Hybrid System Modelling and Simulation with Dirac Deltas

Cláudio Gomes
University of Antwerp
Claudio.Gomes@uantwerp.be

Yentl Van Tendeloo
University of Antwerp
Yentl.VanTendeloo@uantwerp.be

Joachim Denil
University of Antwerp
Flanders Make
Joachim.Denil@uantwerp.be

Paul De Meulenaere
University of Antwerp
Flanders Make
Paul.DeMeulenaere@uantwerp.be

Hans Vangheluwe
University of Antwerp
McGill University
Flanders Make
Hans.Vangheluwe@uantwerp.be

December 17, 2016

Revision History

Revision	Date	Author(s)	Description
0.1	15.12.2016	Cláudio et al.	Draft version.

Contents

Abstract

Physical systems are inherently continuous, disallowing any kind of discontinuity. For a wide variety of problems, creating detailed continuous models is, at the very least, impractical. Hybrid models can abstract away short transient behaviour (thus introducing discontinuities) in order to simplify the study of such systems. For example, when modelling a bouncing ball, the bounce can be abstracted as a discontinuous change of the velocity, instead of resorting to the physics of the ball (de-)compression to keep the velocity signal continuous. Impulsive differential equations can be used to model and simulate hybrid systems such as the bouncing ball. In this approach, the force acted on the ball by the floor is abstracted as an infinitely large function in an infinitely small interval of time, that is, an impulse. Current simulators cannot handle such approximations well due to the limitations of machine precision.

In this paper, we explore the simulation of impulsive differential equations, where impulses are first class citizens. We present two approaches for the simulation of impulses: symbolic and numerical. Our contribution is a theoretically founded implementation of both approaches in a CBD modelling and simulation tool. Furthermore, we investigate the conditions for which one approach is better than the other.

Keywords: Dirac delta, hybrid system, distribution theory, causal block diagrams.

1 Introduction

Physical systems are, by nature, continuous [10] and a model of a physical system has to fulfill a pragmatic purpose [1]. If that purpose is to analyze some

behavior at a macro time scale, it is impractical – or even currently impossible [15] – to create high fidelity continuous models that also explain the behavior at micro time scales. Abstraction is a powerful method, necessary to create useful models [11].

As a simple example, consider a ball with unit mass that bounces once on the floor. The macro time scale behavior is the overall trajectory of the ball, whereas the micro time scale behavior captures the compression of the ball in the floor. It can be modelled by following first order Ordinary Differential Equations (ODEs):

$$y^{(2)} = -g + F_c(t) \quad \text{with} \quad y(0) = y_0 \quad \text{and} \quad y'(0) = v_0 \tag{1}$$

where $y^{(2)}$ denotes the acceleration of the ball (second derivative of the position), and y_0 and v_0 are constants. $F_c(t)$ denotes the force impinged by the floor as a result of the collision, which we want to abstract. There are two possible approaches to find the solution $y(t)$ that satisfies Eq. (1), both based on using conservation of momentum to ensure the abstraction of $F_c(t)$ is correct.

Separation of dynamics separates the time continuum into two groups: before and after the contact. Momentum conservation gives the velocity of the ball immediately after the collision. Let t_c denote that time at which the contact occurs. Eq. (1) is then split in two:

$$\begin{pmatrix} y^{(2)} & = -g \\ y(0) & = y_0 \\ y'(0) & = v_0 \end{pmatrix} \quad \text{for } 0 \le t < t_c, \text{ and } \quad \begin{pmatrix} y^{(2)} & = -g \\ y(t_c) & = y(t_c^-) \\ y'(t_c) & = -y'(t_c^-) \end{pmatrix} \quad \text{for } t \ge t_c \tag{2}$$

These are solved separately. t_c can be found from the solution to the leftmost ODE. A simulator can be used to compute the solution in the interval $t \in [0, t_c)$. Afterward, the same simulator can compute the solution in the interval $0 \le t < t_c$, with the new initial conditions. The activity of re-setting the same simulator with new initial conditions is called re-initialization and it is a common approach to the simulation of systems with discontinuities [2].

The **Impulsive forces** approach acknowledges that, since the velocity is the integral of the acceleration, during the collision, $F_c(t)$ must be large enough to cause complete "inversion" of the velocity, even if its shape is unknown. To be concrete, let t_c^- denote the time at which the collision starts and t_c^+ the time at which the collision ends. Then, after the collision, the velocity of the ball is given by

$$y'(t_c^+) = y'(t_c^-) + \int_{t_c^-}^{t_c^+} -g + F_c(\tau) d\tau \tag{3}$$

Conservation of momentum implies that $y'(t_c^+) = -y'(t_c^-)$, and the effect that $-g$ has on the integral can be neglected. Thus:

$$\int_{t_c^-}^{t_c^+} F_c(\tau) d\tau = -2y'(t_c^-) \tag{4}$$

independently of the shape that $F_c(\tau)$ assumes. The Dirac impulse [4] formalizes the abstraction of a large continuous function, whose integral is a known value

over a small interval of time. More details will be given in Section 2 but it suffices to postulate that $\int_{0^-}^{0^+} \delta(\tau)d\tau = 1$. Hence, setting $F_c(\tau) = -2y'(t_c^-)\delta(t-t_c)$ satisfies Eq. (4) and Eq. (3). With this approach, Eq. (1) is rewritten to

$$y^{(2)} = -g - 2y'(t_c^-)\delta(t-t_c) \quad \text{with} \quad y(0) = y_0 \quad \text{and} \quad y'(0) = v_0 \tag{5}$$

which is an impulsive differential equation [8].

In the bouncing ball example, both the separation of dynamics and impulse based approaches assume that the duration of the contact is infinitely small and are able to reach the same solution. The separation of dynamics allows the modeller more freedom in setting the initial states after an impulse occurs (e.g., the initial condition in Eq. (2) could have been $y(t_c) = 0$ instead of $y(t_c) = y(t_c^-)$). This means more mistakes can be made (e.g., violating physical laws).

The re-initialization technique can be used for the simulation of impulsive differential equations. However, for complex models where impulses are regularly exchanged, a direct manipulation – symbolic or numerical – of impulses avoids the need to reset the whole state of the simulator. A symbolic manipulation means that impulses are encoded as first class citizens in the operations of the simulator. And by numerical we mean that impulses are approximated as very large values.

Simulators with support for Dirac impulses have been proposed in [13] and more recently in [9], but both approaches deal with the symbolic computation of impulses, and not their numerical approximation, nor evaluation or comparison is advanced for the approach.

Our *research hypothesis* is: (RH) a simulator which manipulates Dirac deltas symbolically (such as the one proposed in [13]) is more accurate than one that just operates with approximated impulses.

Our *contribution* can be divided into: (C1) a derivation of the decision and inversion symbolic operations performed on Dirac impulses; (C2) a numerical approximation of impulses; (C3) the conditions for which manipulating impulses symbolically yields more accurate solutions than the numerical approximations;

C1 builds on the work of [13], where many of the symbolic operations on Dirac deltas are derived. Here, we derive the remaining operations – decision and inversion – and provide justification to the use of distributions as a formalization of a correct impulse symbolic simulator. C2 and C3 answer RH. An implementation of a simulator capable of simulating Causal Block Diagrams with impulses, symbolically and numerically, as well as the experiments used in this paper, are available at `http://msdl.cs.mcgill.ca/people/claudio/diraccbds`.

The next section provides some background and Section 3 derives the main symbolic operations on impulses. Section 4 describes how impulses are approximated and compares the two approaches (C2 and C3).

2 Background

2.1 Causal Block Diagrams

Causal Block Diagrams (CBD) is a formalism used to model causal systems [14, 6], commonly used in tools such as Simulink®. A CBD is expressed with three main entities: Blocks, Links and Ports. Blocks denote operations or refer to other CBDs, defined elsewhere. A Block may contain multiple input ports and a single output port, except if it represents a CBD, which then may contain any number of output ports. Links establish the data-flow by connecting output ports to input ports. Fig. 1 a) shows an example of a CBD that models a bouncing ball. The CBD on the left has 6 blocks, 3 representing the CBDs defined in Fig. 1 b), c) and d).

