Constructive Collisions*

Edward A. Lee

EECS Department, UC Berkeley, Berkeley, CA, USA
eal@eecs.berkeley.edu

Abstract. This paper studies the semantics of models for discrete physical phenomena such as rigid body collisions. The paper combines generalized functions (specifically the Dirac delta function), superdense time, modal models, and constructive semantics to get a rich, flexible, efficient, and rigorous approach to modeling such systems. It shows that many physical scenarios that have been problematic for modeling techniques manifest as nonconstructive models, and that constructive versions of some of the models properly reflect uncertainty in the behavior of the physical systems that plausibly arise from the principles of quantum mechanics. The paper argues that these modeling difficulties are not reasonably solved by more detailed continuous models of the underlying physical phenomena. Such more detailed models simply shift the uncertainty to other aspects of the model. Since such detailed models come with a high computational cost, there is little justification in using them unless the goal of modeling is specifically to understand these more detailed physical processes. An implementation of these methods in the Ptolemy II modeling and simulation environment is described.

1 The Problem

Many physical phenomena are naturally modeled as being discrete rather than continuous. Modeling and simulating combinations of discrete and continuous dynamics, however, are challenging. Collisions of rigid objects and friction between moving objects are classic examples. Diodes and switches in electrical circuits present similar problems. All known solutions have significant limitations.

The difficulties stem from a number of sources. First, discontinuities make signals non-differentiable, which complicates simulation and analysis. Second, discrete phenomena can cause **chattering** around the discontinuity, where the solution repeatedly bounces across a discrete boundary. Third, discrete models more easily lead to Zeno conditions than continuous models, where an infinite number of events occur in a finite time. Finally, and perhaps most importantly, physical phenomena that are most naturally modeled as discrete are among the most poorly behaved and least understood. They frequently exhibit intrinsic nondeterminism and chaotic behaviors.

* This work was supported in part by the iCyPhy Research Center (Industrial Cyber-Physical Systems, supported by IBM and United Technologies), and the Center for Hybrid and Embedded Software Systems (CHESS) at UC Berkeley (supported by the National Science Foundation, NSF awards #0720882 (CSR-EHS: PRET), #1035672 (CPS: Medium: Ptides), and #0931843 (ActionWebs), the Naval Research Laboratory (NRL #N0013-12-1-G015), and the following companies: Bosch, National Instruments, and Toyota).

S. Bensalem et al. (Eds.): FPS 2014 (Sifakis Festschrift), LNCS 8415, pp. 161–176, 2014.

These sources of difficulty are worth separating. For example, it is not appropriate to condemn a model because it fails to deterministically model an intrinsically nondeterministic physical phenomenon. Nor is it fair to condemn a model for exhibiting Zeno behavior if the Zeno condition manifests outside the regime of parameters for which the model is suited.

Stewart [25] gives an excellent overview of approaches that have been used towards solving these problems for collisions and friction between macroscopic physical objects. In this regime, a solution that admits discrete behaviors can use generalized functions, most commonly the Dirac delta function, Lebesgue integration, measure theory, and differential inclusions. Stewarts argues for embracing discrete behaviors in models, and shows that a well-known paradox in the study of rigid body known as the Painlevé paradox can be resolved by admitting impulsive forces into the model.

A different (and more common) approach is to dispense with discrete models and insist on detailed modeling of the continuous dynamics. Collisions between rigid objects, for example, involve localized plastic deformation, viscous damping in the material, and acoustic wave propagation. Much experimental and theoretical work has been done to refine models of such phenomena, leading to considerable insight into the underlying physical phenomena. We contend, however, that such detailed modeling rarely helps in developing insight about macroscopic system behavior. So when the goal is, for example, to design robotic machinery, it is better to use simpler, more abstract models.

State-of-the-art design and simulation tools, however, do not support simpler models with discrete behaviors well. Modelica [26], for example, is a widely used language with well-supported libraries of models for a large variety of physical systems. Otter, et al. in [23] state that "at the moment, it is not possible to implement the solution with impulses ... in a generic way in Modelica." They offer continuous approximations as an alternative, categorizing three approaches for collisions: impulsive, spring-damper ignoring contact area, and spring-damper including contact area. They describe a library in Modelica that uses the latter two approaches.

Continuous models may indeed more accurately represent the physics, but they come at the price of greatly increased simulation cost and, perhaps more importantly, greatly increased modeling detail. The increased simulation cost is a consequence of the stiffness of the resulting differential equations. The increased modeling detail requires designers to specify much more detail about materials and systems than may be reasonable, particularly at early stages of design. Moreover, such detailed models may just shift the uncertainty from the modeling approximations to the determination of parameters. Is a robot designer able to characterize acoustic propagation in steel for a particular shape of robot arm in a particular range of temperatures and as the product ages? Probably not. So a detailed simulation model based on continuous physical processes may not be any more trustworthy than a much less detailed model.

In contrast, models that are created for the purpose of providing computer animations, like those described in Erleben et al. [9], are closer to what we need for understanding system dynamics. Computer animation has the very practical driving force that it must exhibit some behavior in reasonable time, so simulation efficiency is important.

The goal of this paper is improve the trustworthiness of less detailed, more abstract models. The approach is to put the semantics of the models on a solid foundation. If

the meaning of a model is absolutely clear, it is much easier to tell whether the model is faithful to the physical system it is modeling, and it is much easier to draw trusted conclusions from executions of the model.

