Fundamental Limits of Cyber-Physical and Hybrid System Modeling

*Invited Talk*

The 3nd International Workshop on Symbolic and Numerical Methods for Reachability Analysis (SNR)
a satellite event of ETAPS 17, Uppsala, Sweden

Edward A. Lee

UC Berkeley

April 22, 2017
Fundamental Limits of Cyber-Physical Systems Modeling

EDWARD A. LEE, EECS Department, UC Berkeley

This article examines the role of modeling in the engineering of cyber-physical systems. It argues that the role that models play in engineering is different from the role they play in science, and that this difference should direct us to use a different class of models, where simplicity and clarity of semantics dominate over accuracy and detail. I argue that determinism in models used for engineering is a valuable property and should be preserved whenever possible, regardless of whether the system being modeled is deterministic. I then identify three classes of fundamental limits on modeling, specifically chaotic behavior, the inability of computers to numerically handle a continuum, and the incompleteness of determinism. The last of these has profound consequences.

CCS Concepts: • Theory of computation → Timed and hybrid models; • Computing methodologies → Modeling methodologies; • Software and its engineering → Domain specific languages

Additional Key Words and Phrases: Chaos, continuums, completeness

ACM Reference Format:
DOI: http://dx.doi.org/10.1145/2912149
In this example, the modeling framework is calculus and Newton’s laws.

**The model**

\[
x(t) = x(0) + \int_0^t v(\tau) d\tau \\
v(t) = v(0) + \frac{1}{m} \int_0^t F(\tau) d\tau
\]

**The target** (the thing being modeled)

**Fidelity** is how well the model and its target match
A model is deterministic if, given the initial state and the inputs, the model defines exactly one behavior.

Deterministic models have proved extremely valuable in the past:

- Differential equations
- Synchronous digital logic
- Instruction-set architectures
- Single-threaded imperative programs
A model is deterministic if, given the initial state and the inputs, the model defines exactly one behavior.

Deterministic models have proved extremely valuable in the past:

- Differential equations
- Synchronous digital logic
- Instruction-set architectures
- Single-threaded imperative programs
A model is deterministic if, given the initial state and the inputs, the model defines exactly one behavior.

Advantages:
- Enables testing
- Enables fault detection
- Makes simulation more effective
- Improves understanding
- Aligns with most of physics
Newton’s Cradle – The Model

Image by Dominique Toussaint, GNU Free Documentation License, Version 1.2 or later.
Newton’s Cradle – A Physical Realization
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.
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*. 

- 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*.
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*. 
The question is not whether deterministic models can describe the behavior of cyber-physical systems (with high fidelity).

The question is whether we can build cyber-physical systems whose behavior matches that of a deterministic model (with high probability).
The question is not whether deterministic models can describe the behavior of cyber-physical systems (with high fidelity).

The question is whether we can build cyber-physical systems whose behavior matches that of a deterministic model (with high probability).
Cyber Models

Physical System:

