



HAL
open science

Compile-Time Impulse Analysis in Modelica

Albert Benveniste, Benoît Caillaud, Mathias Malandain

► **To cite this version:**

Albert Benveniste, Benoît Caillaud, Mathias Malandain. Compile-Time Impulse Analysis in Modelica. MODELICA 2021 - 14th International Modelica Conference, Sep 2021, Linköping, Sweden. pp.1-11. hal-03281394

HAL Id: hal-03281394

<https://inria.hal.science/hal-03281394>

Submitted on 8 Jul 2021

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution 4.0 International License

Compile-Time Impulse Analysis in Modelica

Albert Benveniste, Benoît Caillaud, Mathias Malandain¹

¹Inria Centre de Rennes Bretagne Atlantique, University of Rennes 1, France,
{albert.benveniste,benoit.caillaud,mathias.malandain}@inria.fr

Abstract

Since its 3.3 release, Modelica offers the possibility to specify models of dynamical systems with multiple modes having different DAE-based dynamics. However, the handling of mode changes by the current Modelica tools is not satisfactory. An important difficulty is the occurrence of impulsive behavior at some mode changes, for some variables. In this paper, we propose a compile-time algorithm for identifying such impulsive behaviors and quantifying them in terms of their magnitude orders. Such algorithm can be used as an additional step of the structural analysis of Modelica models.

Keywords: multimode DAE, structural analysis, impulsive behaviors

1 Introduction

Modelica and other languages supporting object-oriented modeling of physical systems rely on the formalism of DAEs. Compilers of such languages perform sophisticated preprocessing prior to generating simulation code (Casella, 2015). Index analysis and reduction (Mattsson and Söderlind, 1993) is one such important processing, where selected equations are differentiated one or more times until the Jacobian matrix with respect to the leading variables (i.e., the variables of maximal differentiation degree in the system) becomes structurally regular.

Since its 3.3 release, Modelica offers the possibility of specifying *multimode dynamics*, by describing state machines with different DAE dynamics in each different state (Elmqvist et al., 2012). This feature enables describing large complex cyber-physical systems with different behaviors in different modes.

While being very valuable, this possibility has been the source of serious difficulties for non-expert users. Although many large-scale complex Modelica models are properly handled, some physically meaningful models do not give rise to correct simulation results—it is actually not difficult to construct such problematic programs, thus, chances are significant to produce such bad cases in large models. In (Benveniste et al., 2020), we propose a structural analysis that is valid for multimode DAE models, both within each mode and at mode changes, illustrated in the companion paper (Benveniste et al., 2021).

One specific problem is due to the existence, in many physical models, of impulsive behaviors for some variables. With existing tools, such models give rise to simulations

collapsing at runtime. Impulsive behaviors are already a problem from a mathematical standpoint, as they do not fall within the existing concepts of solutions of a DAE system—the definition used in (Campbell and Gear, 1995) assumes smoothness of the trajectories.

To cope with this issue, *distributions* were considered by some authors. To our knowledge, the most comprehensive approach was provided by Stephan Trenn. In his PhD thesis (Trenn, 2009b) and his article (Trenn, 2009a), he pointed out the difficulty in defining piecewise smooth distributions: several mathematically coherent definitions of the “Dirac part” of such a distribution can be considered, so that it has no intrinsic definition. This indicates that distributions are not the ultimate answer to deal with impulsive variables in multimode DAE systems. Still, Trenn was able in (Liberzon and Trenn, 2012) to define complete solutions for a class of switched DAE systems in which each mode is in *quasi-linear form* and switching conditions are time-based, not state-based.

Another important step forward was done in (Benveniste et al., 2019). An interesting subclass of multimode DAE systems was identified, which possibly exhibit impulsive variables at mode changes. They extend the “quasi-linear systems” proposed by Trenn in the sense that switching conditions are no longer restricted to time-based ones, instead including state-based switching conditions. The analysis and discretization schemes proposed in (Benveniste et al., 2019) are mathematically sound. Building on this work, Martin Otter has developed the ModiaMath¹ tool for semi-linear multimode DAE systems. Since this work, this approach was refined and extended by the authors of this paper (Benveniste et al., 2020), and is illustrated on examples in (Benveniste et al., 2021).

Contribution of this paper: A complete structural analysis of multimode DAE systems was only recently proposed by the authors of this paper. In particular, this approach distinguishes between *long modes*, in which the dynamics is continuous-time and governed by a DAE system for a positive duration, and *transient modes*, which are zero duration events at which restarts can occur; note that, as a result, chattering behavior such as encountered when applying sliding mode control is not supported.

We develop here another important aspect of our approach, by focusing on impulsive behaviors. We explain

¹<https://modiasim.github.io/ModiaMath.jl/stable/man/Overview.html>

this aspect on the Cup-and-Ball example, a mild variation of the popular 2D pendulum in which the straight rod is replaced by a rope. When the rope gets straight, an impulse typically occurs for the tension if an idealized model is considered. To analyze this behavior, we propose a general compile-time analysis, acting as an additional step of the multimode structural analysis presented in the companion paper (Benveniste et al., 2021).