To provide a solid foundation for abstract models, this paper embraces discrete phenomena modeled using generalized functions, and uses an extended model of time known as **superdense time** to cleanly mix discrete and continuous dynamics. In addition, the technique in this paper supports **modal models**, where a multiplicity of distinct abstract models, each with a well-defined regime of applicability, are combined to model the same system (as in **hybrid systems** [20,1]). Finally, the modeling framework is given a constructive fixed-point semantics [5], like that in synchronous-reactive languages [3]. We conjecture that nonconstructive models are suspect on physical grounds, and show that a number of well-known problematic scenarios with modeling discrete physical phenomena result in nonconstructive models. The techniques in this paper have been implemented as a Ptolemy II simulation tool [24], and the models displayed in this paper are all available online at `http://ptolemy.org/constructive/models`.

This paper is a shortened version of a technical report [13] that includes many more examples and more detailed analysis.

2 Time

Time is central to our approach to modeling. We require a model of time that combines a time continuum, over which physical dynamics can evolve, and discrete events, modeling abrupt changes in state of the system. In this section, we review the superdense model of time, the notion of discreteness, and the notion of piecewise continuity, which is essential for our models to work well with practical ordinary differential equation (ODE) solvers.

2.1 Superdense Time

We use a model of time known as **superdense time** [20,17]. A superdense time value is a pair (t,n), called a **time stamp**, where t is the **model time** and n is an **index** (also called a **microstep**). The model time represents the time at which some event occurs, and the microstep represents the sequencing of events that occur at the same model time. Two time stamps (t,n_1) and (t,n_2) can be interpreted as being **simultaneous** (in a weak sense) even if $n_1 \neq n_2$. Strong simultaneity requires the time stamps to be equal (both in model time and microstep).

To understand the role of the microstep, consider Newton's cradle, a toy with five steel balls suspended by strings. If you lift the first ball and release it, it strikes the second ball, which does not move. Instead, the fifth ball reacts by rising. Consider the momentum p of the second ball as a function of time. The second ball does not move, so its momentum must be everywhere zero. But the momentum of the first ball is somehow transferred to the fifth ball, passing through the second ball. So the momentum cannot be always zero.

Let $\mathbb{R}$ represent the real numbers. Let $p: \mathbb{R} \to \mathbb{R}$ be a function that represents the momentum of this second ball, and let τ be the time of the collision. Then

$$p(t) = \begin{cases} P & \text{if } t = \tau \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

for some constant P and for all $t \in \mathbb{R}$. Before and after the instant of time τ, the momentum of the ball is zero, but at time τ, it is not zero. Momentum is proportional to velocity, so

$$p(t) = Mv(t),$$

where M is the mass of the ball. Hence, combining with (1),

$$v(t) = \begin{cases} P/M & \text{if } t = \tau \\ 0 & \text{otherwise.} \end{cases} \tag{2}$$

The position of a mass is the integral of its velocity,

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

where $x(0)$ is the initial position. The integral of the function given by (2) is zero at all t, so the ball does not move, despite having a non-zero momentum at an instant.

The above physical model mostly works to describes the physics, but it has two flaws. First, it violates the basic physical principle of conservation of momentum. At the time of the collision, all three middle balls will simultaneously have non-zero momentum, so seemingly, aggregate momentum has magically increased.

Second, the model cannot be directly converted into a discrete representation (see Section 2.3 below). A discrete representation of a signal is a sequence of values that are ordered in time. Any such representation of the momentum in (1) or velocity in (2) is ambiguous. If the sequence does not include the value at the time of the collision, then the representation does not capture the fact that momentum is transferred through the ball. If the representation does include the value at the time of the collision, then the representation is indistinguishable from a representation of a signal that has a non-zero momentum over some interval of time, and therefore models a ball that does move. In such a discrete representation, there is no semantic distinction between an instantaneous event and a rapidly varying continuous event.

Superdense time solves both problems. Specifically, the momentum of the second ball can be unambiguously represented by a sequence of samples where $p(\tau, 0) = 0$, $p(\tau, 1) = P$, and $p(\tau, 2) = 0$, where τ is the time of the collision. The third ball has non-zero momentum only at superdense time $(\tau, 2)$. At the time of the collision, each ball first has zero momentum, then non-zero, then zero again, all in an instant. The event of having non-zero momentum is weakly simultaneous for all three middle balls, but not strongly simultaneous. Momentum is conserved, and the model is unambiguously discrete.

One could argue that the physical system is not actually discrete. Even well-made steel balls will compress, so the collision is actually a continuous process, not a discrete event. This may be true, but when building models, we do not want the modeling

formalism to force us to construct models that are more detailed than is appropriate. Such a model of Newton's cradle would be far more sophisticated, and the resulting non-linear dynamics would be far more difficult to analyze. The fidelity of the model may improve, but at a steep price in understandability and analyzability. Moreover, if the properties of the material and the dynamics of the collision are not well understood, the fidelity of the model may actually degrade as more detail is added.

The Newton's cradle example shows that physical processes that include instantaneous events are better modeled using functions of the form $p\colon \mathbb{R}\times\mathbb{N}\to\mathbb{R}$, where $\mathbb{N}$ represents the natural numbers, rather than the more conventional $p\colon \mathbb{R}\to\mathbb{R}$. The latter is adequate for continuous processes, but not for discrete events. At any time $t\in\mathbb{R}$, the signal p has a sequence of values, ordered by their microsteps. This signal cannot be misinterpreted as a rapidly varying continuous signal.