Cyber Model:

```java
/**
 * Reset the output receivers, which are the inside receivers of
 * the output ports of the container.
 * @exception IllegalStateException 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.
Discrete modeling of collisions?  
Or continuous?

- localized plastic deformation
- viscous damping
- acoustic wave propagation

But:

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

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.
Discrete modeling of collisions?  
Or continuous?

- localized plastic deformation
- viscous damping
- acoustic wave propagation

But:

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

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.
Discrete modeling of collisions?  
Or continuous?  
- localized plastic deformation  
- viscous damping  
- acoustic wave propagation  

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

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.
Outline of This Talk:

- Complexity
- Uncertainty
- Chaos
- Discretizing the Continuum
- Determinism is Incomplete
- Discreteness is Unavoidable
Outline of This Talk:

- Complexity
- Uncertainty
- Chaos
- Discretizing the Continuum
- Determinism is Incomplete
- Discreteness is Unavoidable
“Iron wing” prototype of an Airbus A350.

Will virtual prototyping ever reach a sufficient level of fidelity for such a system?

Cf. Electronic design automation, where virtual prototyping works fine for billion-transistor chips.
Outline of This Talk:

- Complexity
- **Uncertainty**
- Chaos
- Discretizing the Continuum
- Determinism is Incomplete
- Discreteness is Unavoidable
Uncertainty

We can’t construct deterministic models of what we don’t know.

For this, nondeterminism is useful.

Bayesian probability (which is mostly due to Laplace) quantifies uncertainty.

Portrait of Reverend Thomas Bayes (1701 - 1761) that is probably not actually him.
Outline of This Talk:

- Complexity
- Uncertainty
- Chaos
- Discretizing the Continuum
- Determinism is Incomplete
- Discreteness is Unavoidable

Determinism does not imply predictability.

(Lorenz, 1963; Thiele and Kumar, 2015)
Lorenz attractor:

\[
\begin{align*}
\dot{x}_1(t) &= \sigma(x_2(t) - x_1(t)) \\
\dot{x}_2(t) &= (\lambda - x_3(t))x_1(t) - x_2(t) \\
\dot{x}_3(t) &= x_1(t)x_2(t) - bx_3(t)
\end{align*}
\]

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

(Lorenz, 1963)
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.
Mathematical:

\[
\begin{align*}
\dot{x}_1(t) &= \sigma(x_2(t) - x_1(t)) \\
\dot{x}_2(t) &= (\lambda - x_3(t))x_1(t) - x_2(t) \\
\dot{x}_3(t) &= x_1(t)x_2(t) - bx_3(t)
\end{align*}
\]
Mathematical:

\[ \dot{x}_1(t) = \sigma(x_2(t) - x_1(t)) \]
\[ \dot{x}_2(t) = (\lambda - x_3(t))x_1(t) - x_2(t) \]
\[ \dot{x}_3(t) = x_1(t)x_2(t) - bx_3(t) \]

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

Computational:
Outline of This Talk:

- Complexity
- Uncertainty
- Chaos
- Discretizing the Continuum
- Determinism is Incomplete
- Discreteness is Unavoidable
Newton’s Second Law with Impulsive Forces

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

\[
x(t) = x(0) + \int_0^t v(\tau) \, d\tau
\]

\[
v(t) = v(0) + \frac{1}{m} \int_0^t F(\tau) \, d\tau
\]

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 $\delta$ is the Dirac delta function.
Consider modeling collisions of masses in motion. Simple $F = ma$ model:

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

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 $\delta$ is the Dirac delta function.
NOTE: The output $v$ depends immediately on the input $F_i$, if it is present.
NOTE: The output $v$ depends immediately on the input $F_i$, if it is present.
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.
Bouncing Ball Model

- e: 0.8 coefficient of restitution
- m: 1.0 mass

- express the impulsive force as: $-(1+e)v^*m$
The velocity and position of the ball lie in a continuum. The surface is modeled as discrete.
The velocity and position of the ball lie in a continuum. The surface is modeled as discrete.
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.
Regimes of Validity of a Model

All models are wrong, some are useful. (Box and Draper, 1987)

Modal models (switched 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

All models are wrong, some are useful. (Box and Draper, 1987)

Modal models (switched 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

Switch out of the free-fall mode when that model is no longer valid.
In theory, without the modal model, the ideal model prescribes an infinite number of bounces in finite time:
The following model exhibits Zeno behavior if the Collatz Conjecture is false, and otherwise does not:

Collatz Conjecture: For any natural number $n \geq 1$, if $n$ is even, divide it by 2; if $n$ is odd multiply it by 3 and add 1. Repeat the process indefinitely. The conjecture is that no matter what number you start with, you will always eventually reach 1. (due to Ben Lickly)
Outline of This Talk:

- Complexity
- Uncertainty
- Chaos
- Discretizing the Continuum
- Determinism is Incomplete
- Discreteness is Unavoidable
Discrete and continuous, cyber and physical.

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

Even purely physical models illustrate the subtleties.
Any set of deterministic models rich enough to encompass Newton’s laws plus discrete transitions is incomplete.
