HYBRID SYSTEM MODELLING AND SIMULATION WITH DIRAC DELTAS

Cláudio Gomes
Yentl Van Tendeloo

University of Antwerp
{firstname}.{lastname}@uantwerp.be

Joachim Denil
Paul De Meulenaere

University of Antwerp, Flanders Make
{firstname}.{lastname}@uantwerp.be

Hans Vangheluwe

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

ABSTRACT

For a wide variety of problems, creating detailed continuous models of physical systems is impractical. Hybrid models can abstract away short transient behaviour 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. 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 impulse. Current simulators cannot handle such approximations well due to the limitations of machine precision.

In this paper, we present two approaches for the simulation of impulses: symbolic and numerical. Our contribution is a theoretically founded description of both approaches in a Causal Block Diagram simulator, and an analysis of the conditions for which a symbolic approach is better than a numerical one.

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

1 INTRODUCTION

A model of a physical system has to pragmatically capture only the essential features for a task at hand (Cellier 1991). If the task is to analyze some behavior at a macro time scale, it is impractical – or even currently impossible (Stewart 2000) – to create high fidelity continuous models that also explain the behavior at micro time scales (we exclude quantum systems). Abstraction is the only way to deal with complexity and create useful models (Mosterman and Biswas 1998).

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 means of first order Ordinary Differential Equations (ODEs):

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

SpringSim-TMS/DEVS, 2017 April 23-26, Virginia Beach, VA, USA;

where $y^{(2)}$ denotes the acceleration of the ball (second derivative of the position), and y_0 and v_0 are the initial conditions of the dynamical model (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 Equation (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. Equation (1) is then split in two:

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

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 $0 \leq t < t_c$, 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 (Cellier and Kofman 2006).

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 (Dirac 1981) 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 Equation (4) and Equation (3). With this approach, Equation (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 (Lakshmikantham et al. 1989).

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 Equation (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 (Nilsson 2003) and more recently in (Lee et al. 2015), 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 (Nilsson 2003)) 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 (Nilsson 2003), 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 (Posse et al. 2002, Gomes et al. 2016), 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. Figure 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 Figure 1 b), c) and d).

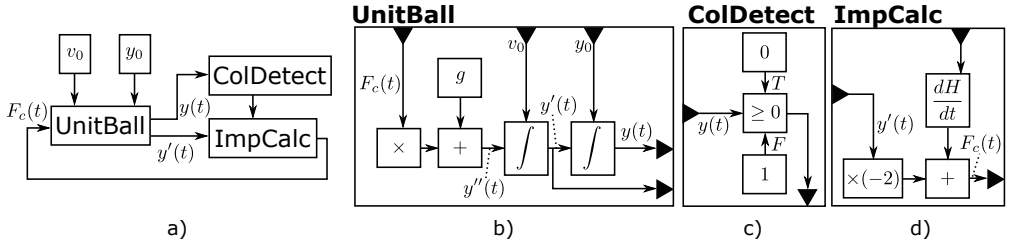

Figure 1: CBD model of a Bouncing Ball with unit mass (left), and specifications of each block (right).

The basic procedure for the simulation of a CBD model is show in Algorithm 1 (Mustafiz et al. 2016, Vangheluwe et al. 2014). For the sake of simplicity, this procedure assumes that there are no algebraic loops in the model (these are supported by the provided simulator though). In Figure 1 the integrator blocks only depend on their previous inputs, therefore no algebraic loop is formed.

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, where $h > 0$ is a constant input;

2.2 Distributions

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

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

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

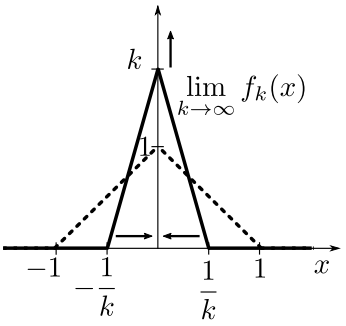

Figure 2: Limit approximation of a Dirac delta.