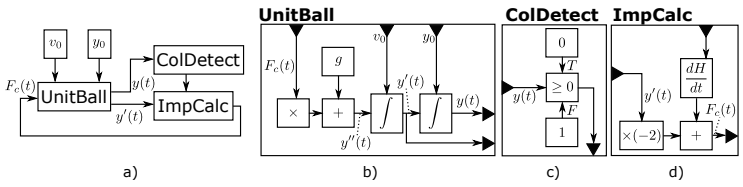

Figure 1: CBD model of a Bouncing Ball with unit mass (right), and specifications of each block (left).

The basic procedure for the simulation of a CBD model is show in Algorithm 1 [12, 17]. For the sake of simplicity, this procedure assumes that there are no algebraic loops in the model[1] [2].

Algorithm 1 CBD Simulator.

1. Flatten CBD by replacing all blocks that refer to CBDs by their specification.
2. Repeat until the end of the simulation:
   (a) Compute dependency graph between blocks and topologically sort the blocks using the dependency graph;
   (b) For each block, compute the outputs from its inputs;
   (c) Advance simulated time by h units of time;

2.2 Distributions

Consider any function $f_k(x)$ that satisfies the conditions:

$$\left(f_k(x) = 0 \text{ if } -\frac{1}{k} < x < \frac{1}{k}\right) \text{ and } \left(\int_{-1/k}^{1/k} f_k(x)dx = 1\right) \tag{6}$$

[1] Algebraic loops are supported by the provided simulator though.

[2] In Fig. 1 the integrator blocks only depend on their previous inputs, therefore no algebraic loop is formed.

An example can be $f_k(x) = \begin{cases} \max(0, k + k^2 x) \text{ if } x \leq 0 \\ \max(0, k - k^2 x) \text{ otherwise} \end{cases}$ plotted in dash stroke in Fig. 2 for $k = 1$.

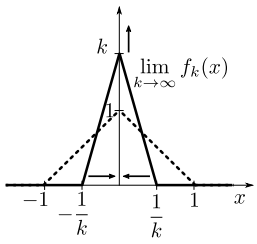

Figure 2: Limit approximation of a Dirac delta.

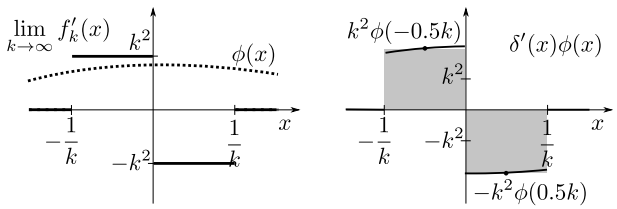

Figure 3: Intuitive visualization of Eq. (9).

Claudio: Verify the above function against the conditions.

A Dirac delta $\delta(x)$ function can be constructed in a way that obeys to Eq. (6) by taking the limit [16], depicted in Fig. 2:

$$\delta(x) = \lim_{k \to \infty} f_k(x) \tag{7}$$

One can informally say that

$$\delta(x) = \begin{cases} 0 \text{ if } x \neq 0 \\ \infty \text{ otherwise} \end{cases}$$

but this is as far as the utility of the statement goes. A more interesting question is what happens when $\delta(x)$ is combined with a smooth (that is, infinitely differentiable) function $\phi(x)$ within an integral, that is, $\int_{-\infty}^{\infty} \delta(x)\phi(x)dx$. According to Equations 6 and 7, and since ϕ is smooth, we have that $\delta(x)\phi(x) = \delta(x)\phi(0)$. To see why, let

$$\varphi_k(x) = \begin{cases} \phi(x) \text{ if } -\frac{1}{k} < x < \frac{1}{k} \\ 0 \text{ otherwise} \end{cases}$$

Then:

$$\delta(x)\phi(x) = \lim_{k \to \infty} (f_k(x)\phi(x)) = \lim_{k \to \infty} (f_k(x)\varphi_k(x)) = \lim_{k \to \infty} f_k(x)\varphi_k(0)$$

The observed behavior then becomes:

$$\begin{aligned} \int_{-\infty}^{\infty} \delta(x)\phi(x)dx &= \int_{-\infty}^{\infty} \delta(x)\phi(0)dx = \phi(0)\int_{-\infty}^{\infty} \delta(x)dx \\ &= \phi(0)\left[\lim_{k \to \infty} \int_{-\frac{1}{k}}^{\frac{1}{k}} f_k(x)dx\right] = \phi(0) \end{aligned} \tag{8}$$

which tells us that the Dirac delta $\delta(x-n)$ can be used to sample a signal ϕ at point n.

Consider now what happens when $\delta(x)$ is differentiated, depicted in Fig. 3:

$$\begin{aligned}
\int_{-\infty}^{\infty} \delta'(x)\phi(x)dx &= \lim_{k\to\infty} \int_{-1/k}^{1/k} \frac{f_k(x) - f_k(x-1/k)}{1/k}\phi(x)dx \\
&= \lim_{k\to\infty} \int_{-1/k}^{0} \frac{f_k(0) - f_k(-1/k)}{1/k}\phi(x)dx + \int_{0}^{1/k} \frac{f_k(1/k) - f_k(0)}{1/k}\phi(x)dx \\
&= \lim_{k\to\infty} \left[\int_{-1/k}^{0} k^2\phi(x)dx + \int_{0}^{1/k} -k^2\phi(x)dx \right] \\
&= \lim_{k\to\infty} \left[k^2\phi(-\frac{1}{2k})\frac{1}{k} - k^2\phi(\frac{1}{2k})\frac{1}{k} \right] \\
&= \lim_{k\to\infty} \left[-\frac{\phi(\frac{1}{2k}) - \phi(-\frac{1}{2k})}{1/k} \right] \\
&= -\phi'(0)
\end{aligned} \tag{9}$$

One can generalize the derivative of a Dirac delta to:

$$\int_{-\infty}^{\infty} \delta^{(i)}(x)\phi(x)dx = (-1)^i\phi^{(i)}(0) \tag{10}$$

These examples show that, even though $\delta(x)$ is not a function in the classical sense, one can observe interesting behavior, and identify an impulse from its interaction with other functions. This is the essential idea of distributions. A distribution is a function identified by the way it interacts with other *test* functions [7, 5, 16], and not by its direct plot (as with classical functions).

Any function $f(x)$ for which $\int_{-\infty}^{\infty} |f(x)|dx < \infty$, can be a distribution by writing it as $\langle f, \phi \rangle$, with

$$\langle f, \phi \rangle \triangleq \int_{-\infty}^{\infty} f(x)\phi(x)dx \quad \text{for any test function } \phi \tag{11}$$

A test function is smooth and becomes 0 outside a bounded interval.

Two distributions are equal if *the result of their interactions with any smooth function is the same*, that is, $f = g \iff \langle f, \phi \rangle = \langle g, \phi \rangle$ for all ϕ. A consequence is that the two sides of Eq. (7) denote the same distribution. This notion of equality allows badly behaved functions (discontinuous and/or with impulses) to exist, as long as their integral is finite. As exemplified before, test functions play a key role because they tend to compensate for the bad shape of the distribution (recall Eq. (10)).