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:

\[
\begin{align*}
v_1' &= \frac{v_1(m_1 - m_2) + 2m_2v_2}{m_1 + m_2} \\
v_2' &= \frac{v_2(m_2 - m_1) + 2m_1v_1}{m_1 + m_2}.
\end{align*}
\]

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.
Simultaneous, Noncausal 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.
One solution: nondeterministic interleaving of the collisions:

At superdense time \((\tau, 3)\), handle the new collision.
Simultaneous, Noncausal Collisions

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 of the collisions yields the right result (for any choice of interleaving), but only if the masses are the same!
If the masses are different, the behavior depends on which collision is handled first!

![Positions of Balls](a)

![Positions of Balls](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.
Arbitrary interleaving and nondeterministic results appear to be defensible on physical grounds.
Let $\tau$ be the time between collisions. Consider a sequence of models for $\tau > 0$ where $\tau \rightarrow 0$. Sequence of models is Cauchy. Then consider $\tau < 0$ and $\tau \rightarrow 0$. Sequence is again Cauchy.

Every model in each 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.
Let $\tau$ be the time between collisions. Consider a sequence of models for $\tau > 0$ where $\tau \to 0$. Sequence of models is Cauchy. Then consider $\tau < 0$ and $\tau \to 0$. Sequence is again Cauchy.

Every model in each 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.
Outline of This Talk:

- Complexity
- Uncertainty
- Chaos
- Discretizing the Continuum
- Determinism is Incomplete
- Discreteness is Unavoidable

Deterministic models may become nondeterministic at the limits when mixing discrete and continuous behaviors.
Outline of This Talk:

- Complexity
- Uncertainty
- Chaos
- Discretizing the Continuum
- Determinism is Incomplete
- Discreteness is Unavoidable
Continuous Models of Collisions

For the ball collision example, we could defensibly reject discrete models and model the balls as squishy, springy objects. The resulting model is chaotic:

But in general, discreteness cannot be avoided without also rejecting causality.
For the ball collision example, we could defensibly reject discrete models and model the balls as squishy, springy objects. The resulting model is chaotic:

But in general, discreteness cannot be avoided without also rejecting causality.
A flyback diode is a commonly used circuit that prevents arcing when disconnecting an inductive load (like a motor) from a power source.
A flyback diode is a commonly used circuit that prevents arcing when disconnecting an inductive load (like a motor) from a power source.

If the diode is reverse biased,

\[ j(t) = \frac{1}{L} \int_0^t w(\tau) d\tau, \]
A flyback diode is a commonly used circuit that prevents arcing when disconnecting an inductive load (like a motor) from a power source.

If the diode is reverse biased,

$$ j(t) = \frac{1}{L} \int_{0}^{t} w(\tau) d\tau, $$

Hence, for the load component, $w$ is an input and $j$ is an output. The environment cannot arbitrarily set the current $j$ independent of the history.
A flyback diode is a commonly used circuit that prevents arcing when disconnecting an inductive load (like a motor) from a power source.

If the diode is reverse biased,

\[ j(t) = \frac{1}{L} \int_0^t w(\tau) d\tau, \]

Moreover, the output \( j \) does not depend immediately on the input \( w \) (there is no direct feedthrough).
Discreteness is Unavoidable

A flyback diode is a commonly used circuit that prevents arcing when disconnecting an inductive load (like a motor) from a power source.

If the switch is closed, by Ohm’s law,

\[ w(t) = u(t) - j(t)R \]
Discreteness is Unavoidable

A flyback diode is a commonly used circuit that prevents arcing when disconnecting an inductive load (like a motor) from a power source.

If the switch is closed, by Ohm’s law,

\[ w(t) = u(t) - j(t)R \]

so we can consider \( j \) to be the input and \( w \) to be the output. In this case, there is direct feedthrough.
A flyback diode is a commonly used circuit that prevents arcing when disconnecting an inductive load (like a motor) from a power source.

Hence, with the switch closed and the diode reverse biased, we have a constructive model with causality as shown below:
A flyback diode is a commonly used circuit that prevents arcing when disconnecting an inductive load (like a motor) from a power source.

Hence, with the switch closed and the diode reverse biased, we have a constructive model with causality as shown below:

This model is constructive.
A flyback diode is a commonly used circuit that prevents arcing when disconnecting an inductive load (like a motor) from a power source.

When the switch is opened, the current $j$ is forced to zero and the diode becomes forward biased.
A flyback diode is a commonly used circuit that prevents arcing when disconnecting an inductive load (like a motor) from a power source.

When the switch is opened, the current $j$ is forced to zero and the diode becomes forward biased. In this case, $j$ has to be the output of the controller, not the input, and its output $j$ does not depend on the input $w$ (there is no direct feedthrough).