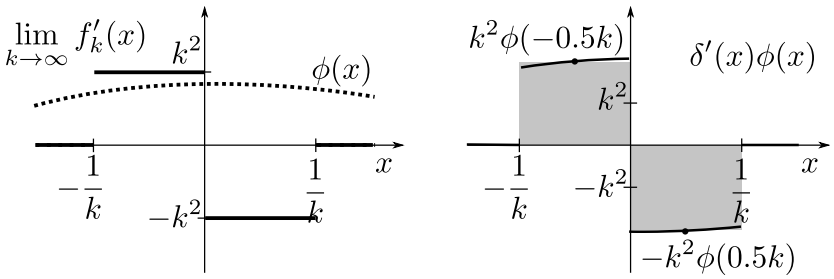

Figure 3: Intuitive visualization of Equation (9).

A Dirac delta $\delta(x)$ function can be constructed in a way that obeys to Equation (6) by taking the limit (Strichartz 2003), depicted in Figure 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 $\delta(x)\phi(x) = \delta(x)\phi(0)$. The observed behavior then becomes:

$$\int_{-\infty}^{\infty} \delta(x)\phi(x)dx = \int_{-\infty}^{\infty} \delta(x)\phi(0)dx = \phi(0)\int_{-\infty}^{\infty} \delta(x)dx = \phi(0) \tag{8}$$

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 Figure 3:

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

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 (Horvath 1970, Friedlander and Joshi 1998, Strichartz 2003), 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 Equation (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 Equation (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 Equation (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 Equation (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.

It is clear that derivatives of the solution are discontinuous functions, so differentiating $y(t)$ twice and considering the result as a distribution yields:

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

which, according to the distribution equality, obeys to Equation (5). In the next section we apply this theory for the derivation of the symbolic computations on impulses.

3 SYMBOLIC MANIPULATION OF DIRAC DELTAS

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, following the formalization in (Nilsson 2003), 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 the derivations in this section can disregard the left and right limits of the impulse part 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 Equation (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 Equation (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 Figure 4 a). We now provide a derivation of each operation. For more details, see (Gomes et al. 2017).

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

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

Operationally, it adds the two impulse-free signals $u(t)$ and $v(t)$, and adds the coefficients a_{ij} of the impulses in $U(t)$, with the coefficients b_{ij} of the impulses in $V(t)$. $Y(t)$ is of the form of Equation (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.

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:

$$\langle Y(t), \varphi(t) \rangle = \langle H(c(t)), \varphi(t) \rangle \tag{19}$$

where $H(x)$ is the Heaviside distribution defined in Equation (12).

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 Equation (17). Under these assumptions, the derivation is:

$$\langle Y(t), \varphi(t) \rangle = \langle U(t)H(C(t)) + V(t)(1 - H(C(t))), \varphi(t) \rangle \tag{20}$$

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

$$\langle Y(t), \varphi(t) \rangle = \langle U'(t), \varphi(t) \rangle = \langle u'(t), \varphi(t) \rangle + \left\langle \sum_{i=0}^{n_u} \sum_{\tau_j \in \{\tau_j\}} a_{ij} \delta^{(i+1)}(t - \tau_j)(t), \varphi(t) \right\rangle \tag{21}$$

The term $\langle u'(t), \varphi(t) \rangle$ must be further developed because it can have discontinuities. 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:

$$\begin{aligned}
\langle u'(t), \varphi(t) \rangle &= \langle u'(t^-), \varphi(t) \rangle + \left\langle \left[(u(t^+) - u(t^-))H(t - t_d) \right]', \varphi(t) \right\rangle \\
&= \langle u'(t^-), \varphi(t) \rangle - \langle (u(t^+) - u(t^-))H(t - t_d), \varphi'(t) \rangle = \langle u'(t^-), \varphi(t) \rangle - \int_{t_d}^{\infty} (u(t^+) - u(t^-))\varphi'(t)dt \\
&= \langle u'(t^-)(1 - H(t - t_d)) + u'(t^+)H(t - t_d) + (u(t_d^+) - u(t_d^-))\delta(t - t_d), \varphi(t) \rangle
\end{aligned} \tag{22}$$