Consider the Heaviside distribution H defined by:

$$H(x) = \begin{cases} 1 \text{ if } x \geq 0 \\ 0 \text{ otherwise} \end{cases} \tag{12}$$

Even though it has a discontinuity at the origin, it can be differentiated as a distribution:

$$\langle H', \phi \rangle = \int_{-\infty}^{\infty} H'(x)\phi(x)dx = [H(x)\phi(x)]_{-\infty}^{\infty} - \int_{-\infty}^{\infty} H(x)\phi'(x)dx = -\int_{0}^{\infty} \phi'(x)dx = -[\phi(x)]_{0}^{\infty} = \phi(0) \tag{13}$$

According to Eq. (8), $\langle \delta, \phi \rangle = \phi(0)$. So, by distribution equality,

$$H' = \delta \tag{14}$$

To see the relationship between distributions and impulsive differential equations, recall the bouncing ball model in Eq. (5), and suppose that its solution is given by

$$y(t) = \underbrace{\left(y_0 + v_0 t - \frac{1}{2}gt^2\right)}_{\text{before collision}}(1 - H(t-t_c)) + \underbrace{\left(y(t_c) - y'(t_c^-)(t-t_c) - \frac{1}{2}g(t-t_c)^2\right)}_{\text{after collision}} H(t-t_c)$$

where t_c is the time at which the collision occurs. This solution is the mix of two trajectories: the one before the bounce, and the one after. It is clear that the solution is a discontinuous function, so considering it a distribution and differentiating it twice yields:

$$\begin{aligned}
\left\langle y^{(2)}, \phi \right\rangle &= \int_{-\infty}^{\infty} y^{(2)}(t)\phi(t)dt = \int_{-\infty}^{\infty} y(t)\phi^{(2)}(t)dt \\
&= \int_{-\infty}^{\infty} \left(y_0 + v_0 t - \frac{1}{2}gt^2\right)(1 - H(t-t_c))\phi^{(2)}(t)dt \\
&\quad + \int_{-\infty}^{\infty} \left(y(t_c) - y'(t_c)(t-t_c) - \frac{1}{2}g(t-t_c)^2\right) H(t-t_c)\phi^{(2)}(t)dt \\
&= \int_{-\infty}^{t_c} \left(y_0 + v_0 t - \frac{1}{2}gt^2\right)\phi^{(2)}(t)dt + \int_{t_c}^{\infty} \left(y(t_c) - y'(t_c)(t-t_c) - \frac{1}{2}g(t-t_c)^2\right)\phi^{(2)}(t)dt \\
&= \left[\left(y_0 + v_0 t - \frac{1}{2}gt^2\right)\phi'(t)\right]_{-\infty}^{t_c} - \left([(v_0 t - gt)\,\phi(t)]_{-\infty}^{t_c} - \int_{-\infty}^{t_c} -g\phi(t)dt\right) \\
&\quad + \left[\left(y(t_c) - y'(t_c)(t-t_c) - \frac{1}{2}g(t-t_c)^2\right)\phi'(t)\right]_{t_c}^{\infty} \\
&\quad - \left(\left[\left(-y'(t_c) - g(t-t_c)\right)\phi(t)\right]_{t_c}^{\infty} - \int_{t_c}^{\infty} -g\phi(t)dt\right) \\
&= y(t_c^-)\phi'(t_c) - y'(t_c^-)\phi(t_c) - g\int_{-\infty}^{t_c} \phi(t)dt - y(t_c^-)\phi'(t) - y'(t_c^-)\phi(t_c) - g\int_{t_c}^{\infty} \phi(t)dt \\
&= -2y'(t_c^-)\phi(t_c) - g\int_{-\infty}^{\infty} \phi(t)dt \\
&= -2y'(t_c^-)\int_{-\infty}^{\infty} \delta(t-t_c)\phi(t)dt - g\int_{-\infty}^{\infty} \phi(t)dt \\
&= \int_{-\infty}^{\infty} \left(-g - 2y'(t_c^-)\delta(t-t_c)\right)\phi(t)dt = \left\langle -g - 2y'(t_c^-)\delta(t-t_c), \phi(t) \right\rangle
\end{aligned} \tag{15}$$

which obeys to Eq. (5). In the next section we apply this theory for the derivation of the symbolic computations on impulses.

3 Symbolic Manipulation of Dirac Deltas

Claudio: The limits of the sum of taus are inconsistent between the survey and report.

In the formalization of Dirac CBDs, the signals transmitted in the links between blocks are distributions that can have discontinuities and impulses (and any derivative of the impulses). Formally, let $\{\tau_j\} \subset \mathbb{R}$ denote the sequence of all times at which an impulse (or any derivative of it) occurs. Then, the signals are of the form:

$$S(t) = s(t) + \sum_{i=0}^{n} \sum_{\tau_j \in \{\tau_j\}} a_{ij} \delta^{(i)}(t - \tau_j) \tag{16}$$

where s denotes a *piece-wise continuous* impulse-free function, n is the maximum derivative order of any impulse in the signal and a_{ij} is a (possibly zero) constant called the impulse coefficient. At any time τ_j impulses occur, both the right and left limits of $S(t)$ have to have the same impulses. This ensures that computations involving impulses disregarding the left and right limits of the signal. Formally,

$$S(t^+) - S(t^-) = s(t^+) - s(t^-) \tag{17}$$

Discontinuities in the impulse free part of the signal, that is, $s(t^+) \neq s(t^-)$, are allowed.

The CBD blocks denote manipulations on distributions in the form of Eq. (16). To ensure that these are compositional, we need to show that the output $Y(t)$ of each block b is of the form of Eq. (16), assuming that the inputs are too. Furthermore, to conform strictly to the theory, the moment any signal is plotted, it should be plotted as $\langle S(t), \varphi(t) \rangle$, for suitable test function. However, the utility of this simulation package is that it allows the modeller to observe the discontinuities and impulses in a signal, so any impulse is plotted with an arrow. The height of the arrow represents the impulse coefficient. An example is shown in Fig. 4 a). We now provide a derivation of each operation.

The **Sum** block takes two signals ($U(t)$ and $V(t)$) as inputs and produces

$Y(t)$ as output:

$$
\begin{aligned}
Y(t) = U(t)+V(t) &\Leftrightarrow \langle Y(t),\varphi(t)\rangle = \langle U(t)+V(t),\varphi(t)\rangle \qquad \text{for any test function } \phi \\
&= \langle U(t),\varphi(t)\rangle + \langle V(t),\varphi(t)\rangle \text{ by linearity} \\
&= \left\langle u(t)+\sum_{i=0}^{n_u}\sum_{j=0}^{m_u} a_{ij}\delta^{(i)}(t-\tau_j),\varphi(t)\right\rangle + \left\langle v(t)+\sum_{i=0}^{n_v}\sum_{\tau_j\in\{\tau_j\}} a_{ij}\delta^{(i)}(t-\tau_j),\varphi(t)\right\rangle \\
&= \langle u(t),\varphi(t)\rangle + \langle v(t),\varphi(t)\rangle + \sum_{i=0}^{n_u}\sum_{j=0}^{m_u} a_{ij}\left\langle \delta^{(i)}(t-\tau_j),\varphi(t)\right\rangle + \sum_{i=0}^{n_v}\sum_{\tau_j\in\{\tau_j\}} a_{ij}\left\langle \delta^{(i)}(t-\tau_j),\varphi(t)\right\rangle \\
&\quad \text{by inverse linearity} \\
&= \left\langle u(t)+v(t)+\sum_{i=0}^{n_u}\sum_{j=0}^{m_u} a_{ij}\delta^{(i)}(t-\tau_j)+\sum_{i=0}^{n_v}\sum_{\tau_j\in\{\tau_j\}} a_{ij}\delta^{(i)}(t-\tau_j),\varphi(t)\right\rangle
\end{aligned}
\tag{18}
$$

Operationally, it adds the two impulse-free signals $u(t)$ and $v(t)$, and adds the coefficients of the impulses. $Y(t)$ is of the form of Eq. (16) so the derivation is complete.