Since distributions fail to properly handle impulsive behaviors in general, our mathematical tool for this is *nonstandard analysis* (Robinson, 1996; Cutland, 1988; Lindstrøm, 1988), which allows for a correct use of infinities and infinitesimals in mathematical analysis. We use this setting in two ways:

- First, we discretize the DAE dynamics in each long mode using an explicit first-order Euler scheme with an *infinitesimal* time step ∂ ; this provides us with an approximation of the DAE solutions up to an infinitesimal error. Infinitesimal time steps are also used to capture restarts at mode changes: the values of states in the new mode are computed, from values before the change, in one or several infinitesimal time steps.
- Second, we compute *impulse orders*, i.e., orders of magnitude of algebraic variables at mode changes, for both long and transient modes, with reference to the infinitesimal time step ∂ ; for example, an order of $1/\partial$ for an algebraic variable indicates that this variable is impulsive.

We develop a compile-time calculus that evaluates the impulse order of every algebraic variable, thus revealing its impulsive/non-impulsive nature. Finite impulse orders can be used to renormalize impulsive variables when implementing a numerical scheme that approximates the restart values for each state variable of the system, thus improving conditioning.

In the next section, we investigate the Cup-and-Ball example, a two-mode variation of the celebrated pendulum in Cartesian coordinates. In Section 3, we develop the impulse analysis in its generality and explain how it can be mechanized.

2 The Cup-and-Ball example



Figure 1. The Cup-and-Ball game.

We sketch here a multimode extension of the popular example of the pendulum in Cartesian coordinates (Pantelides, 1988), namely the Cup-and-Ball game illustrated by Figure 1. A ball, modeled by a point mass, is attached to

one end of a rope, while the other end of the rope is fixed, to the origin of the plane in the model. The ball is subject to the unilateral constraint set by the rope, but moves freely while the distance between the ball and the origin is less than its actual length. The system is assumed closed and subject to no external interaction.

2.1 The model

The considered model of the two-dimensional Cup-and-Ball game is:

$$\begin{cases} 0 = x'' + \lambda x & (e_1) \\ 0 = y'' + \lambda y + g & (e_2) \\ 0 \leq L^2 - (x^2 + y^2) & (\kappa_1) \\ 0 \leq \lambda & (\kappa_2) \\ 0 = [L^2 - (x^2 + y^2)] \times \lambda & (\kappa_3) \end{cases} \quad (1)$$

where the *dependent variables* are the position (x, y) of the ball in Cartesian coordinates and the rope tension λ .

The subsystem $(\kappa_1, \kappa_2, \kappa_3)$ expresses that the distance of the ball from the origin is less than or equal to L , the tension is nonnegative, and one cannot have a nonzero tension and a distance less than L at the same time. This is known as a *complementarity condition*, written as $0 \leq L^2 - (x^2 + y^2) \perp \lambda \geq 0$ in the *nonsmooth systems* literature (Acary and Brogliato, 2008), and is an adequate modeling of ideal valves, diodes (Cellier and Kofman, 2006, Chapter 9.10), and contact in mechanics.

Note that, not only an impulsive behavior is expected on the torques, but an other possible difficulty is present, as subsystem $(\kappa_1, \kappa_2, \kappa_3)$ of (1) leaves the impact law at mode change insufficiently specified; it could be fully elastic, fully inelastic, or in between. *We expect both of these aspects to be detected at compile time, using some kind of structural analysis.*

However, before such a structural analysis is possible, some changes are required in the model. As a matter of fact, constraints κ_1 and κ_2 are unilateral, which is not supported by Modelica and related languages. Therefore, using the technique presented in (Mattsson et al., 1999), we redefine the graph of this complementarity condition as a parametric curve, represented by the following three equations:

$$\begin{aligned} s &= \text{if } \gamma \text{ then } -\lambda \text{ else } L^2 - (x^2 + y^2) \\ 0 &= \text{if } \gamma \text{ then } L^2 - (x^2 + y^2) \text{ else } \lambda \\ \gamma &= [s \leq 0] \end{aligned}$$

which allows us to rewrite model (1) as follows:

$$\begin{cases} 0 = x'' + \lambda x & (e_1) \\ 0 = y'' + \lambda y + g & (e_2) \\ \gamma = [s \leq 0] & (k_0) \\ \text{if } \gamma \text{ do } 0 = L^2 - (x^2 + y^2) & (k_1) \\ \text{and } 0 = \lambda + s & (k_2) \\ \text{if not } \gamma \text{ do } 0 = \lambda & (k_3) \\ \text{and } 0 = (L^2 - (x^2 + y^2)) - s & (k_4) \end{cases} \quad (2)$$

We then observe that the subsystem collecting equations $(k_0)-(k_4)$ is a logico-numerical fixpoint equation, with dependent variables x, y, λ, γ . A possible solution would

consist in performing a relaxation, by iteratively updating the numerical variables based on the previous value for the guards, and then re-evaluating the guard based on the updated values of the numerical variables, hoping for a fixpoint to occur. Such fixpoint equation, however, can have zero, one, several, or infinitely many solutions. No characterization exists that could serve as a basis for a (graph-based) structural analysis. *We thus decided to refuse solving such mixed logico-numerical systems.* As a consequence, we are unable to evaluate guard γ , so that the mode the system is in cannot be determined: model (2) is rejected.