A flyback diode is a commonly used circuit that prevents arcing when disconnecting an inductive load (like a motor) from a power source.

The load component also reverses causality, where $j$ becomes the input and $w$ becomes the output, because $w$ equals the voltage drop of a forward-biased diode.
A flyback diode is a commonly used circuit that prevents arcing when disconnecting an inductive load (like a motor) from a power source.

Hence, with the switch opened and the diode forward biased, we have a constructive model with causality as shown below:
A flyback diode is a commonly used circuit that prevents arcing when disconnecting an inductive load (like a motor) from a power source.

When the switch goes from closed to open, the causality and direct feedthrough properties of the two components reverse.
Discreteness is Unavoidable

A flyback diode is a commonly used circuit that prevents arcing when disconnecting an inductive load (like a motor) from a power source.

When the switch goes from closed to open, the causality and direct feedthrough properties of the two components reverse.

There is no logic that can transition from $A$ causes $B$ to $B$ causes $A$ smoothly without passing through non-constructive models.
A **flyback diode** is a commonly used circuit that prevents arcing when disconnecting an inductive load (like a motor) from a power source.

When the switch goes from closed to open, the causality and direct feedthrough properties of the two components reverse.

There is no logic that can transition from \( A \) causes \( B \) to \( B \) causes \( A \) smoothly without passing through non-constructive models.

**Discreteness is unavoidable!**
Concluding Remarks

- Scientists and engineers use models differently.
  - Deterministic models are useful.
  - Chaotic deterministic models have limited predictive power.
  - All models have a limited regime of validity.
  - Modal models makes these regimes explicit.
  - Discrete models may exhibit Zeno conditions.
  - Detecting Zeno conditions is hard.
  - Determinism is incomplete.
  - Discreteness is unavoidable.
Scientists and engineers use models differently.

- Deterministic models are useful.
- Chaotic deterministic models have limited predictive power.
- All models have a limited regime of validity.
- Modal models make these regimes explicit.
- Discrete models may exhibit Zeno conditions.
- Detecting Zeno conditions is hard.
- Determinism is incomplete.
- Discreteness is unavoidable.
Scientists and engineers use models differently.

Deterministic models are useful.

Chaotic deterministic models have limited predictive power.

All models have a limited regime of validity.

Modal models makes these regimes explicit.

Discrete models may exhibit Zeno conditions.

Detecting Zeno conditions is hard.

Determinism is incomplete.

Discreteness is unavoidable.
Scientists and engineers use models differently.

Deterministic models are useful.

Chaotic deterministic models have limited predictive power.

All models have a limited regime of validity.

Modal models makes these regimes explicit.

Discrete models may exhibit Zeno conditions.

Detecting Zeno conditions is hard.

Determinism is incomplete.

Discreteness is unavoidable.
Concluding Remarks

- Scientists and engineers use models differently.
- Deterministic models are useful.
- Chaotic deterministic models have limited predictive power.
- All models have a limited regime of validity.
- Modal models make these regimes explicit.
  - Discrete models may exhibit Zeno conditions.
  - Detecting Zeno conditions is hard.
  - Determinism is incomplete.
  - Discreteness is unavoidable.
Concluding Remarks

- Scientists and engineers use models differently.
- Deterministic models are useful.
- Chaotic deterministic models have limited predictive power.
- All models have a limited regime of validity.
- Modal models makes these regimes explicit.
- Discrete models may exhibit Zeno conditions.
  - Detecting Zeno conditions is hard.
  - Determinism is incomplete.
  - Discreteness is unavoidable.
Concluding Remarks

- Scientists and engineers use models differently.
- Deterministic models are useful.
- Chaotic deterministic models have limited predictive power.
- All models have a limited regime of validity.
- Modal models make these regimes explicit.
- Discrete models may exhibit Zeno conditions.
- Detecting Zeno conditions is hard.
  - Determinism is incomplete.
  - Discreteness is unavoidable.
Scientists and engineers use models differently.

- Deterministic models are useful.
- Chaotic deterministic models have limited predictive power.
- All models have a limited regime of validity.
- Modal models makes these regimes explicit.
- Discrete models may exhibit Zeno conditions.
- Detecting Zeno conditions is hard.
- Determinism is incomplete.

Discreteness is unavoidable.
Scientists and engineers use models differently.

- Deterministic models are useful.
- Chaotic deterministic models have limited predictive power.
- All models have a limited regime of validity.
- Modal models makes these regimes explicit.
- Discrete models may exhibit Zeno conditions.
- Detecting Zeno conditions is hard.
- Determinism is incomplete.
- Discreteness is unavoidable.
This book explores how engineers use models. I argue that these models are not discovered preexisting truths, but rather are invented in a fundamentally human creative process.