The **Negation** block is derived in the same way. It negates the impulse-free part of the signal and all the coefficients.

$$
\begin{aligned}
\langle Y(t),\varphi(t)\rangle &= \langle -U(t),\varphi(t)\rangle \\
&= \left\langle -\left(u(t)+\sum_{i=0}^{n_u}\sum_{j=0}^{m_u} a_{ij}\delta^{(i)}(t-\tau_j)\right),\varphi(t)\right\rangle \\
&= -\langle u(t),\varphi(t)\rangle - \sum_{i=0}^{n_u}\sum_{j=0}^{m_u} a_{ij}\left\langle \delta^{(i)}(t-\tau_j),\varphi(t)\right\rangle \\
&= \left\langle -u(t)-\sum_{i=0}^{n_u}\sum_{j=0}^{m_u} a_{ij}\delta^{(i)}(t-\tau_j),\varphi(t)\right\rangle
\end{aligned}
\tag{19}
$$

The **Switch** block outputs either 0 or 1, according to a condition on the input $C(t)$. A family of pathological cases related to impulses can be identified if $C(t)$ is allowed to have impulses. For example, does a function which at a point $t_0 > 0$ is positive, but has a negative impulse (e.g., $C(t_0) = c(t_0)^2 - \delta(0)$) cross the zero? So $C(t)$ is assumed to be free of impulses, that is, $C(t) = c(t)$. The derivation is then:

$$
\langle Y(t),\varphi(t)\rangle = \langle H(c(t)),\varphi(t)\rangle \tag{20}
$$

where $H(x)$ is the Heaviside distribution defined in Eq. (12). Alternatively, it can be written as:

$$
\langle Y(t),\varphi(t)\rangle = \left\langle \begin{cases} 1 \text{ if } c(t)\geq 0 \\ 0 \text{ otherwise} \end{cases},\varphi(t)\right\rangle \tag{21}
$$

Note however, that $c(t)$ can still be piece-wise continuous. Then the left limit of the output, denoted as $Y(t^-)$, is evaluated with $c(t^-) > 0$ and the right limit $Y(t^+)$ with $c(t^+) \geq 0$. In other words, for any time t_0 where $c(t_0^-) < 0$ and $c(t_0^+) > 0$, $H(c(t_0^-)) = 0$ and $H(c(t_0^+)) = 1$.

The **Decision** block is a generalization of the Switch block. It forwards one of two inputs ($U(t)$ and $V(t)$) to the output $Y(t)$ according to an input condition $C(t) \geq 0$. Similarly to the Switch block, we have to assume that $C(t) = c(t)$ but in addition, at any time t_0 where $c(t_0^-) < 0$ and $c(t_0^+) \geq 0$, there can be no impulses on either $U(t_0)$ or $V(t_0)$. If this were allowed, a signal could be created which violates Eq. (17). Under these assumptions, the derivation is:

$$\langle Y(t), \varphi(t) \rangle = \langle U(t)H(C(t)) + V(t)(1 - H(C(t))), \varphi(t) \rangle \tag{22}$$

The output of the Decision block can also be written as

$$\langle Y(t), \varphi(t) \rangle = \left\langle \begin{cases} U(t) \text{ if } c(t) \geq 0 \\ V(t) \text{ otherwise} \end{cases}, \varphi(t) \right\rangle \tag{23}$$

The **Derivative** block is partially derived as follows:

$$\begin{aligned} \langle Y(t), \varphi(t) \rangle &= \langle U'(t), \varphi(t) \rangle \\ &= \left\langle u'(t) + \left[\sum_{i=0}^{n_u} \sum_{j=0}^{m_u} a_{ij} \delta^{(i)}(t - \tau_j)(t) \right]', \varphi(t) \right\rangle \\ &= \langle u'(t), \varphi(t) \rangle + \left\langle \sum_{i=0}^{n_u} \sum_{j=0}^{m_u} a_{ij} \delta^{(i+1)}(t - \tau_j)(t), \varphi(t) \right\rangle \end{aligned} \tag{24}$$

The term $\langle u'(t), \varphi(t) \rangle$ must be further developed because it can have discontinuities. For example, when the Derivative block is connected to the output of a Decision block. Let $\{t_d\}$ be the countable sequence of times at which $u(t_d^-) \neq u(t_d^+)$. In the proximity of a discontinuity at time t_d, $u(t)$ is described as $u(t) = u(t^-)(1 - H(t - t_d)) + u(t^+)H(t - t_d)$. Intuitively, it can be seen as the output signal of a Decision block. $\langle u'(t), \varphi(t) \rangle$ is then:

$$\begin{aligned} \langle u'(t), \varphi(t) \rangle &= \left\langle \left[u(t^-)(1 - H(t - t_d)) + u(t^+)H(t - t_d) \right]', \varphi(t) \right\rangle \\ &= \left\langle \left[u(t^-) + (u(t^+) - u(t^-))H(t - t_d) \right]', \varphi(t) \right\rangle \\ &= \left\langle u'(t^-) + \left[(u(t^+) - u(t^-))H(t - t_d) \right]', \varphi(t) \right\rangle \\ &= \langle u'(t^-), \varphi(t) \rangle + \left\langle \left[(u(t^+) - u(t^-))H(t - t_d) \right]', \varphi(t) \right\rangle \end{aligned} \tag{25}$$

The distribution $\left\langle [(u(t^+)-u(t^-))H(t-t_d)]', \varphi(t) \right\rangle$ can be further simplified:

$$\begin{aligned}
\left\langle \left[(u(t^+)-u(t^-))H(t-t_d)\right]', \varphi(t) \right\rangle &= -\left\langle (u(t^+)-u(t^-))H(t-t_d), \varphi'(t) \right\rangle \\
&= -\int_{-\infty}^{\infty} (u(t^+)-u(t^-))H(t-t_d)\varphi'(t)dt \\
&= -\int_{t_d}^{\infty} (u(t^+)-u(t^-))\varphi'(t)dt \\
&\text{integration by parts} \\
&= -\left(\left[(u(t^+)-u(t^-))\varphi(t)\right]_{t_d}^{\infty} \right) - \int_{t_d}^{\infty} (u'(t^+)-u'(t^-))\varphi(t)dt \\
&= (u(t_d^+)-u(t_d^-))\varphi(t_d) + \int_{-\infty}^{\infty} (u'(t^+)-u'(t^-))H(t-t_d)\varphi(t)dt \\
&= (u(t_d^+)-u(t_d^-))\left\langle \delta(t-t_d), \varphi(t) \right\rangle + \left\langle (u'(t^+)-u'(t^-))H(t-t_d), \varphi(t) \right\rangle
\end{aligned} \tag{26}$$

In conclusion, arround each discontinuity $t_d \in T_d$, we have:

$$\left\langle u'(t), \varphi(t) \right\rangle = \left\langle u'(t^-)(1-H(t-t_d)) + u'(t^+)H(t-t_d) + (u(t_d^+)-u(t_d^-))\delta(t-t_d), \varphi(t) \right\rangle \tag{27}$$

This is interpreted as originating an impulse, with a coefficient given by the magnitude of the discontinuity. The derivative signal can itself be discontinuous, if its left limit is also different than its right limit. Across all possible discontinuities $t_d \in T_d$ of $u(t)$, and abstracting the discontinuities in $u'(t)$, the signal in Eq. (27) is in the form of Eq. (16). Concretely:

$$\left\langle u'(t), \varphi(t) \right\rangle = \left\langle u'(t) + \sum_{t_d \in T_d} (u(t_d^+)-u(t_d^-))\delta(t-t_d), \varphi(t) \right\rangle \tag{28}$$

Joining Eq. (27) and Eq. (24) yields:

$$\langle Y(t), \varphi(t) \rangle = \left\langle u'(t) + \sum_{t_d \in \{t_d\}} (u(t_d^+)-u(t_d^-))\delta(t-t_d) + \sum_{i=0}^{n_u} \sum_{\tau_j \in \{\tau_j\}} a_{ij}\delta^{(i+1)}(t-\tau_j)(t), \varphi(t) \right\rangle \tag{29}$$