To break the fixpoint equation defining γ , we choose to restrict ourselves to guards defined by left-limits; in this example, this yields $\gamma = [s^- \leq 0]$, where $s^-(t) =_{\text{def}} \lim_{u \nearrow t} s(u)$ (the modification is highlighted in red):

$$\left\{ \begin{array}{ll} 0 = x'' + \lambda x & (e_1) \\ 0 = y'' + \lambda y + g & (e_2) \\ \gamma = [s^- \leq 0] & (k_0) \\ \text{if } \gamma \text{ do } 0 = L^2 - (x^2 + y^2) & (k_1) \\ \quad \text{and } 0 = \lambda + s & (k_2) \\ \text{if not } \gamma \text{ do } 0 = \lambda & (k_3) \\ \quad \text{and } 0 = (L^2 - (x^2 + y^2)) - s & (k_4) \end{array} \right. \quad (3)$$

We are now ready to associate a structural analysis to model (3) that will be valid both in long modes with DAE dynamics, and at mode changes. To achieve this, we will replace derivatives by their corresponding forward Euler schemes, which will bring everything to a discrete progress of time (both continuous dynamics and mode changes).

To avoid introducing approximation errors, we will use an *infinitesimal* time step ∂ , which is made mathematically formal by relying on *nonstandard analysis*.

2.2 Using nonstandard analysis

Nonstandard analysis (Robinson, 1996; Lindström, 1988; Benveniste et al., 2012) extends the set \mathbb{R} of real numbers into a superset ${}^*\mathbb{R}$ of *hyperreals* (also called *nonstandard reals*) that includes infinite sets of infinitely large numbers and infinitely small numbers. Key properties of hyperreals, needed for the informal discussion of the Cup-and-Ball example, are the following:

There exist infinitesimals, defined as hyperreals that are smaller in absolute value than any real number: an infinitesimal $\partial \in {}^*\mathbb{R}$ is such that $|\partial| < a$ for any positive $a \in \mathbb{R}$. For x, y two hyperreals, write $x \approx y$ if $x - y$ is an infinitesimal.

All relations, operators, and propositional formulas that are valid over \mathbb{R} are also valid over ${}^*\mathbb{R}$. For example, ${}^*\mathbb{R}$ is a totally ordered set. The arithmetic operations $+$, \times , etc. can be lifted to ${}^*\mathbb{R}$. We say that a hyperreal x is *finite* if there exists some standard finite positive real number a such that $|x| < a$.

For every finite hyperreal $x \in {}^*\mathbb{R}$, there is a unique standard real number $\text{st}(x) \in \mathbb{R}$ such that $\text{st}(x) \approx x$, and $\text{st}(x)$ is called the *standard part* (or *standardization*) of x .

Standardizing more complex objects, such as functions or systems of equations, requires some care (see Theorem 1, Section 2.5).

Every real function lifts in a systematic way to a hyperreal function. This allows us to write $f(x)$ where f is a real function (regardless of its continuity properties) and x is a nonstandard number.

Continuity and derivatives. Let $t \mapsto x(t)$ be an \mathbb{R} -valued (standard) signal ($t \in \mathbb{R}$). Then:

$$x \text{ is continuous at instant } t \in \mathbb{R} \text{ if and only if, for any infinitesimal } \partial \in {}^*\mathbb{R}, \text{ one has } x(t + \partial) \approx x(t); \quad (4)$$

$$x \text{ is differentiable at instant } t \in \mathbb{R} \text{ if and only if there exists } a \in \mathbb{R} \text{ such that, for any infinitesimal } \partial \in {}^*\mathbb{R}, \frac{x(t + \partial) - x(t)}{\partial} \approx a. \text{ In this case, } a = x'(t). \quad (5)$$

We can then consider the time index set $\mathbb{T} \subseteq {}^*\mathbb{R}$:

$$\mathbb{T} = 0, \partial, 2\partial, 3\partial, \dots = \{n\partial \mid n \in {}^*\mathbb{N}\} \quad (6)$$

where ∂ is a positive infinitesimal, and ${}^*\mathbb{N}$ denotes the set of *hyperintegers*, consisting of all integers augmented with additional infinite numbers called *nonstandard*. The important features of \mathbb{T} are:

(1) Any finite real time $t \in \mathbb{R}_+$, where \mathbb{R}_+ denotes the set of nonnegative real numbers, is infinitesimally close to some element of \mathbb{T} (informally, \mathbb{T} covers \mathbb{R}_+ and can be used to index continuous-time dynamics); and

(2) \mathbb{T} is “discrete”: every instant $n\partial$ has a predecessor $(n-1)\partial$ (except for $n=0$) and a successor $(n+1)\partial$.