Superdense time is ordered lexicographically (like a dictionary), which means that $(t_1,n_1)<(t_2,n_2)$ if either $t_1<t_2$, or $t_1=t_2$ and $n_1<n_2$. Time stamps are a particular realization of **tags** in the tagged-signal model of [14].

2.2 Piecewise Continuity

So that we can leverage standard, well-understood numerical integration methods, we require signals to be piecewise-continuous in a specific technical sense. Consider a real-valued superdense-time signal $x\colon \mathbb{R}\times\mathbb{N}\to\mathbb{R}$. At each real-time $t\in\mathbb{R}$, we require that $x(t,n)$ settle to a final value and stay there. Specifically, we require that for all $t\in\mathbb{R}$, there exist an $m\in\mathbb{N}$ such that

$$\forall n>m,\quad x(t,n)=x(t,m). \tag{3}$$

A violation of this constraint is called a **chattering Zeno** condition. Such conditions would prevent an execution from progressing beyond real time t, assuming the execution is constrained to produce all values in chronological order.

Assuming x has no chattering Zeno condition, then there is a least value of m satisfying (3). We call this value of m the **final microstep** and $x(t,m)$ the **final value** of x at t. We call $x(t,0)$ the **initial value** at time t. If $m=0$, then x has only one value at time t.

Define the **initial value function** $x_i\colon \mathbb{R}\to\mathbb{R}$ by

$$\forall\, t\in\mathbb{R},\quad x_i(t)=x(t,0).$$

Define the **final value function** $x_f\colon \mathbb{R}\to\mathbb{R}$ by

$$\forall\, t\in\mathbb{R},\quad x_f(t)=x(t,m_t),$$

where m_t is the final microstep at time t. Note that x_i and x_f are conventional continuous-time functions. A **piecewise continuous** signal is defined to be a function x of the form $x\colon \mathbb{R}\times\mathbb{N}\to\mathbb{R}$ with no chattering Zeno conditions that satisfies three requirements:

1. the initial value function x_i is continuous on the left at all $t\in\mathbb{R}$;
2. the final value function x_f is continuous on the right at all $t\in\mathbb{R}$; and
3. x has only one value at all $t\in\mathbb{R}\backslash D$, where D is a discrete subset of $\mathbb{R}$.

The last requirement is a subtle one that deserves further discussion. First, the notation $\mathbb{R}\backslash D$ refers to a set that contains all elements of the set $\mathbb{R}$ except those in the set D. D is constrained to be a discrete set, as defined below. Intuitively, D is a set of time values that can be counted in temporal order. It is easy to see that if $D = \emptyset$ (the empty set), then $x_i = x_f$, and both x_i and x_f are continuous functions. Otherwise each of these functions is piecewise continuous.

Such piecewise-continuous signals coexist nicely with standard ODE solvers. At the time of a discontinuity or discrete event, the final value signal provides the initial boundary condition for the solver. The solver then works with an ordinary continuous signal until the time of the next discontinuity or discrete event, and the solver provides the initial value of the signal at the time of that next event.

2.3 Discreteness

A set D is a **discrete set** if it is a totally ordered set (for any two elements d_1 and d_2, either $d_1 \leq d_2$ or $d_1 > d_2$), and there exists a one-to-one function $f\colon D \to \mathbb{N}$ that is **order preserving**. Order preserving simply means that for all $d_1, d_2 \in D$ where $d_1 \leq d_2$, we have that $f(d_1) \leq f(d_2)$. The existence of such a one-to-one function ensures that we can arrange the elements of D *in temporal order*. Notice that D is a countable set, but not all countable sets are discrete. For example, the set $\mathbb{Q}$ of rational numbers is countable but not discrete. There is no such one-to-one function.

A **discrete-event** signal is a function defined on a lower subset[1] of superdense time to some value set, where the signal is non-absent only at a discrete subset of times. I.e., a discrete-event signal is absent almost everywhere, and the superdense times at which it is not absent form a discrete set. An **event** in a discrete-event signal is a time-value pair $((t,n),v)$, where (t,n) is a superdense time and v is a non-absent value. A discrete-event signal has a first event, a second event, etc., i.e. an ordered countable set of events.

The concept of piecewise continuity can be extended to discrete-event signals. Specifically, let ε represent the absence of an event. Then a discrete-event signal is a partial function

$$x\colon \mathbb{R} \times \mathbb{N} \to \{\varepsilon\} \cup U,$$

where U is some value set, where $x(t,n) = \varepsilon$ for all $(t,n) \in (\mathbb{R} \times \mathbb{N}) \setminus D$, where D is a discrete set. That is, the signal is absent almost everywhere, and is present only at a discrete subset of superdense times. A **piecewise-continuous discrete-event signal** is defined as a discrete-event signal whose initial value and final value functions always yield absent,

$$\forall t \in \mathbb{R}, \quad x_i(t) = \varepsilon, \quad x_f(t) = \varepsilon.$$

Such signals can coexist easily with numerical ODE solvers, since the signals seen by the solver, which are initial and final value signals, are simply absent. The solver ignores them.

The above definitions are used in [12]. Benveniste et al. in [4] define "discrete" differently to mean that "each instant has unique previous and next instants." This is a much weaker definition than ours here. We prefer our definition, because every event

[1] A **lower set** L is a subset of an ordered set S such that for all $x \in L$ and for all $y < x$, $y \in L$.

in a discrete-event signal has a finite number of predecessor events in the signal. This property is essential to being able to compute the events in a signal.