For example, suppose the input to the Derivative block is the signal

$$U(t) = (v_0 - gt)(1-H(t-t_d)) + (-(v_0-gt_d) - g(t-t_d))H(t-t_d)$$

With v_0,g constants. Then the output of the Derivative block is

$$\langle Y(t), \varphi(t) \rangle = -g - 2(v_0 - gt_d)\delta(t-t_d)$$

The **Integrator** block is derived as:

$$\begin{aligned}
\langle Y(t), \varphi(t) \rangle &= \left\langle \int_0^t U(x)dx, \varphi(t) \right\rangle \\
&= \left\langle \int_0^t u(x) + \sum_{i=0}^{n_u} \sum_{j=0}^{m_u} a_{ij} \delta^{(i)}(x-\tau_j)dx, \varphi(t) \right\rangle \\
&= \left\langle \int_0^t u(x)dx, \varphi(t) \right\rangle + \sum_{i=0}^{n_u} \sum_{j=0}^{m_u} a_{ij} \left\langle \int_0^t \delta^{(i)}(x-\tau_j)dx, \varphi(t) \right\rangle \\
&= \left\langle \int_0^t u(x)dx, \varphi(t) \right\rangle + \sum_{j=0}^{m_u} a_{0j} \left\langle \int_0^t \delta(x-\tau_j)dx, \varphi(t) \right\rangle + \sum_{i=1}^{n_u} \sum_{j=0}^{m_u} a_{ij} \left\langle \delta^{(i-1)}(t-\tau_j), \varphi(t) \right\rangle \\
&= \left\langle \int_0^t u(x)dx, \varphi(t) \right\rangle + \sum_{j=0}^{m_u} a_{0j} \left\langle H(x-\tau_j), \varphi(t) \right\rangle + \sum_{i=1}^{n_u} \sum_{j=0}^{m_u} a_{ij} \left\langle \delta^{(i-1)}(t-\tau_j), \varphi(t) \right\rangle \\
&= \left\langle \int_0^t u(x)dx + \sum_{j=0}^{m_u} a_{0j} H(x-\tau_j) + \sum_{i=1}^{n_u} \sum_{j=0}^{m_u} a_{ij} \delta^{(i-1)}(t-\tau_j), \varphi(t) \right\rangle
\end{aligned} \tag{30}$$

Which matches the intuition: the Integrator block computes the normal integration of the impulse-free signal, reduces the order of any impulse derivative, and computes a discontinuity whenever an impulse is integrated (recall Eq. (14)). The magnitude of the discontinuity is the impulse coefficient.

The **Product** of two distributions is not defined in general. However, if one of the input signals to the Product block is impulse-free[3], then it can be derived

[3]This restriction cannot be enforced automatically by the tool.

as:

$$
\begin{aligned}
\langle Y(t),\varphi(t)\rangle &= \langle U(t)V(t),\varphi(t)\rangle \\
&= \left\langle u(t)\left(v(t)+\sum_{i=0}^{n_v}\sum_{\tau_j\in\{\tau_j\}} a_{ij}\delta^{(i)}(t-\tau_j)\right),\varphi(t)\right\rangle \text{ by assumption} \\
&= \langle u(t)v(t),\varphi(t)\rangle + \left\langle \sum_{i=0}^{n_v}\sum_{\tau_j\in\{\tau_j\}} a_{ij}u(t)\delta^{(i)}(t-\tau_j),\varphi(t)\right\rangle \\
&= \langle u(t)v(t),\varphi(t)\rangle + \sum_{i=0}^{n_v}\sum_{\tau_j\in\{\tau_j\}} a_{ij}\left\langle u(t)\delta^{(i)}(t-\tau_j),\varphi(t)\right\rangle \\
&\text{by product property} \\
&= \langle u(t)v(t),\varphi(t)\rangle + \sum_{i=0}^{n_v}\sum_{\tau_j\in\{\tau_j\}} a_{ij}\left\langle \delta^{(i)}(t-\tau_j),u(t)\varphi(t)\right\rangle \\
&\text{by derivative} \\
&= \langle u(t)v(t),\varphi(t)\rangle + \sum_{i=0}^{n_v}\sum_{\tau_j\in\{\tau_j\}} a_{ij}(-1)^i\left\langle \delta(t-\tau_j),[u(t)\varphi(t)]^{(i)}\right\rangle \\
&\text{by general Leibniz rule} \\
&= \langle u(t)v(t),\varphi(t)\rangle + \sum_{i=0}^{n_v}\sum_{\tau_j\in\{\tau_j\}} a_{ij}(-1)^i\left\langle \delta(t-\tau_j),\sum_{k=0}^{i}\binom{i}{k}u^{(k)}(t)\varphi^{(i-k)}(t)\right\rangle \\
&\text{by linearity} \\
&= \langle u(t)v(t),\varphi(t)\rangle + \sum_{i=0}^{n_v}\sum_{\tau_j\in\{\tau_j\}} a_{ij}(-1)^i\sum_{k=0}^{i}\binom{i}{k}\left\langle \delta(t-\tau_j),u^{(k)}(t)\varphi^{(i-k)}(t)\right\rangle \\
&\text{by delta property} \\
&= \langle u(t)v(t),\varphi(t)\rangle + \sum_{i=0}^{n_v}\sum_{\tau_j\in\{\tau_j\}} a_{ij}(-1)^i\sum_{k=0}^{i}\binom{i}{k}u^{(k)}(\tau_j)\varphi^{(i-k)}(\tau_j) \\
&\text{by reverse delta property and } u^{(k)}(\tau_j) \text{ constant} \\
&= \langle u(t)v(t),\varphi(t)\rangle + \sum_{i=0}^{n_v}\sum_{\tau_j\in\{\tau_j\}} a_{ij}(-1)^i\sum_{k=0}^{i}\binom{i}{k}u^{(k)}(\tau_j)\left\langle \delta(t-\tau_j),\varphi^{(i-k)}(t)\right\rangle \\
&\text{by property } (-1)^{i-k}(-1)^{i-k}=1 \\
&= \langle u(t)v(t),\varphi(t)\rangle + \sum_{i=0}^{n_v}\sum_{\tau_j\in\{\tau_j\}} a_{ij}(-1)^i\sum_{k=0}^{i}\binom{i}{k}u^{(k)}(\tau_j)(-1)^{i-k}(-1)^{i-k}\left\langle \delta(t-\tau_j),\varphi^{(i-k)}(t)\right\rangle \\
&\text{by reverse derivative property} \\
&= \langle u(t)v(t),\varphi(t)\rangle + \sum_{i=0}^{n_v}\sum_{\tau_j\in\{\tau_j\}} a_{ij}(-1)^i\sum_{k=0}^{i}\binom{i}{k}u^{(k)}(\tau_j)(-1)^{i-k}\left\langle \delta^{(i-k)}(t-\tau_j),\varphi(t)\right\rangle \\
&= \langle u(t)v(t),\varphi(t)\rangle + \sum_{i=0}^{n_v}\sum_{\tau_j\in\{\tau_j\}} {}^{14}a_{ij}\sum_{k=0}^{i}\binom{i}{k}u^{(k)}(\tau_j)(-1)^i(-1)^{i-k}\left\langle \delta^{(i-k)}(t-\tau_j),\varphi(t)\right\rangle \\
&\text{by fact } (-1)^i(-1)^{i-k}=(-1)^k \\
&= \langle u(t)v(t),\varphi(t)\rangle + \sum_{i=0}^{n_v}\sum_{\tau_j\in\{\tau_j\}} a_{ij}\sum_{k=0}^{i}\binom{i}{k}u^{(k)}(\tau_j)(-1)^k\left\langle \delta^{(i-k)}(t-\tau_j),\varphi(t)\right\rangle
\end{aligned}
\tag{31}
$$

Essentially, each product between a function and a Dirac delta derivative yields a set of simultaneous impulses of decreasing derivative orders, that are multiplied with increasing derivatives of the function. The computation of the coefficients, therefore, requires that the derivatives of $u(t)$, order up to (and including) n_v, be known or estimated. To get the intuition, recall Eq. (9) and Fig. 3, and note that

$$\left\langle \delta'(t-\tau_j)u(t), \varphi(t) \right\rangle = -\left[u(t-\tau_j)\varphi(t-\tau_j)\right]' = \left\langle u(t)\delta' - u'(t)\delta, \varphi(t) \right\rangle \tag{32}$$

For example, if $U(t) = -gt$ and $V(t) = 1 + 20\delta^{(2)}(t-1.44)$, then the Product block computes:

$$\begin{aligned}\langle U(t)V(t), \varphi(t)\rangle &= \langle -gt \times 1, \varphi(t)\rangle + 20\sum_{k=0}^{2}\binom{2}{k}u^{(k)}(1.44)(-1)^k\left\langle \delta^{(2-k)}(t-1.44), \varphi(t)\right\rangle \\ &= \langle -gt, \varphi(t)\rangle - 28.8g\left\langle \delta^{(2)}(t-1.44), \varphi(t)\right\rangle + 40g\left\langle \delta^{(1)}(t-1.44), \varphi(t)\right\rangle\end{aligned} \tag{33}$$