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 Equation (22) is in the form of Equation (16). Joining Equation (22) and Equation (21) yields:

$$\langle Y(t), \varphi(t) \rangle = \left\langle u'(t) + \sum_{t_d \in \{t_d\}} (u(t_d^+) - u(t_d^-))\delta(t - t_d) + \sum_{i=0}^{n_u} \sum_{\tau_j \in \{\tau_j\}} a_{ij} \delta^{(i+1)}(t - \tau_j)(t), \varphi(t) \right\rangle \tag{23}$$

The **Integrator** block is derived as:

$$\langle Y(t), \varphi(t) \rangle = \left\langle \int_0^t U(x)dx, \varphi(t) \right\rangle = \left\langle \int_0^t u(x)dx + \sum_{\tau_j \in \{\tau_j\}} a_{0j} H(x - \tau_j) + \sum_{i=1}^{n_u} \sum_{\tau_j \in \{\tau_j\}} a_{ij} \delta^{(i-1)}(t - \tau_j), \varphi(t) \right\rangle \tag{24}$$

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 Equation (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 (this restriction cannot be enforced automatically by the tool), then it can be derived as:

$$\begin{aligned}\langle Y(t),\varphi(t)\rangle &= \langle U(t)V(t),\varphi(t)\rangle \\ &= \langle u(t)v(t),\varphi(t)\rangle + \sum_{i=0}^{n_v}\sum_{\tau_j\in\{\tau_j\}} a_{ij}\sum_{k=0}^{i}\binom{i}{k}u^{(k)}(\tau_j)(-1)^k\left\langle \delta^{(i-k)}(t-\tau_j),\varphi(t)\right\rangle\end{aligned} \tag{25}$$

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 Equation (9) and Figure 3, and note that

$$\left\langle \delta'(t-\tau_j)u(t),\varphi(t)\right\rangle = -\left[u(t-\tau_j)\varphi(t-\tau_j)\right]' = \left\langle u(t)\delta' - u'(t)\delta,\varphi(t)\right\rangle \tag{26}$$

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

$$\langle Y(t),\varphi(t)\rangle = \left\langle \frac{1}{u(t)},\varphi(t)\right\rangle \tag{27}$$

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:

$$Y_{\text{int}}(t) \approx Y_{\text{int}}(t-h) + U(t)\times h \quad \text{and} \quad Y_{\text{der}}(t) \approx \frac{U(t)-U(t-h)}{h} \tag{28}$$

The signals, in the form of Equation (16), can be encoded in many ways. For example, at each time step, 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. (Nilsson 2003) encodes the left limit of the signal, along with all its derivatives (lazily evaluated). Another example encoding can be seen in (Lee et al. 2015) that is based in the super dense time encoding. Figure 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 intuitive, especially with derivatives of impulses (recall Equation (25) and Equation (26)), 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.