Let x be a nonstandard signal indexed by \mathbb{T} . We define the *forward-* and *backward-shifted* signals x^\bullet and ${}^\bullet x$ through

$$x^\bullet(n\partial) =_{\text{def}} x((n+1)\partial) \text{ and } {}^\bullet x((n+1)\partial) =_{\text{def}} x(n\partial),$$

implying that an initial value for ${}^\bullet x(0)$ must be provided. For f a function of the tuple X of signals, we set $(f(X))^\bullet =_{\text{def}} f(X^\bullet)$ where the forward shift $X \mapsto X^\bullet$ applies pointwise to all the components of the tuple. For example, $f^\bullet(x, y)(t) = f(x(t+\partial), y(t+\partial))$.

By (5), this allows us to represent, up to an infinitesimal, the derivative x' of a signal by its first-order explicit Euler approximation $\frac{1}{\partial}(x^\bullet - x)$. Solutions of multi-mode DAE systems may, however, be non-differentiable and even non-continuous at events of mode change. To give a meaning to x' at any instant, *we decide to define it everywhere as the nonstandard first-order Euler increment.*

Hence, we implicitly add to every system the following two equations, for each state variable x :

$$x' = \frac{x^\bullet - x}{\partial}; \quad x'' = \frac{x^{\bullet 2} - 2x^\bullet + x}{\partial^2}, \quad (7)$$

where

$$\begin{aligned} x^\bullet(t) &=_{\text{def}} x(t + \partial), \\ x^{\bullet 2}(t) &=_{\text{def}} x(t + 2\partial) \text{ and, generally} \\ x^{\bullet n}(t) &=_{\text{def}} x(t + n\partial). \end{aligned}$$

Equation (7) means that the derivatives x', y', x'', y'' are interpreted using the explicit first-order Euler scheme with an *infinitesimal time step* ∂ . Note that (7) implies

$$x'' = \frac{x' - x'}{\partial} . \quad (8)$$

This yields the nonstandard expansion of the corrected model (3):

$$\left\{ \begin{array}{ll} 0 = x'' + \lambda x & (e_1) \\ 0 = y'' + \lambda y + g & (e_2) \\ \gamma^\bullet = [s \leq 0]; \gamma(0) = F & (k_0) \\ \text{if } \gamma \text{ do } 0 = L^2 - (x^2 + y^2) & (k_1) \\ \quad \text{and } 0 = \lambda + s & (k_2) \\ \text{if not } \gamma \text{ do } 0 = \lambda & (k_3) \\ \quad \text{and } 0 = (L^2 - (x^2 + y^2)) - s & (k_4) \end{array} \right. \quad (9)$$

This model is understood in the nonstandard setting, meaning that the derivatives are expanded using (7). Therefore, the leading variables in all modes are $\lambda, s, x^{\bullet 2}, y^{\bullet 2}$.

We are ready to concentrate on structural analysis and we will focus on the main difficulty with this Cup-and-Ball model, namely the mode change $\gamma: F \rightarrow T$, when the rope gets straight. The reader is referred to the companion paper (Benveniste et al., 2021) for omitted details.

2.3 Structural analysis of mode change $\gamma: F \rightarrow T$

Due to equation (k_1) , the mode $\gamma = T$ (where the rope is straight) requires index reduction. We thus augment model (9) with the two latent equations shown in **red**:

$$\left\{ \begin{array}{ll} 0 = x'' + \lambda x & (e_1) \\ 0 = y'' + \lambda y + g & (e_2) \\ \gamma^\bullet = [s \leq 0]; \gamma(0) = F & (k_0) \\ \text{if } \gamma \text{ do } 0 = L^2 - (x^2 + y^2) & (k_1) \\ \quad \text{and } 0 = L^2 - (x^2 + y^2)^\bullet & (k_1^\bullet) \\ \quad \text{and } 0 = L^2 - (x^2 + y^2)^\bullet 2 & (k_1^{\bullet 2}) \\ \quad \text{and } 0 = \lambda + s & (k_2) \\ \text{if not } \gamma \text{ do } 0 = \lambda & (k_3) \\ \quad \text{and } 0 = (L^2 - (x^2 + y^2)) - s & (k_4) \end{array} \right. \quad (10)$$

Note that the two latent equations (k_1^\bullet) and $(k_1^{\bullet 2})$ were obtained by *shifting* (k_1) *forward*, not by differentiating it as usually performed—the two, however, are equivalent from the structural analysis standpoint, because of equalities (7).