The **Inverter** block, to be well defined, requires that the input is free of impulses:

$$\langle Y(t), \varphi(t)\rangle = \left\langle \frac{1}{u(t)}, \varphi(t)\right\rangle \tag{34}$$

This section has provided the relationship between DiracCBDs and Distributions, and derived each block used in the CBD formalism. In the DiracCBDs simulator the impulses are treated symbolically but the rest of the signals are discretized approximations of the continuous system. In particular, if $h > 0$ is the simulation time step (recall Algorithm 1), then the integrator block can be approximated by recursive right Riemann's sum and the derivative block by finite difference:

$$Y_{\text{int}}(t) \approx Y_{\text{int}}(t-h) + U(t) \times h \quad \text{and} \quad Y_{\text{der}}(t) \approx \frac{U(t) - U(t-h)}{h} \tag{35}$$

The signals, in the form of Eq. (16), can be encoded in many ways. For example, the left and right limit of the signals can be stored so that discontinuities can be easily identified, and a matrix of impulse coefficients kept. Each block then takes the input left/right limit and the matrix, and produces an output of the same form. [13] encodes the left limit of the signal, along with all its derivatives (lazily evaluated). Another example encoding can be seen in [9] that is based in the super dense time encoding. Fig. 4 a) shows the bouncing ball trace produced by our simulator.

4 Numerical Approximation of Dirac Deltas

The previous section offered insight on how to symbolically manipulate impulses in the context of simulation. This manipulation can get complex and counter

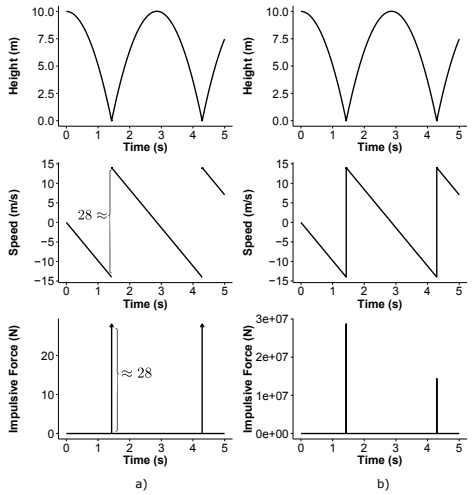

Figure 4: Bouncing ball trace, with perfectly elastic collision. The symbolic-impulse approach is a) and the approximated-impulse one is b).

intuitive, especially with derivatives of impulses (recall Eq. (31) and Eq. (32)), and requires a special encoding of the signals. The purpose of this section is to describe an alternative approach by approximating Dirac deltas as large functions, and see how it compares to the symbolic approach.

A Dirac delta impulse is a arbitrary high function at a very small interval of time that obeys to Eq. (6). In a numerical simulation, the smallest interval is h, so a good numerical approximation of an impulse $\delta(t-\tau_d)$ would be to produce a large value N at a simulation step that is immediately after τ_d:

$$\delta(t-\tau_d) \approx \begin{cases} N \text{ if } 0 \leq t-\tau_d < h \\ 0 \text{ otherwise} \end{cases} \tag{36}$$

We cannot just pick an arbitrary high value for N (e.g., $N = 2^{31}-1$) because of the conditions in Eq. (6).

Eq. (14) holds the key to numerically approximate the impulse. Suppose the Dirac delta is constructed as:

$$\delta(x) = \lim_{k\to\infty} H'_k(x) = \lim_{k\to\infty} \begin{cases} \frac{1}{2}k \text{ if } -\frac{1}{k} \leq x \leq \frac{1}{k} \\ 0 \text{ otherwise} \end{cases} \quad \text{with} \quad H_k(x) = \begin{cases} 0 & \text{if } x < -\frac{1}{k} \\ \frac{1}{2}+\frac{1}{2}kx & \text{if } -\frac{1}{k} \leq x \leq \frac{1}{k} \\ 1 & \text{if } x > \frac{1}{k} \end{cases} \tag{37}$$

where $H_k(x)$ represents a continuous approximation of the Heaviside function, for large enough k: This formulation satisfies the conditions in Eq. (6), and it represents the limit version of the result in Eq. (14). Since h is the smallest interval of time we have, the impulse is approximated as $\delta(t-\tau_d) \approx H'_{1/h}(t-\tau_d)$,

where $H'_{1/h}$ is approximated as in Eq. (35). The smaller h is, the more accurate the approximations are.

This approach works remarkably well for the bouncing ball model, producing an *exactly equal* trace shown in Fig. 4 b). Numerical approximations different than those in Eq. (35) may make the traces differ. One immediate effect is that the values plotted for the impulsive force change by several orders of magnitude. Due to the numerical approximation for the derivative, the smaller h is, the larger the impulse will be. While smaller h increases the accuracy, it may cause overflows due to the limits of machine precision. Note that the time at which the ball touches the floor still needs to be accurately located [18] by adjusting h and our simulator does this (both the symbolic and the numerical versions) which explains why Fig. 4 b) has different values for the impulses.

Now consider the following artificial signal and its distributional derivatives up to order n:

$$S(t) = H(t-\tau_d)\ ;\ S'(t) = \delta(t-\tau_d)\ ;\ S^{(2)}(t) = \delta'(t-\tau_d)\ ;\ \ldots\ ;\ S^{(n)}(t) = \delta^{(n-1)}(t-\tau_d) \tag{38}$$

We compare this signal with the approximation $\tilde{S}(t) \approx H_{1/h}(t-\tau_d)$ from Eq. (37), differentiated according to the approximation in Eq. (35). For example, the derivative $H'_{1/h}(t-\tau_d)$ is approximated as:

$$H'_{1/h}(\tau_d) \approx \frac{H_{1/h}(\tau_d) - H_{1/h}(\tau_d - h)}{h} \approx \frac{1}{h}$$

Table 2 gives the values of the these approximations around time τ_d. The table shows that $(n+1)$ time steps are necessary in the numerical simulator to represent the information that is symbolically stored in an impulse derivative of order n. This shifts the numerical solution by $n \times h$ time units, which may cause significant errors. Intuitively, the symbolic impulse derivatives represent a compact version of what the numerical equivalent would do across the multiple infinitesimal time steps that are abstracted away.

Table 2: Approximated derivatives of $S(t)$ (Eq. (38)).

Time	$S(t)$	$S'(t)$	$S^{(2)}(t)$	$S^{(3)}(t)$	...	$S^{(n-1)}(t)$	$S^{(n)}(t)$
$\tau_d - h$	0	0	0	0	...	0	0
τ_d	1	$1/h$	$1/h^2$	$1/h^3$	...	$1/h^n$	$1/h^n$
$\tau_d + h$	1	0	$-1/h^2$	$-2/h^3$	...	$-(n-2)/h^n$	$-(n-1)/h^n$
$\tau_d + 2h$	1	0	0	$1/h^3$	...	$\binom{n-2}{2}/h^n$	$\binom{n-1}{2}/h^n$
$\tau_d + 3h$	1	0	0	0	...	$-\binom{n-2}{3}/h^n$	$-\binom{n-1}{3}/h^n$
...	...	...	...	...	...	...	...
$\tau_d + (n-1)h$	1	0	0	0	...	0	$(-1)^{n-1}\binom{n-1}{n-1}/h^n$
$\tau_d + nh$	1	0	0	0	...	0	0

In the approximation of $S^{(n)}(t)$, the table, resembling Pascal's triangle, yields

the maximum magnitude of the values computed:

$$\binom{k-1}{k-1}/h^k \quad \text{where} \quad k = \mathit{floor}\left(\frac{n}{2}\right) \tag{39}$$

Eq. (39) can be used to get a rough estimate of the maximum magnitude of values computed in a simulation of an impulsive differential equation. For example, if the maximum impulse derivative order is n, and the maximum amplitude of any discontinuity in the simulation is D, then the maximum magnitude is $D\binom{k-1}{k-1}/h^k$. This result follows from Eq. (39) and the properties of distributions, described in Section 2.2 and Section 3.