We define a **continuous-time signal** to be a signal whose value is not absent at any superdense time. Clearly, a continuous-time signal is not a discrete-event signal. But we can have signals that are neither continuous-time signals nor discrete-event signals. A signal that is not a discrete-event signal will need to either be represented symbolically or numerically approximated in any computation. Standard ODE solvers produce estimated **samples** of continuous-time signals. The time spacing between samples is determined by the step-size control of the solver. The samples themselves are defined on a discrete subset of superdense time.

2.4 An Alternative Model of Time

Note that an alternative model of time that can accomplish the same objectives as superdense time is studied in [4,22]. Their construction is based on nonstandard analysis [19], which, similarly to superdense time, has an infinite number of points at every real time point. These points are represented as convergent sequences, and a total order is induced over these sequences by means of a measure-theoretic construction. It has the property that every non-standard time has an immediate predecessor and an immediate successor, which the authors say provides an operational semantics. However, while an operational semantics does require the notion of a discrete step of computation, it also requires that the number of steps preceding any given step be finite. That is not automatically provided by the nonstandard semantics, and when it is provided, the solutions seem to be isomorphic with our much simpler superdense time construction. Hence, it does not appear to this author that anything is gained by going to a more complex mathematical formulation.

3 Constructive Fixed-Point Semantics

The next essential element in our rigorous modeling framework is the **constructive fixed-point semantics** [5], which defines the meaning of a model as a composition of components. The semantics we choose for hybrid systems is that of [18], which is implemented in Ptolemy II [6].

A model is assumed to be a graph of **actors**, as shown in Figure 1. An execution of the model (a simulation) will choose a discrete subset $D \subset \mathbb{R} \times \mathbb{N}$ of superdense time values at which to evaluate the model. The elements of D will be selected in order by a solver, beginning at some start time, say $(0,0)$. At each $(t,m) \in D$, the solver will find a value for each of the signals in the model, using a constructive procedure described below. For example, at the start time $(0,0)$, the solver will find the values $x(0,0), y(0,0)$, and $z(0,0)$ in Figure 1.

After finding the values of all signals at a time $(t,m) \in D$, the solver will choose the next time (t',m') at which to evaluate the model. To do this, it must first ensure that it has found the *final value* of each signal. If it has not, then it will increment only the microstep, so $(t',m') = (t,m+1)$. If it has found the final value of all signals, then it will consult an **event queue**, which contains a record of future discrete events,

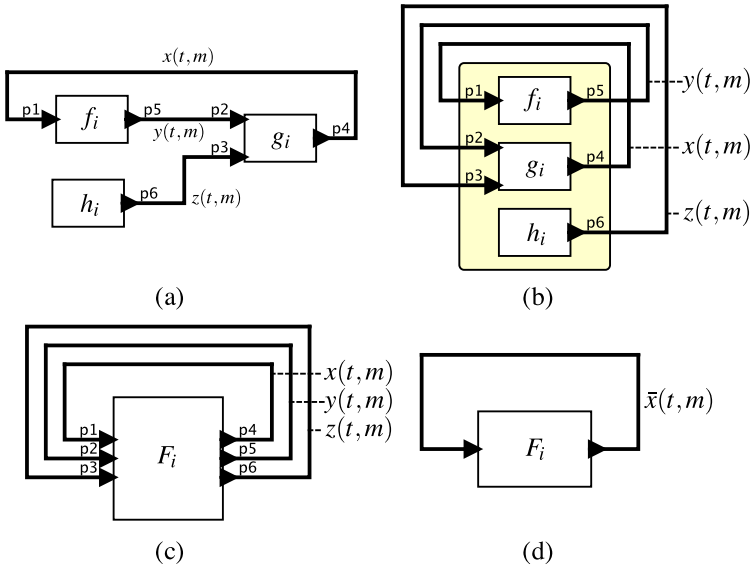

Fig. 1. A composition of actors with a constructive fixed-point semantics

and a **numerical ODE solver**, which determines a step size Δ that achieves a desired numerical accuracy. It then chooses the lesser of $(t+\Delta, 0)$ and (t_n, m), where (t_n, m) is the superdense time of the earliest event in the event queue.

Each superdense time in D is called a **tick**. The set D of ticks is discrete. The ticks can thus indexed by the natural numbers, so that

$$D = \{\tau_0, \tau_1, \cdots, \tau_i, \cdots\},$$

where $i < j \in \mathbb{N}$ implies that $\tau_i < \tau_j$.

At each $(t,m) \in D$, the solver needs to find the value of each signal. To support this, each actor provides a function from its input values to its output values. In brief, each function is required to be monotonic on a flat partial order and pointwise extensions of this order. In such partial orders, every chain is finite, and hence every monotonic function is continuous. The well-known Kleene fixed-point theorem [8] then provides a guarantee that a least fixed point exists and is unique, and provides a terminating procedure for finding that fixed point. For a more tutorial exposition, consult [13].