To perform structural analysis at the considered mode change, we first unfold model (10) at the successive instants

$${}^{\bullet 2}t \stackrel{\text{def}}{=} t - 2\partial, \quad {}^\bullet t \stackrel{\text{def}}{=} t - \partial, \quad \text{and } t,$$

where t denotes the current instant. In the following, equation (e_1) at the instant $t - 2\partial$ (respectively, $t - \partial$) will be denoted by $({}^{\bullet 2}e_1)$ (resp., $({}^\bullet e_1)$).

In this unfolding, the two equations (k_1) and (k_1^\bullet) are in structural conflict with selected equations from the previous two instants, shown in **blue** in the following subsystem,

whose dependent variables are the leading variables at instants $t - 2\partial$ and $t - \partial$, namely $x, y, {}^{\bullet 2}\lambda; x^\bullet, y^\bullet, {}^\bullet\lambda$:

$$\left\{ \begin{array}{ll} 0 = \frac{x - 2x + {}^{\bullet 2}x}{\partial^2} + {}^{\bullet 2}\lambda \cdot {}^{\bullet 2}x & ({}^{\bullet 2}e_1) \\ 0 = \frac{y - 2y + {}^{\bullet 2}y}{\partial^2} + {}^{\bullet 2}\lambda \cdot {}^{\bullet 2}y + g & ({}^{\bullet 2}e_2) \\ 0 = \frac{x - 2x + x}{\partial^2} + {}^\bullet\lambda \cdot x & ({}^\bullet e_1) \\ 0 = \frac{y - 2y + y}{\partial^2} + {}^\bullet\lambda \cdot y + g & ({}^\bullet e_2) \\ 0 = L^2 - (x^2 + y^2) & (k_1) \\ 0 = L^2 - (x^2 + y^2)^\bullet & (k_1^\bullet) \end{array} \right.$$

This conflict can be detected from structural information only, using the Dulmage-Mendelsohn decomposition (Dulmage and Mendelsohn, 1958). We propose to resolve this conflict by applying the following causality principle:

Principle 1 (Causality) *What was done at the previous instant cannot be undone at the current instant.*

Applying Principle 1 leads to erasing, in model (10), equations (k_1) and (k_1^\bullet) at the instant of mode change ${}^\bullet\gamma = F, \gamma = T$. This yields the following system:

$$\text{at } \left[\begin{array}{l} {}^\bullet\gamma = F \\ \gamma = T \end{array} \right] : \left\{ \begin{array}{ll} 0 = x'' + \lambda x & (e_1) \\ 0 = y'' + \lambda y + g & (e_2) \\ 0 = L^2 - (x^2 + y^2)^\bullet 2 & (k_1^{\bullet 2}) \\ 0 = \lambda + s & (k_2) \end{array} \right. \quad (11)$$

It uniquely determines all the leading variables from the state variables x, y and x^\bullet, y^\bullet . In turn, equations (k_1) and (k_1^\bullet) , which were erased from this model, are not satisfied.

At the next instant, i.e., when ${}^{\bullet 2}\gamma = F, {}^\bullet\gamma = T, \gamma = T$, the same argument is used. We thus erase, in model (10), the only equation (k_1) at the next instant. This yields the following system:

$$\text{at } \left[\begin{array}{l} {}^{\bullet 2}\gamma = F \\ {}^\bullet\gamma = T \\ \gamma = T \end{array} \right] : \left\{ \begin{array}{ll} 0 = x'' + \lambda x & (e_1) \\ 0 = y'' + \lambda y + g & (e_2) \\ 0 = L^2 - (x^2 + y^2)^\bullet & (k_1^\bullet) \\ 0 = L^2 - (x^2 + y^2)^\bullet 2 & (k_1^{\bullet 2}) \\ 0 = \lambda + s & (k_2) \end{array} \right. \quad (12)$$

Note that (k_1^\bullet) is a consistency equation that is satisfied by the state variables x^\bullet, y^\bullet . In turn, equation (k_1) , which was erased from this model, is not satisfied. At subsequent instants, equation erasure is no longer needed: the process amounted to *delaying by a few nonstandard instants the satisfaction of some of the constraints in the new mode*, which actually took zero standard time. This completes the nonstandard structural analysis of the mode change $\gamma: F \rightarrow T$, i.e., when the rope gets straight.

2.4 Impulse analysis at mode change $\gamma: F \rightarrow T$

We now focus on identifying possible impulsive behaviors at this mode change. This is achieved by analyzing nonstandard systems (11) and (12) defining the values for restart. The intent is that the former will set the restart positions, whereas the latter will set the restart velocities.