To summarize, for models that contain no impulse derivatives, the numerical approximation is equivalent to the symbolic manipulation. For models that have impulse derivatives, the symbolic approach is more accurate than the numerical one, due to the delay introduced, which is proportional to the highest derivative order of the impulses.

5 Conclusion

This work has explored the use of Dirac delta impulses for the modelling and simulation of hybrid systems represented by impulsive differential equations. Section 4 compares a simulator that approximates impulses numerically with a simulator that encodes the impulses explicitly in the signal. We conclude that for models that have no impulse derivatives, both approaches are exactly the same, for the approximations in Eq. (35). The models that *we* have used *do not* have impulse derivatives and we failed at finding models of physical systems that include impulse derivatives. The numerical approach described in Section 4 is also more efficient, due to a simpler encoding. The encoding is a stream of real numbers, as opposed to Eq. (16).

For models that have impulse derivatives however, the numerical approach introduces a delay in the signal, proportional to the highest derivative order of the impulses, which causes inaccuracies. Furthermore, if the models are to be run on an embedded platform, where overflows pose a more significant risk, then the symbolic approach may be used. Eq. (39) provides a rough estimate of the magnitudes involved when using the numerical approach. As can be seen, the smaller the simulation step size h is, the larger Eq. (39) will be. This poses an interesting challenge because in general h has to be small in order to get accurate results. For the symbolic approach, we hypothesize that it may require less bits to perform the same computations on an embedded platform, because the magnitudes involved may not be dominated by the impulse coefficients. Finally, the symbolic approach has the benefit of begin able to identify when operations that are not defined in theory, are being computed (e.g., an inversion of an impulse). As [13] points out, this is not fool proof because the signals are computed point-wise but, compared to the numerical approach, it is an improvement.

Acknowledgment

This research was partially supported by Flanders Make vzw, the strategic research centre for the manufacturing industry, and PhD fellowship grants from the Agency for Innovation by Science and Technology in Flanders (IWT).

References

[1] François Edouard Cellier. *Continuous system modeling*. Springer Science & Business Media, 1991.

[2] François Edouard Cellier and Ernesto Kofman. *Continuous System Simulation*. Springer Science & Business Media, 2006.

[3] S P Corwin, S Thompson, and S M White. Solving ODEs and DDEs with Impulses1. *JNAIAM*, 3(1-2):139–149, 2008.

[4] Paul Adrien Maurice Dirac. *The principles of quantum mechanics*. Number 27. Oxford university press, 1981.

[5] Friedrich Gerard Friedlander and Mark Suresh Joshi. *Introduction to the Theory of Distributions*. Cambridge University Press, 1998.

[6] Cláudio Gomes, Joachim Denil, and Hans Vangheluwe. Causal-Block Diagrams. Technical report, 2016.

[7] John Horvath. An Introduction to Distributions. *The American Mathematical Monthly*, 77(3):227, mar 1970.

[8] Vangipuram Lakshmikantham, Drumi D Bainov, and Pavel S Simeonov. *Theory of impulsive differential equations*, volume 6. World scientific, 1989.

[9] Edward A Lee, Mehrdad Niknami, Thierry S Nouidui, and Michael Wetter. Modeling and Simulating Cyber-physical Systems Using CyPhySim. In *Proceedings of the 12th International Conference on Embedded Software*, EMSOFT '15, pages 115–124, Piscataway, NJ, USA, 2015. IEEE Press.

[10] Pieter J. Mosterman and Gautam Biswas. Formal specifications for hybrid dynamical systems. In *IJCAI-97*, pages 568–577, 1997.

[11] Pieter J. Mosterman and Gautam Biswas. A theory of discontinuities in physical system models. *Journal of the Franklin Institute*, 335(3):401–439, apr 1998.

[12] Sadaf Mustafiz, Cláudio Gomes, Bruno Barroca, and Hans Vangheluwe. Modular Design of Hybrid Languages by Explicit Modeling of Semantic Adaptation. In *Proceedings of the Symposium on Theory of Modeling & Simulation: DEVS Integrative M&S Symposium*, DEVS '16, pages 29:1—-29:8, San Diego, CA, USA, 2016.

[13] Henrik Nilsson. Functional automatic differentiation with dirac impulses. *ACM SIGPLAN Notices*, 38(9):153–164, sep 2003.

[14] Ernesto Posse, Juan de Lara, and Hans Vangheluwe. Processing causal block diagrams with graphgrammars in atom3. In *European Joint Conference on Theory and Practice of Software (ETAPS), Workshop on Applied Graph Transformation (AGT)*, pages 23–34, 2002.

[15] D Stewart. Rigid-Body Dynamics with Friction and Impact. *SIAM Review*, 42(1):3–39, jan 2000.

[16] Robert S Strichartz. *A guide to distribution theory and Fourier transforms*. World Scientific, 2003.

[17] Hans Vangheluwe, Joachim Denil, Sadaf Mustafiz, Daniel Riegelhaupt, and Simon Van Mierlo. Explicit Modelling of a CBD Experimentation Environment. In *Proceedings of the Symposium on Theory of Modeling & Simulation - DEVS Integrative*, DEVS '14, San Diego, CA, USA, 2014. Society for Computer Simulation International.

[18] Fu Zhang, Murali Yeddanapudi, and Pieter Mosterman. Zero-crossing location and detection algorithms for hybrid system simulation. In *IFAC World Congress*, pages 7967–7972, 2008.

A Experiment: Bouncing Ball

A.1 Physical Model

A free falling ball, with no friction, is governed by the following ordinary differential equation:

$$\begin{aligned} v(t)' &= -g \\ y(t)' &= v(t) \end{aligned} \tag{40}$$

Momentum conservation allows us to abstract what happens during the moment of collision, and give us the velocity of the ball after the collision:

$$v(t_0 + \varepsilon) = -v(t_c - \varepsilon)$$

where t_0 is the time of the collision, ε is an infinitely small constant, and the mass of the ball is assumed to be 1.

For the bouncing ball, the use of Dirac delta provide a compact notation to represent the model as an impulsive differential equation [3]:

$$\begin{aligned} v(t)' &= -g + \sum_{i=0} -2v(t_i - \varepsilon)\delta(t - t_i) \\ y(t)' &= v(t) \end{aligned} \tag{41}$$

where:
t_i denotes a time of the collision;

$v(t_i - \varepsilon)$ denotes the speed of the ball immediately before the collision;
$-2v(t_i - \varepsilon)$ represents the change in the speed of the ball, cause by the impulse occurring at time t_i, that is $v(t_0 + \varepsilon) - v(t_c - \varepsilon) = -2v(t_c - \varepsilon)$;

Normally, Eq. (41) can be solved analytically by splitting the time continuum into open intervals and defining the appropriate initial values. For example, if the initial values are $y(0) = 10$ and $v(0) = 0$, then, in the interval $t \in (0, t_0)$ the trajectory of the ball is given by Eq. (40) which can be easily solved. t_0 can be found by solving $y(t_0) = 0$. In the interval $t \in (t_0, t_1)$, the initial values are $y(0) = 0$ and $v(t_1 + \varepsilon) = -v(t_1 - \varepsilon)$, and the solution is again given by Eq. (40).

The current example is simple enough to be solved analytically. For most cases in practice, the impulsive differential equations can be non-linear, and simulation is used to observe the behavior.

A.2 CBD Model

Fig. 1 shows the CBD of the bouncing ball. The Ball CBD represents the ball dynamics in free fall, assuming no friction and constant gravitational acceleration. When collisions occur, an impulse occurs, as in Eq. (41). In CBDs, this impulse is generated when the derivative of a discontinuity is performed. The CollisionDetector emits a discontinuity from 0 to 1, when the position of the ball y crosses the 0 threshold. The ImpulseCalculator essentially performs the derivative of the signal coming from the CollisionDetector (represented by the RisingEdge block), which is an impulse when a collision is detected, and multiplies that with $-2v$.