4 Collisions

We now consider the first of the families of discrete physical phenomena, collisions between rigid objects. Consider two objects with masses m_1 and m_2 colliding on a one-dimensional frictionless surface. We would like to treat the collision as an instantaneous event and are interested in determining the velocity after a collision. Newton's laws of motion imply that total momentum is conserved. If the velocities of the masses before

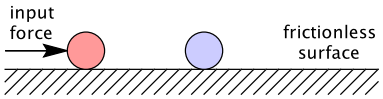

Fig. 2. Two balls on a frictionless surface

the collision are v_1 and v_2, and after the collision are v_1' and v_2', then conservation of momentum requires that

$$m_1 v_1' + m_2 v_2' = m_1 v_1 + m_2 v_2. \tag{4}$$

For notational simplicity, we leave off the dependence on time of the velocities, for now. Consider first perfectly elastic collisions, where no kinetic energy is lost. Conservation of kinetic energy requires that

$$\frac{m_1 (v_1')^2}{2} + \frac{m_2 (v_2')^2}{2} = \frac{m_1 (v_1)^2}{2} + \frac{m_2 (v_2)^2}{2}. \tag{5}$$

We have two equations and two unknowns, v_1' and v_2'. Because of the quadratic, there are two solutions to these equations. The trivial solution represents the absence of a collision, where $v_1' = v_1$ and $v_2' = v_2$. The second solution is

$$v_1' = \frac{v_1(m_1 - m_2) + 2m_2 v_2}{m_1 + m_2} \tag{6}$$

$$v_2' = \frac{v_2(m_2 - m_1) + 2m_1 v_1}{m_1 + m_2}. \tag{7}$$

Note that if $m_1 = m_2$, then the two masses simply exchange velocities.

In practice, most collisions of macroscopic physical objects lose kinetic energy. A common way to model this is to use an empirical quantity called the **coefficient of restitution**, denoted e and defined to be the relative speed after a collision divided by the relative speed before the collision. Using such a coefficient, the velocities after the collision are given by [2]

$$v_1' = \frac{e m_2 (v_2 - v_1) + m_1 v_1 + m_2 v_2}{m_1 + m_2} \tag{8}$$

$$v_2' = \frac{e m_1 (v_1 - v_2) + m_1 v_1 + m_2 v_2}{m_1 + m_2}. \tag{9}$$

The coefficient of restitution is determined experimentally for a particular pair of materials and must lie in the range $0 \leq e \leq 1$. Note that if $e = 1$, this reduces to elastic collision as given in (6) and (7). If $e = 0$, then momentum is still conserved, but the loss of kinetic energy is maximized. In this case, the resulting speeds of the two objects are identical. They collide and then travel together, not bouncing at all.

Note that if $m_1 = m_2$, then these equations reduce to

$$v_1' = (v_1(1-e) + v_2(1+e))/2 \tag{10}$$

$$v_2' = (v_2(1-e) + v_1(1+e))/2. \tag{11}$$

Another useful special case is where one of the masses is fixed (it cannot be moved), so the other will bounce off it. This follows from (8) and (9) if we let $v_2 = 0$ and determine the limit as $m_2 \to \infty$. In this case, we find

$$v_1' = -ev_1 \ . \tag{12}$$

4.1 Dirac Delta Functions

Stewart [25] points out that many of the difficulties in modeling collisions are a consequence of overly restricting the mathematical domains that are used. Impulsive forces, for example, can be naturally modeled using the **Dirac delta function**, a function $\delta : \mathbb{R} \to \mathbb{R}^+$ given by

$$\forall\, t \in \mathbb{R},\ t \neq 0, \quad \delta(t) = 0, \quad \text{and}$$
$$\int_{-\infty}^{\infty} \delta(\tau) d\tau = 1.$$

That is, the signal value is zero everywhere except at $t = 0$, but its integral is unity. At $t = 0$, therefore, its value cannot be finite. Any finite value would yield an integral of zero. This is indicated by $\mathbb{R}^+$ in the form of the function, $\delta : \mathbb{R} \to \mathbb{R}^+$, where $\mathbb{R}^+$ represents the **extended reals**, which includes infinity. Dirac delta functions are widely used in modeling continuous-time systems (see [16], for example), so it is important to be able to include them in simulations.

The Ptolemy II Integrator actor, used in Figure 3, directly supports Dirac delta functions. Specifically, the actor accepts a discrete-event signal at the input port labeled "impulse," and it interprets the real time of each event that arrives at that port as the time offset of the Dirac delta function and the value of the event as the weight of the Dirac delta function.

There are two reasons for providing a distinct input port for Dirac delta inputs vs. ordinary continuous-time inputs. First, the value of a Dirac impulse at the time it occurs is not a real number, so we would need some extended data type to include such weighted non-real values. But more importantly, at a superdense time (t, m), the output of the Integrator does not depend on the value of the input at the "derivative" input port, but it *does* depend on the value of the input at the "impulse" port! There is **direct feedthrough** from the impulse input to the output. The Integrator actor is both strict and nonstrict, depending on which input is being considered.

This causality distinction is essential to the soundness of our modeling approach. In fact, we will show below that many problematic modeling problems with discrete physical phenomena manifest as nonconstructive models. For example, any direct feedback from the output of the Integrator to the impulse input will result in a causality loop, and hence a nonconstructive model. Glockner [11] also advocates separately treating ordinary continuous-time signals and impulsive signals, whose models "split into the atomic and the Lebesgue part."

4.2 Modeling Collisions as Impulses

Figure 3 shows a Ptolemy II composite actor that models Newton's equations of motion, $F = ma$. The model has three parameter, the mass m of the ball, and the initial position

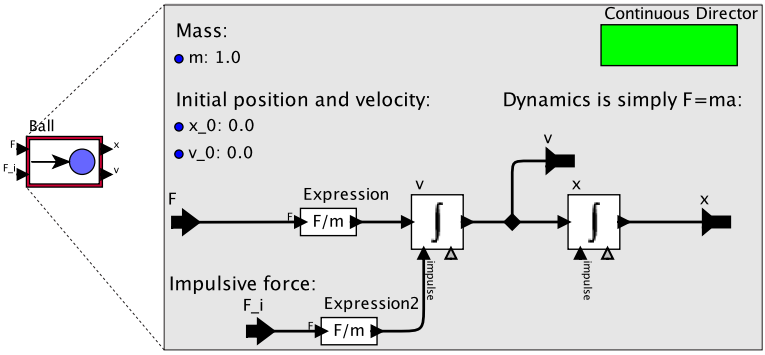

Fig. 3. A Ptolemy II model of a ball

x_0 and velocity v_0. The model uses Newton's second law to output the velocity v and position x as a function of time. There are two inputs, a real-valued force F and an impulsive force F_i. F_i is required to be a piecewise-continuous discrete-event signal, and its value represents the weight of a Dirac delta function.