Our impulse analysis not only identifies impulsive variables but also quantifies their order of magnitude, thanks to the following notion of impulse order:

Definition 1 (Impulse order and analysis)

1. Given a nonstandard system of equations E defining the values for restart, say that a dependent variable x has impulse order $\sigma \in \mathbb{R}$ in E , if the solution of system E is such that $x\partial^{-\sigma}$ is provably a finite non-zero (standard) real number. Let $\llbracket x \rrbracket$ denote the impulse order of x . By convention, the constant 0 has impulse order $-\infty$.
2. Say that x is impulsive if $\llbracket x \rrbracket > 0$.
3. The impulse analysis of a system of equations S is the system of constraints satisfied by the impulse orders of the dependent variables of S .

Impulse analysis relies on the following generic assumption, which expresses that DAE within long modes must be reinitialized with finite values for the state variables:

Assumption 1 *State variables are not impulsive; that is, for any state variable v , one has $\llbracket v \rrbracket \leq 0$.*

As an example, if, in the new mode, a variable x is differentiated up to order n , then its $(n - 1)$ -th derivative is a state variable and thus subject to Assumption 1. Consequently, its k -th order derivatives for $k = 0, \dots, n - 2$ are continuous at the considered mode change.

We are now ready to successively analyze System (11) and System (12).

System (11): The state variables are x, y, x', y' . By Assumption 1, we get the following prior information, which expresses that velocities are not impulsive:

$$\llbracket x^\bullet - x' \rrbracket \leq 0 \quad ; \quad \llbracket y^\bullet - y' \rrbracket \leq 0 \quad . \quad (13)$$

Conditions (13) imply that positions should be continuous. While performing our impulse analysis, we include equation (8) relating second derivatives and first derivatives. System (11) involves equation $(e_1) : x'' + \lambda x = 0$, which, by using (8), rewrites

$$x^\bullet - x' + \partial \lambda x = 0 \quad . \quad (14)$$

By (13), equation (14) implies $\llbracket \lambda \rrbracket \leq 1$. Exploiting all equations of System (11) yields the following information

$$\llbracket \lambda \rrbracket = \llbracket s \rrbracket \leq 1 \quad , \quad (15)$$

whereas other dependent variables have impulse order zero. System (12) is handled similarly, with the same conclusion. In Section 3, we mechanize the impulse analysis for an arbitrary restart system. Prior to doing this, we now explain how this impulse analysis can be exploited for generating effective code for restart.

2.5 Using impulse analysis in code generation

Code generation for restarts consists in standardizing non-standard systems (11) and (12). See the introduction of Section 2.2 for the meaning of “standardization”; note, however, that standardizing systems of equations requires more care than standardizing numbers, due to impulsive behaviors and singularity issues that result.

We can exploit the impulse analysis through the following three different approaches. The method of Section 2.5.1 is mostly described for didactic purposes, as it requires the symbolic elimination of variables, which can be very costly or even impossible in nonlinear systems. In practice, the methods of Sections 2.5.2 and 2.5.3 shall be used; both of these sections briefly address this topic.

2.5.1 Eliminating impulsive variables

When this is practical, the simplest method from a conceptual point of view is to eliminate impulsive variables from the restart system, as they are of no use for restarting the new mode.

We still focus here on the standardization of the mode change $\gamma : F \rightarrow T$, i.e., when the rope gets straight. Our task is to standardize systems (11) and (12), by targeting discrete-time dynamics, for the two successive instants composing the restart phase. This will provide us with restart values for positions and velocities.

By (15), tensions λ and λ^\bullet are both candidates to be impulsive, hence so are s and s^\bullet by (k_2, k_2^\bullet) . We eliminate the impulsive variables by ignoring (k_2, k_2^\bullet) , combining (e_1) and (e_2) to eliminate λ , and (e_1^\bullet) and (e_2^\bullet) to eliminate λ^\bullet . This yields:

$$\text{at } \begin{bmatrix} \bullet \gamma = F \\ \gamma = T \end{bmatrix} : \begin{cases} 0 = y''x + gx - x''y \\ 0 = L^2 - (x^2 + y^2)^\bullet 2 \end{cases} \quad (16)$$

$$\text{at } \begin{bmatrix} \bullet 2 \gamma = F \\ \bullet \gamma = T \\ \gamma = T \end{bmatrix} : \begin{cases} 0 = y''x + gx - x''y \\ 0 = L^2 - (x^2 + y^2)^\bullet \\ 0 = L^2 - (x^2 + y^2)^\bullet 2 \end{cases} \quad (17)$$

In System (16), we expand second derivatives using (7), whereas, in System (17), we expand them using (8). Consequently, System (16) has dependent variables $x^{\bullet 2}, y^{\bullet 2}$, whereas System (17) has dependent variables x^\bullet, y^\bullet . We are now ready to standardize the two systems.