Arround the point of a single collision, the model represents the following equations:

$$\begin{aligned} a(t) &= \frac{I(t) - mg}{m} \\ v(t) &= \int_0^t a(u)du + v(0) \\ y(t) &= \int_0^t v(u)du + y(0) \\ c(t) &= \begin{cases} 1 \text{ if } y(t) \leq 0 \\ 0 \text{ otherwise} \end{cases} \\ I(t) &= -2v(t)c(t)' \end{aligned} \tag{42}$$

where:
$I(t)$ is the impulsive force;
$a(t)$ is the total acceleration imposed on the ball;
$v(t)$ is the velocity of the ball;
$c(t)$ is the output of the CollisionDetector block;
$m = 1$ is the mass of the ball;

Eq. (42), for the purposes of simulation, is approximated by:

$$\begin{aligned} a(t) &= \frac{I(t)-mg}{m} \\ v(t) &= v(t-h)+a(t-h)h \\ y(t) &= y(t-h)+v(t-h)h \\ c(t) &= \begin{cases} 1 \text{ if } y(t) \leq 0 \\ 0 \text{ otherwise} \end{cases} \\ I(t) &= -2v(t)\frac{c(t)-c(t-h)}{h} \end{aligned} \tag{43}$$

where
$h < 1$ is the discretization factor;
$v(0) = 0$ is the initial speed;
$y(0) = 10$ is the initial position;

By inspecting Eq. (42), one notes that a collision only has an effect on the ball at the next time step. This is a common assumption and the other approach, i.e., use the values $a(t)$ and $v(t)$ to approximate $v(t)$ and $y(t)$, respectively, lead to an algebraic loop which can be solved at every point in time, expect when there is a collision. This problem could be solved by using a different discretization approach for each of the integrals in Eq. (42) but that is uncommon, may violate the modularity of CBDs, and still causes an algebraic loop.

For the first part of the analysis, suppose that Fig. 1 is simulated with a classical continuous time simulator, which supports discontinuity location. Suppose that the collision occurs at time t_c. It follows that:

$$\begin{aligned} a(t_c) &= -2v(t_c)\frac{1}{h} - g \\ v(t_c+h) &= -v(t_c) - gh \end{aligned} \tag{44}$$

For a very small time step h, we get exactly what we wanted: the velocity of the ball is inverted in the collision.

No Dirac deltas where used in the previous example. Now we describe the same model, using Dirac deltas. Let t_c be the time of the first collision with the floor. Then the CBD model in Fig. 1 can be written as:

$$\begin{aligned} a(t) &= \frac{I(t)-mg}{m} \\ v(t) &= \int_0^t a(u)du + v(0) \\ y(t) &= \int_0^t v(u)du + y(0) \\ c(t) &= \begin{cases} 1 \text{ if } y(t) \leq 0 \\ 0 \text{ otherwise} \end{cases} \\ I(t) &= -2v(t)\delta(t-t_c) \end{aligned} \tag{45}$$

Noting that $c(t)' = 0 + \delta(t - t_c)$.

Following our implementation of the simulator with support for Dirac deltas, at the time arround the collision, the model in Eq. (45) is discretized to:

$$\begin{aligned} a(t_c) &= -2v(t_c)\delta - g \\ v(t_c) &= v(t_c - h) - mgh \\ v(t_c + h) &= v(t_c) + (-g)\,h - 2v(t_c)y(t_c) \quad &= y(t_c - h) + v(t_c - h)h \\ I(t_c) &= -2v(t_c)\delta \end{aligned} \tag{46}$$

Which is simplified to:

$$v(t_c + h) = -v(t_c) - gh \tag{47}$$

Exactly the same conclusion as Eq. (44).

This result is backed up by the numerical experiment, shown in Fig. 5. In the plot, three solutions are shown:
y_{ct} is the solution obtained via the classical CBD simulator;
y_{imp} is the solution obtained with the CBD simulator with support for impulses;
y_{analytic} is the analytic solution;

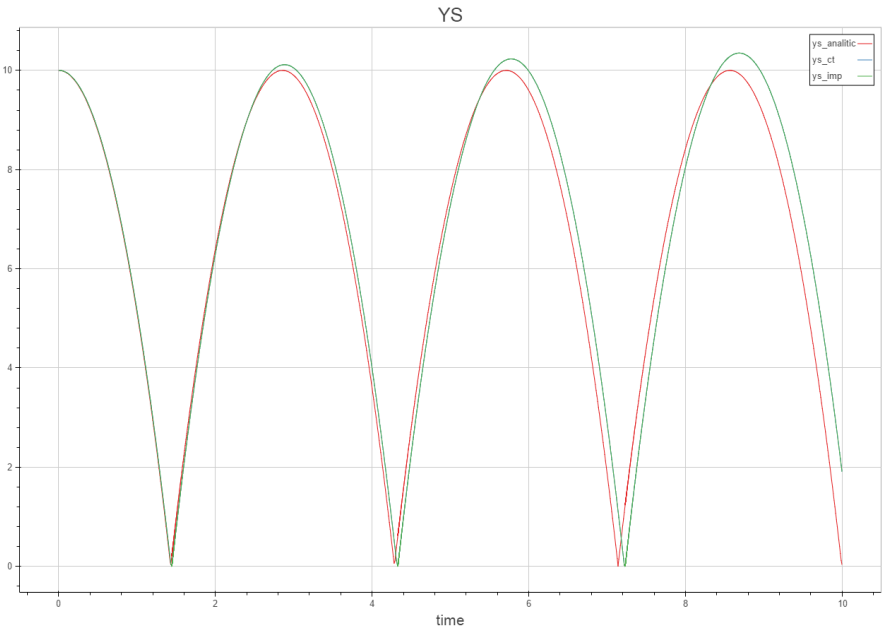

Figure 5: Trajectory of the position of the ball.

The solutions y_{ct} and y_{imp} coincide perfectly. This is confirmed by their norm, shown in Fig. 6.

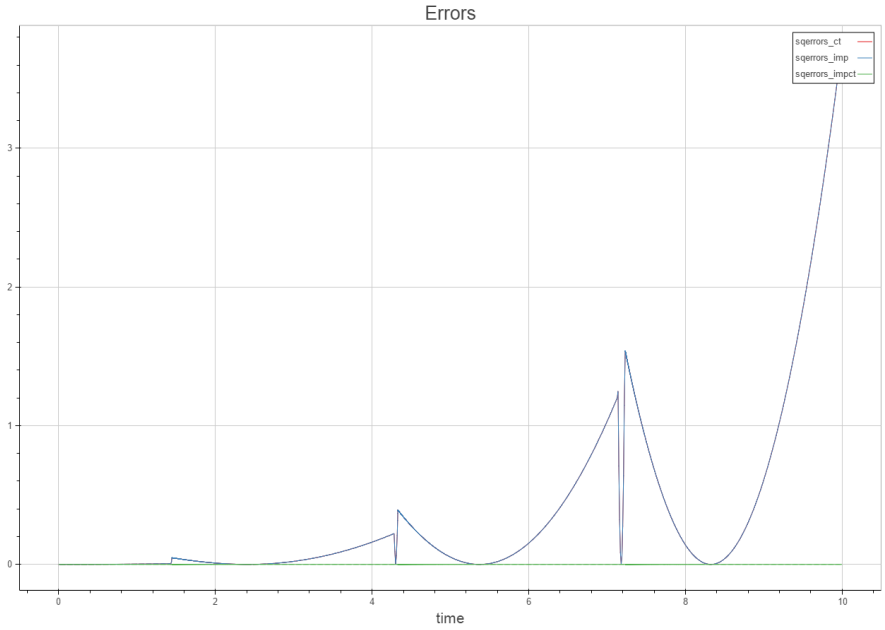

Figure 6: Instantaneous squared errors.

The benefit of the encoding of Dirac deltas in the simulator is that it minimizes the risk of overflows occurring. In the bouncing ball example, the value $-2v(t)\frac{c(t)-c(t-h)}{h}$ can be very big, whereas the version that supports Dirac deltas directly only stores the value $-2v(t)$.