A model that composes two instances of the ball model from Figure 3 is shown in Figure 4. The figure also shows a plot of the positions of two balls of diameter 1.0, where the left ball has an initial velocity of 1.0, and the right ball is standing still. After the collision, the situation is reversed. At the superdense time of the collision, the LevelCrossingDetector actor outputs an event, which enables execution of the **CalculateImpulsiveForce** composite actor. That actor is an instance of a subclass of **EnabledComposite**, which executes the inside model only when the *enable* port at the bottom has a present input with value true. The CalculateImpulsiveForce actor samples the current velocities of the balls and calculates the impulsive force that will change the velocities to those given by equations (8) and (9). The impulsive forces are then routed through a pair of MicrostepDelay actors, which apply the forces in the next microstep. Without these MicrostepDelay actors, we would have a causality loop, because the CalculateImpulsiveForce actor observes the velocities of the balls, and an impulsive force directly affects the velocities.

A collision occurs when the position of the right edge of the left ball coincides with the left edge of the right ball, and when the velocity of the left ball is greater than the velocity of the right ball. If ball 1 is on the left and ball 2 on the right, and the diameter of the left ball is d, then the collision occurs when

$$(x_1 + d \geq x_2) \wedge (v_1 > v_2).$$

However, this statement is fundamentally problematic. A collision occurs at the instant when the above predicate becomes true. First, there may be no such precise instant (suppose the balls are initially touching and we start applying a force to the left ball). Second, computational numerical methods have to approximate the continuums of time and position. In practice, to model such a collision as a discrete event, we need an error tolerance. Lower error tolerance will translate directly into increased computational cost.

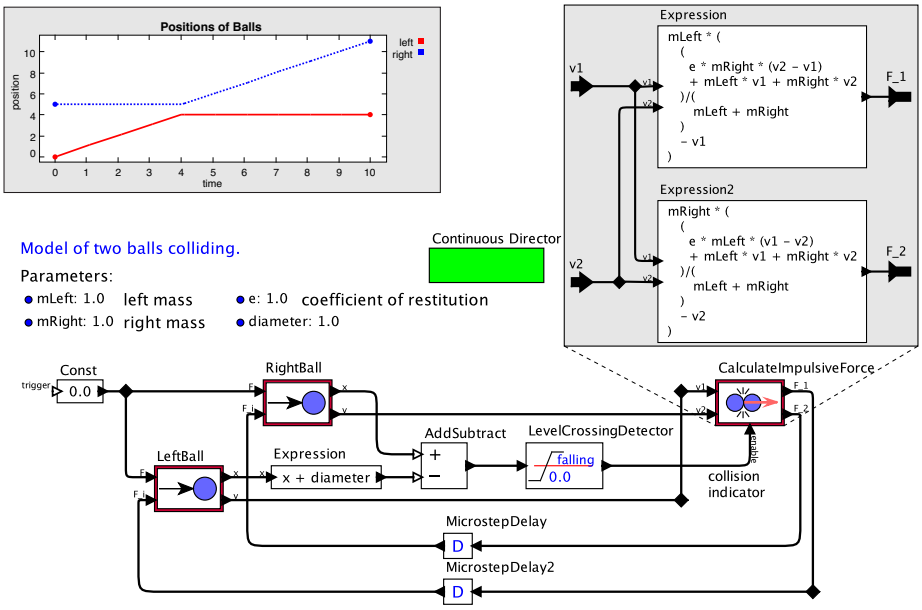

Fig. 4. A Ptolemy II model of two balls that collide

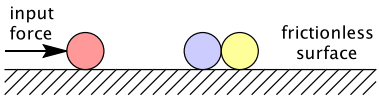

Fig. 5. Three balls on a frictionless surface

In Figure 4, the collision is detected by an actor labeled LevelCrossingDetector, which detects zero crossings of the distance between the two balls. This actor collaborates with the solver to adjust the step size of the numerical ODE solution so that the zero crossing is pinpointed with precision specified by a parameter.

Detecting collisions as zero crossings of the distance function, however, raises another difficulty. Specifically, if two balls are initially touching, the distance starts at zero. If a collision occurs, it does not cross zero. We consider this problem next.

4.3 Simultaneous Collisions

Consider the scenario shown in Figure 5, which is analogous to Newton's cradle. Two balls are initially touching, with zero distance between them, and a third ball approaches them from the left. At the instant of the collision, the left ball will transfer its momentum to the middle ball (assuming it has the same mass), which will then instantly transfer its momentum to the right ball. These two transfers occur at successive superdense time microsteps, so that the total momentum in the system is constant over time.