System (16) to define restart positions: We expand second derivatives using (7):

$$\begin{cases} 0 = (y^{\bullet 2} - 2y^\bullet + y)x - (x^{\bullet 2} - 2x^\bullet + x)y + \partial^2 gx \\ 0 = L^2 - (x^2 + y^2)^\bullet 2 \end{cases} \quad (18)$$

Setting $\partial = 0$ in System (18) yields a structurally regular system, so that we can invoke the following result, proved in (Benveniste et al., 2020):

Theorem 1 (standardizing systems of equations) *For $\mathbf{H} : \mathbb{R}^{n+1} \rightarrow \mathbb{R}^n$ a \mathcal{C}^1 (standard) function, consider the nonstandard system of equations $\mathbf{H}(\partial, X) = 0$ where X is a n -vector of variables. If system $\mathbf{H}(0, X) = 0$ is structurally*

nonsingular, then setting $\partial = 0$ in system $\mathbf{H}(\partial, X) = 0$ yields the correct standardization of it, meaning that the solution $x_*(\partial)$ of $\mathbf{H}(\partial, X) = 0$ standardizes as the solution x_* of $\mathbf{H}(0, X) = 0$.

By this theorem, setting $\partial = 0$ in System (18) yields the correct standardization of it:

$$\begin{cases} 0 = (y^{\bullet 2} - 2y^\bullet + y)x - (x^{\bullet 2} - 2x^\bullet + x)y \\ 0 = L^2 - (x^2 + y^2)^{\bullet 2} \end{cases}$$

Then, in the resulting system, we interpret x and x^\bullet as the left-limit x^- of state variable x in previous mode, and $x^{\bullet 2}$ as the restart value x^+ for the new mode. This yields

$$\begin{cases} 0 = (y^+ - y^-)x^- - (x^+ - x^-)y^- \\ 0 = L^2 - (x^2 + y^2)^+ \end{cases} \quad (19)$$

which determines the restart values for positions. Note that the constraint that the rope is straight is satisfied. Furthermore, as $0 = L^2 - (x^2 + y^2)^-$ also holds (the rope is straight at the mode change), $x^+ = x^-, y^+ = y^-$ is the unique solution of (19): positions are continuous.

System (17) to define restart velocities: We expand second derivatives using (8):

$$\begin{cases} 0 = (y^\bullet - y')x - (x^\bullet - x')y + \partial \cdot gx \\ 0 = L^2 - (x^2 + y^2)^\bullet \\ 0 = L^2 - (x^2 + y^2)^{\bullet 2} \end{cases} \quad (20)$$

By expanding $x^{\bullet 2} = x^\bullet + \partial x^\bullet$, the right-hand side of the last equation rewrites

$$\begin{aligned} L^2 - (x^2 + y^2)^{\bullet 2} &= L^2 - (x^2 + y^2)^\bullet \\ &\quad + 2\partial(x^\bullet x'^\bullet + y^\bullet y'^\bullet) \\ &\quad + \partial^2((x'^\bullet)^2 + (y'^\bullet)^2) \\ &= 0 \text{ (using (20))} \\ &\quad + 2\partial(x^\bullet x'^\bullet + y^\bullet y'^\bullet) \\ &\quad + O(\partial^2) \end{aligned} \quad (21)$$

Using this expansion of $L^2 - (x^2 + y^2)^{\bullet 2}$, setting $\partial = 0$ in (20) yields

$$\begin{cases} 0 = (y'^\bullet - y')x - (x'^\bullet - x')y \\ 0 = x^\bullet x'^\bullet + y^\bullet y'^\bullet \end{cases} \quad (22)$$

where the dependent variables are now x'^\bullet, y'^\bullet , whereas other variables are state variables whose values are determined by previous time steps. Note that System (22) is structurally regular, so that we can invoke Theorem 1, showing that System (22) is the correct standardization of System (20). We are now ready to get effective code for the restart. In System (22), we perform the following substitutions, where superscripts $-$ and $+$ denote left- and right-limits, and the continuity of positions is used:

$$x = x^- ; x^\bullet = x^+ \quad \text{and} \quad x' = x'^- ; x'^\bullet = x'^+ \quad (23)$$

and similarly for y . This finally yields

$$\begin{cases} 0 = (y'^+ - y'^-)x^- - (x'^+ - x'^-)y^- \\ 0 = x^+ x'^+ + y^+ y'^+ \end{cases} \quad (24)$$

System (24) determines x'^+ and y'^+ , which are the velocities for restart. The second equation guarantees that the velocity will be tangent to the constraint. With (19) and (24), we determine the restart conditions for positions and velocities. Invariants from the physics are satisfied.

This is a satisfactory solution when the elimination of impulsive variables is practical. In our example, they entered linearly in the restart system, so that elimination was straightforward. When this is not the case, elimination becomes costly or even impossible. Moreover, generalizing and mechanizing this elimination process appears to be a very difficult task. We thus need to look for alternatives for computing the velocities for restart.