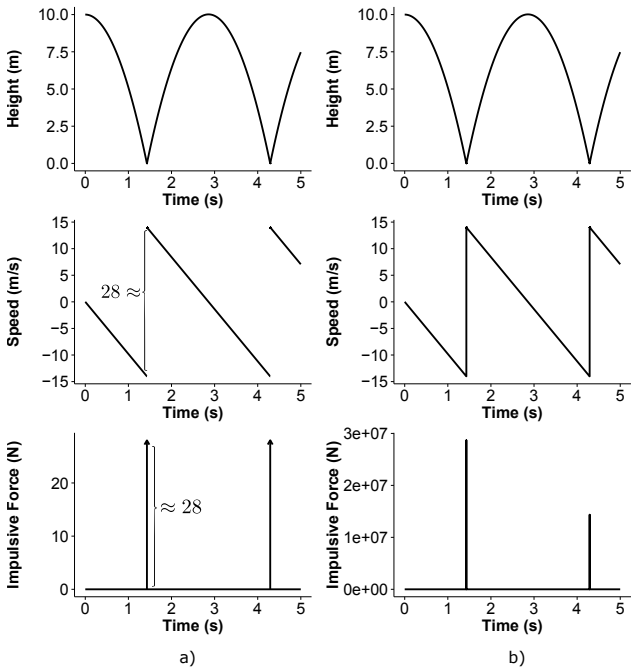

Figure 4: Bouncing ball trace, with perfectly elastic collision. The symbolic-impulse approach is a) and the approximated-impulse one is b).

A Dirac delta impulse is a arbitrary high function at a very small interval of time that obeys to Equation (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:

$$\delta(t-\tau_d) \approx \begin{cases} N \text{ if } 0 \leq t-\tau_d < h \\ 0 \text{ otherwise} \end{cases} \tag{29}$$

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

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

$$\delta(x) = \lim_{k\to\infty} H'_k(x) = \lim_{k\to\infty} \begin{cases} \frac{1}{2}k \text{ if } -\frac{1}{k} \leq x \leq \frac{1}{k} \\ 0 \text{ otherwise} \end{cases} \quad \text{with} \quad H_k(x) = \begin{cases} 0 & \text{if } x < -\frac{1}{k} \\ \frac{1}{2}+\frac{1}{2}kx & \text{if } -\frac{1}{k} \leq x \leq \frac{1}{k} \\ 1 & \text{if } x > \frac{1}{k} \end{cases} \tag{30}$$

where $H_k(x)$ represents a continuous approximation of the Heaviside function, for large enough k: This formulation satisfies the conditions in Equation (6), and it represents the limit version of the result in Equation (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 Equation (28). 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 Figure 4 b). Numerical approximations different than those in Equation (28) 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 (Zhang et al. 2008) by adjusting h and our simulator does this (both the symbolic and the numerical versions) which explains why Figure 4 b) has different values for the impulses.

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

$$S(t) = H(t-\tau_d)\ ;\ S'(t) = \delta(t-\tau_d)\ ;\ S^{(2)}(t) = \delta'(t-\tau_d)\ ;\ \ldots\ ;\ S^{(n)}(t) = \delta^{(n-1)}(t-\tau_d) \tag{31}$$

We compare this signal with the approximation $\tilde{S}(t) \approx H_{1/h}(t-\tau_d)$ from Equation (30), differentiated according to the approximation in Equation (28). 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 1 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 1: Approximated derivatives of $S(t)$ (Equation (31)).

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:

$$\binom{n-1}{k-1}/h^k \quad \text{where} \quad k = \mathit{floor}\left(\frac{n}{2}\right) \tag{32}$$

Equation (32) 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 $D\binom{n-1}{k-1}/h^k$. This result follows from Equation (32) 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. This result is shown in more detail in the technical report (Gomes et al. 2017). 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 Equation (28). 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 Equation (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. Equation (32) 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 Equation (32) 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 being able to identify when operations that are not defined in theory, are being computed (e.g., an inversion of an impulse). As (Nilsson 2003) 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

We are thankful to the anonymous reviewers for the corrections and comments provided. This research was partially supported by Flanders Make vzw, the strategic research centre for the manufacturing industry, and PhD fellowship grants from (1) the Agency for Innovation by Science and Technology in Flanders (IWT), and (2) Research Foundation - Flanders (FWO).

References

Cellier, F. E. 1991. *Continuous system modeling*. Springer Science & Business Media.

Cellier, F. E., and E. Kofman. 2006. *Continuous System Simulation*. Springer Science & Business Media.