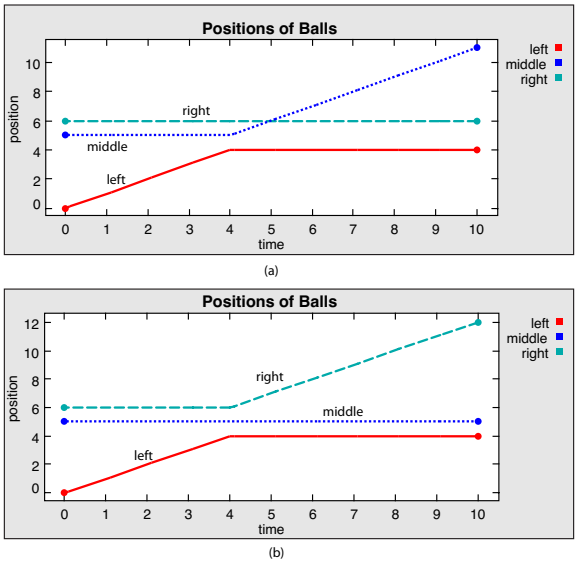

Fig. 6. (a) Second collision is not detected. (b) Second collision is detected.

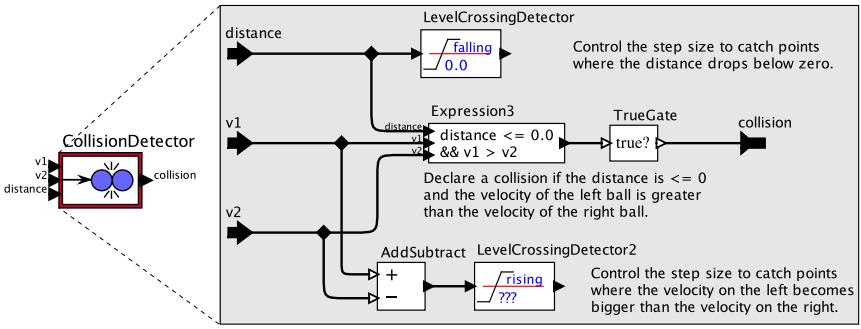

Fig. 7. A Ptolemy II model of a collision detector

The zero-crossing detection strategy in Figure 4, however, will not work for this scenario. It will fail to detect the second collision, because the distance between the middle and the right balls does not cross zero. It is initially zero, and a model like that in Figure 4 will show the left ball passing through the right ball, as shown in Figure 6(a). This difficulty is corrected by using a more sophisticated collision detection shown in Figure 7, which yields the plot in Figure 6(b). This correctly emulates Newton's cradle.

5 Discussion

Many more examples are considered in the extended version of this paper [13]. In that paper, a third scenario for the ball collisions is considered, where two balls

simultaneously collide with a stationary ball from opposite sides. This scenario is fundamentally problematic, and the Newtonian model of collisions given above has difficulty with it. A naive model superimposes the impulsive forces from the two simultaneous collisions, which cancel each other out in the middle ball. The result is a model where upon colliding, all three balls instantly stop. All the energy in the system is instantly lost.

One possible solution is to replace Newton's model with the **Poisson hypothesis**, which postulates that a collision consists of two distinct phases, a **compression phase** and a **restitution phase**. It is possible to construct a model where the two collisions have simultaneous compression phases, storing their kinetic energy as potential energy, and then, one superdense time index later, simultaneously release the potential energy as kinetic energy. Such a model would seem to solve the problem, but actually, it doesn't. There are many ways to assign kinetic energy such that both energy and momentum are conserved. In fact, such a solution simply masks a more fundamental physical problem.

An alternative solution is consider the two simultaneous collisions as being arbitrarily interleaved. That is, one occurs first, then the other. If the balls have the same mass, then it does not matter which one occurs first, and the model yields reasonable behavior. The outside balls bounce back with equal speed. However, if the balls have different masses, then the behavior depends on the order in which the collisions are handled, even though no time elapses between collisions.

In light of the Heisenberg uncertainty principle, these difficulties should not be surprising. The Heisenberg uncertainty principle states that 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. A direct expression of such simultaneous collisions results in a nonconstructive model. To get a constructive model, we have to insert microstep delays and tolerate nondeterminism. Nature, it seems, resolves nonconstructiveness with uncertainty. Chatterjee and Ruina suggest that indeed, a reasonable and practical approach to simulating such systems is to nondeterministically choose an ordering [7].

Note that doing more detailed modeling of the collisions does not solve the problem. It just shifts the uncertainty to other parts of the model. Unlike the two-ball collision, there are multiple solutions that conserve energy and momentum. We conjecture that defensible detailed models could yield the same (or more) variabilities in behaviors.

The extended paper [13] also studies Zeno conditions, using the classical bouncing-ball example. It shows that any finite precision model results in the ball eventually "tunnels" through the surface on which it is supposed to be bouncing. Again, it might seem odd to invoke quantum mechanics when considering macro phenomena such as collisions of balls. But the impulsive model we are using has infinite precision, and in physics, it is at high precisions where quantum mechanical effects become important. The solution is to avoid using models outside the their regime of applicability. Specifically, the idealized bouncing ball model with impulsive collisions becomes inappropriate when the extent of the bounce is comparable to the numerical precision of modeling tool. This is analogous to the situation in physics, where in different regimes, one might use classical Newtonian mechanics, quantum physics, or relativity, and failing to use the right models will yield misleading results.

The solution is to use **modal models** [10,15]. Such models give a multiple distinct models of the dynamics, and provide a state machine that transitions between these models. When the operating conditions exit the regime of applicability of a model of the dynamics, the state machine switches to a different model. Modal models provide an operational semantics for hybrid systems [17] that leverages superdense time.

