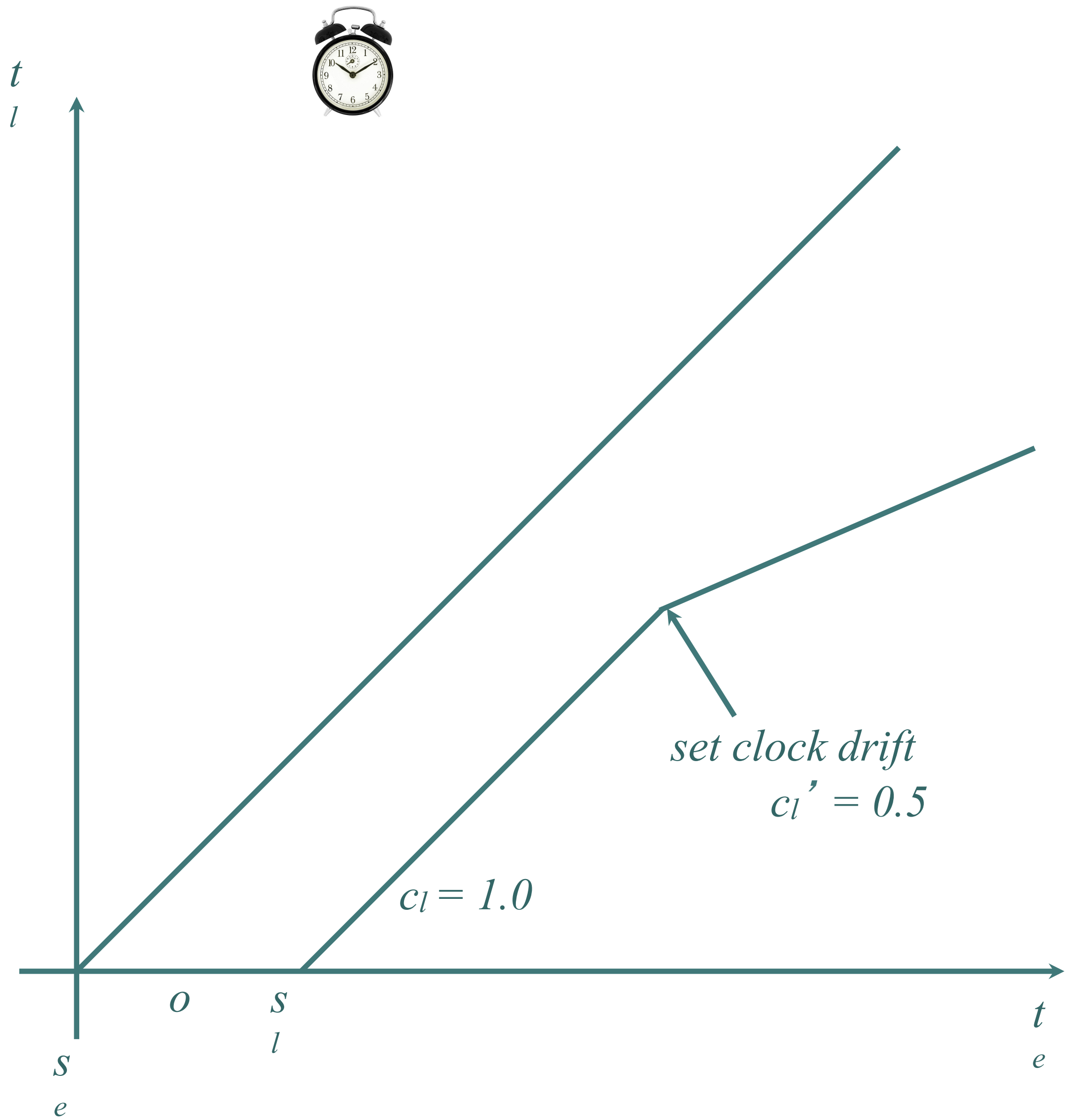
Multiform Time in Ptolemy

More real: clocks drift



Multiform Time in Ptolemy

*Even more real: clock drift **changes**!*



environment time:
t_e

start time:
s_e, s_l

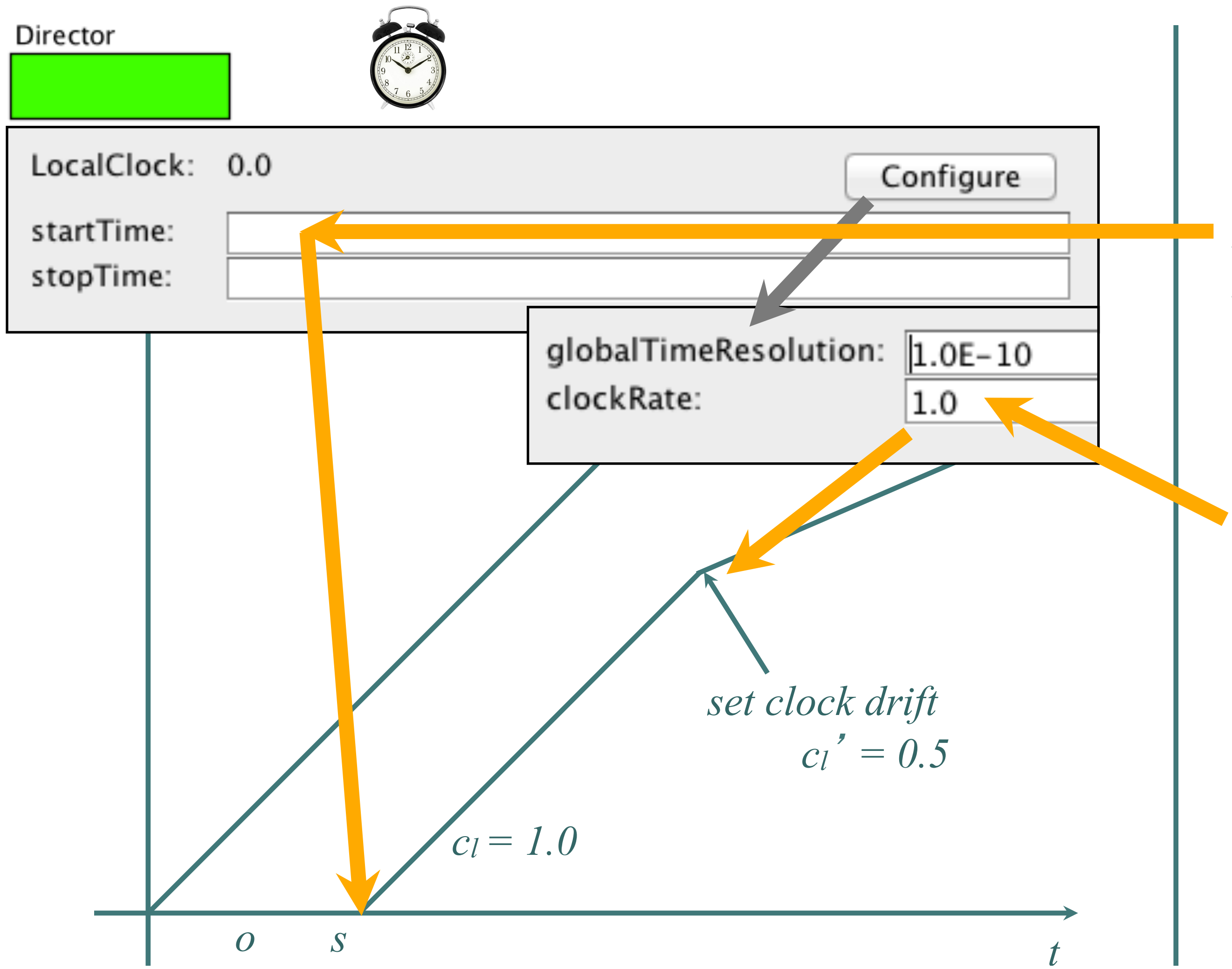
offset:
$o = s_e - s_l$

clock rate:
c_l

local time:
$t_l = (t_e - o) \times c_l$

Multiform Time in Ptolemy

Ptolemy II provides a hierarchy of local clocks



environment time:
t_e

start time:
s_e, s_l

offset:
$o = s_e - s_l$

clock rate:
c_l

local time:
$t_l = (t_e - o) \times c_l$

This can be used, for example, to accurately model time synchronization protocols.

Multiform Time is Intrinsic!

Time

*Time in **physical** laws, mathematical, continuous*

*Time in **digital** systems*
Circuits, discrete clocks, generating well defined periodic signals

Physical

Measured

Relativistic

Newtonian

TAI

Master Clock

Synchronized Clock

Microprocessor Clock

Clock synchronization

NTP
PTP, IEEE 1588
GPS

Source: Patricia Derler and John Eidson

Other Questions about Time:

1. Precision
   - In floating-point formats,
     precision degrades as magnitude increases

2. Clear Semantics of Simultaneity
   - Requires precise addition and subtraction, e.g.

     $$(a + b) + c = a + (b + c).$$

     Floating-point numbers don't have this property.

Floating point numbers are a poor choice for modeling time!

Conclusions

- Modeling time as a simple continuum is not adequate.
  - Superdense time offers clean semantics for instantaneous events.
- Homogeneous time advancing uniformly is not adequate.
  - Hierarchical multiform time enables accurate and practical models of heterogeneous distributed systems.
- Floating point numbers for time are not adequate.
  - A model with invariant precision and precise addition and subtraction is.