Dirac, P. A. M. 1981. *The principles of quantum mechanics*. Number 27. Oxford university press.

Friedlander, F. G., and M. S. Joshi. 1998. *Introduction to the Theory of Distributions*. Cambridge University Press.

Gomes, C., J. Denil, and H. Vangheluwe. 2016. "Causal-Block Diagrams". Technical report, University of Antwerp.

Gomes, C., Y. Van Tendeloo, J. Denil, P. De Meulenaere, and H. Vangheluwe. 2017, feb. "Hybrid System Modelling and Simulation with Dirac Deltas". Technical report, University of Antwerp, Antwerp.

Horvath, J. 1970, mar. "An Introduction to Distributions". *The American Mathematical Monthly* vol. 77 (3), pp. 227.

Lakshmikantham, V., D. D. Bainov, and P. S. Simeonov. 1989. *Theory of impulsive differential equations*, Volume 6. World scientific.

Lee, E. A., M. Niknami, T. S. Nouidui, and M. Wetter. 2015. "Modeling and Simulating Cyber-physical Systems Using CyPhySim". In *Proceedings of the 12th International Conference on Embedded Software*, EMSOFT '15, pp. 115–124. Piscataway, NJ, USA, IEEE Press.

Mosterman, P. J., and G. Biswas. 1998, apr. "A theory of discontinuities in physical system models". *Journal of the Franklin Institute* vol. 335 (3), pp. 401–439.

Mustafiz, S., C. Gomes, B. Barroca, and H. Vangheluwe. 2016. "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, pp. 29:1—-29:8. San Diego, CA, USA.

Nilsson, H. 2003, sep. "Functional automatic differentiation with dirac impulses". *ACM SIGPLAN Notices* vol. 38 (9), pp. 153–164.

Posse, E., J. de Lara, and H. Vangheluwe. 2002. "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)*, pp. 23–34.

Stewart, D. 2000, jan. "Rigid-Body Dynamics with Friction and Impact". *SIAM Review* vol. 42 (1), pp. 3–39.

Strichartz, R. S. 2003. *A guide to distribution theory and Fourier transforms*. World Scientific.

Vangheluwe, H., J. Denil, S. Mustafiz, D. Riegelhaupt, and S. Van Mierlo. 2014. "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, Society for Computer Simulation International.

Zhang, F., M. Yeddanapudi, and P. Mosterman. 2008. "Zero-crossing location and detection algorithms for hybrid system simulation". In *IFAC World Congress*, pp. 7967–7972.

AUTHOR BIOGRAPHIES

CLÁUDIO GOMES is a PhD student in the University of Antwerp and member of the Modelling, Simulation and Design Lab (MSDL), interested in co-simulation of hybrid systems. His email address is Claudio.Gomes@uantwerp.be.

YENTL VAN TENDELOO is a PhD student in the University of Antwerp and member of MSDL. He is currently exploring the relationships between models and meta-models. His email address is Yentl.VanTendeloo@uantwerp.be.

JOACHIM DENIL is a Postdoctoral researcher in the University of Antwerp and member of the Modelling, Simulation and Design Lab (MSDL). His main research interest is multi-paradigm modelling of software-intensive systems. His email address is Joachim.Denil@uantwerp.be.

PAUL DE MEULENAERE is a Professor at the University of Antwerp. His main research field is model-based development methods for embedded real-time systems. This work is being conducted in the research group CoSys-Lab. His email address is Paul.DeMeulenaere@uantwerp.be.

HANS VANGHELUWE is a Professor at the University of Antwerp (Belgium), an Adjunct Professor at McGill University (Canada) and an Adjunct Professor at the National University of Defense Technology (NUDT) in Changsha, China. He heads the Modelling, Simulation and Design (MSDL) research lab. His research interest is the multi-paradigm modelling of complex, software-intensive, cyber-physical systems. His email address is Hans.Vangheluwe@uantwerp.be.