The extended paper [13] also studies physical models of friction, which also exhibits discrete changes in behavior, combinations of friction and collisions, and electrical circuits with discrete behaviors, as in [21]. It shows that the same principles apply.

6 Conclusion

Constructive semantics gives a natural way to separate problems that can be solved with confidence from those that cannot. When, for example, the order of nearly instantaneous collisions is important, a constructive semantics forces us to either choose an order or explicitly choose nondeterminism. Building useful constructive models of combined continuous and discrete behaviors is facilitated by a superdense model of time, an explicit use of impulses (generalized functions), and modal models.

References

1. Alur, R., Courcoubetis, C., Halbwachs, N., Henzinger, T., Ho, P.-H., Nicollin, X., Olivero, A., Sifakis, J., Yovine, S.: The algorithmic analysis of hybrid systems. Theoretical Computer Science 138(1), 3–34 (1995)
2. Beer Jr., F., Johnston, E.R.: Vector equations for Engineers: Dynamics, 6th edn. McGraw Hill (1996)
3. Benveniste, A., Berry, G.: The synchronous approach to reactive and real-time systems. Proceedings of the IEEE 79(9), 1270–1282 (1991)
4. Benveniste, A., Bourke, T., Caillaud, B., Pouzet, M.: The fundamentals of hybrid systems modelers. Journal of Computer and System Sciences 78(3), 877–910 (2012)
5. Berry, G.: The Constructive Semantics of Pure Esterel. Draft version, 3rd edn. (2003)
6. Cardoso, J., Lee, E.A., Liu, J., Zheng, H.: Continuous-time models. In: Ptolemaeus, C. (ed.) System Design, Modeling, and Simulation using Ptolemy II. Ptolemy.org, Berkeley (2014)
7. Chatterjee, A., Ruina, A.: A new algebraic rigid body collision law based on impulse space considerations. Journal of Applied Mechanics 65(4), 939–951 (1998)
8. Davey, B.A., Priestly, H.A.: Introduction to Lattices and Order, 2nd edn. Cambridge University Press (2002)
9. Erleben, K., Sporring, J., Henriksen, K., Dohlmann, H.: Physics-Based Animation. Charles River Media, Inc., Hingham (2005)
10. Feng, T.H., Lee, E.A., Liu, X., Tripakis, S., Zheng, H., Zhou, Y.: Modal models. In: Ptolemaeus, C. (ed.) System Design, Modeling, and Simulation using Ptolemy II. Ptolemy.org, Berkeley (2014)
11. Glockner, C.: Scalar force potentials in rigid multibody systems. In: Pfeiffer, F., Glockner, C. (eds.) Multibody Dynamics with Unilateral Contacts. CISM Courses and Lectures, vol. 421. Springer, Vienna (2000)
12. Lee, E.A.: Modeling concurrent real-time processes using discrete events. Annals of Software Engineering 7, 25–45 (1999)

13. Lee, E.A.: Constructive models of discrete and continuous physical phenomena. Technical report, EECS Department, UC Berkeley (February 2014)
14. Lee, E.A., Sangiovanni-Vincentelli, A.: A framework for comparing models of computation. IEEE Trans. on Computer-Aided Design of Circuits and Systems 17(12), 1217–1229 (1998)
15. Lee, E.A., Tripakis, S.: Modal models in Ptolemy. In: 3rd Int. Workshop on Equation-Based Object-Oriented Modeling Languages and Tools (EOOLT), Oslo, Norway, vol. 47, pp. 11–21. Linköping University Electronic Press, Linköping University (2010)
16. Lee, E.A., Varaiya, P.: Structure and Interpretation of Signals and Systems, 2.0 edn. LeeVaraiya.org (2011)
17. Lee, E.A., Zheng, H.: Operational semantics of hybrid systems. In: Morari, M., Thiele, L. (eds.) HSCC 2005. LNCS, vol. 3414, pp. 25–53. Springer, Heidelberg (2005)
18. Lee, E.A., Zheng, H.: Leveraging synchronous language principles for heterogeneous modeling and design of embedded systems. In: EMSOFT, Salzburg, Austria. ACM (2007)
19. Lindstrøm, T.: An invitation to nonstandard analysis. In: Cutland, N. (ed.) Nonstandard Analysis and Its Applications, pp. 1–105. Cambridge University Press (1988)
20. Maler, O., Manna, Z., Pnueli, A.: From timed to hybrid systems. In: Huizing, C., de Bakker, J.W., Rozenberg, G., de Roever, W.-P. (eds.) REX 1991. LNCS, vol. 600, pp. 447–484. Springer, Heidelberg (1992)
21. Mosterman, P.J., Biswas, G.: A theory of discontinuities in physical system models. Journal of the Franklin Institute 335(3), 401–439 (1998)
22. Mosterman, P.J., Simko, G., Zande, J.: A hyperdense semantic domain for discontinuous behavior in physical system models. In: Multi-Paradigm Modeling, MPM (2013)
23. Otter, M., Elmqvist, H., López, J.D.: Collision handling for the Modelica multibody library. In: Modelica Conference, Hamburg, Germany, pp. 45–53 (2005)
24. Ptolemaeus, C. (ed.): System Design, Modeling, and Simulation using Ptolemy II. Ptolemy.org, Berkeley (2014)
25. Stewart, D.E.: Rigid-body dynamics with friction and impact. SIAM Review 42(1), 3–39 (2000)
26. Tiller, M.M.: Introduction to Physical Modeling with Modelica. Kluwer Academic Publishers (2001)