2.5.2 Rescaling impulsive variables

Focus again on System (12). Impulse analysis told us that λ, s both have impulse order ≤ 1 . We thus rescale them accordingly:

$$\hat{\lambda} =_{\text{def}} \partial^1 \times \lambda \quad \text{and} \quad \hat{s} =_{\text{def}} \partial^1 \times s \quad (25)$$

Using this rescaling together with expansion (8), System (12) rewrites

$$\begin{cases} 0 = x'^\bullet - x' + \hat{\lambda}x & (e_1) \\ 0 = y'^\bullet - y' + \hat{\lambda}y + \partial g & (e_2) \\ 0 = L^2 - (x^2 + y^2)^\bullet & (k_1^\bullet) \\ 0 = L^2 - (x^2 + y^2)^{\bullet 2} & (k_1^{\bullet 2}) \\ 0 = \hat{\lambda} + \hat{s} & (k_2) \end{cases} \quad (26)$$

In System (26), (k_1^\bullet) is a consistency equation satisfied as a result of performing System (11) at the previous instant. We can also discard equation (k_2) , which only serves to determine the auxiliary variable s . Thus, we are left with the sub-system collecting equations $(e_1), (e_2), (k_1^{\bullet 2})$. We can again expand the right-hand side of $(k_1^{\bullet 2})$ by using (21). In the resulting system, we can safely set $\partial \leftarrow 0$ since it yields the following structurally regular system:

$$\begin{cases} 0 = x'^+ - x'^- + \hat{\lambda}x^- & (e_1) \\ 0 = y'^+ - y'^- + \hat{\lambda}y^- & (e_2) \\ 0 = 0 = x^+ x'^+ + y^+ y'^+ & (k_1^{\bullet 2}) \end{cases} \quad (27)$$

System 27 determines $x'^+ = x'^-, y'^+ = y'^-$, and the rescaled impulsive tension $\hat{\lambda}$, as functions of state variables x', y', x, y , which were identified with the left-limits of velocities and positions at previous mode. Note that eliminating the rescaled tension $\hat{\lambda}$ from System (27) yields System (24).

Rescaling impulsive variables is simpler than eliminating them. This method is also promising in terms of designing and implementing algorithms for its mechanization, as

the computation of the impulse orders amounts to finding a minimal solution to a system of linear unilateral constraints. Unfortunately, it does not work in full generality since impulse orders can be infinite, as the following example shows:

$$x = \exp(y/\partial),$$

where y is known to have impulse order zero. Indeed, the impulse order of $(y/\partial)^n$ is n . Since the exponential expands as a power series of infinite support, we deduce that the impulse order of $\exp(y/\partial)$ is the maximum of all impulse orders of $(y/\partial)^n$, hence it is infinite. Thus, impulsive variable x cannot be rescaled.

The last method addresses such cases, at the price of a possibly poor numerical conditioning.

2.5.3 Brute-force solving of the restart system

When none of the above methods apply, it is still possible to solve system (26) with $\partial = \delta$ (a small positive time step) for the original variables λ and s , without rescaling them.

Then, it is proved in (Benveniste et al., 2020), see also (Benveniste et al., 2021) that *solving these systems for their dependent variables and then discarding the values found for the impulsive variables yields a converging approximation for the states and velocities at restart*. Moreover, first numerical experiments on toy examples showed no issue as long as the time step δ was kept reasonably high. Of course, without rescaling, the numerical conditioning is likely to be less favorable, so that rescaling is recommended when impulse orders are finite. Works are in progress for the implementation of this method, coupled with the rescaling of impulsive variables of finite order.

2.6 Handling transient modes: elastic impact

Our reasoning so far produces a behavior in which the two modes (free motion and straight rope) gently alternate; the system always stays in one mode for some positive period of time before switching to the other mode.

This indeed amounts to assuming that the impact is totally inelastic at mode change, an assumption that was not explicit at all in (9). So, what happened? In fact, *the straight rope mode was implicitly assumed to last for at least three nonstandard successive instants, since we allowed ourselves to shift (k_1) forward twice.*

Now, let us instead assume elastic impact, represented by the cascade of mode changes $\gamma: F \rightarrow T \rightarrow F$, reflecting that the straight rope mode is *transient* (it is left immediately after being reached).

We address transient modes in (Benveniste et al., 2020), see also (Benveniste et al., 2021). We show that a structural analysis for elastic impact can still be proposed, by suitably adapting the notion of *differentiation array* proposed by Campbell and Gear (Campbell and Gear, 1995). The so obtained structural analysis proves that our original model (1) for the Cup-and-Ball is underspecified at mode change $\gamma: F \rightarrow T$, when the rope gets straight. This underdetermination implies that the model is ill-defined, as it admits an infinite number of solutions. Completing it by adding an

impact law, which makes sense from a physicist's point of view, is also appropriate from the point of view of our structural analysis.

One possible choice is to complete the model with an elastic impact law. This indeed corrects the restart system at $\gamma = T$ in the cascade of mode changes $\gamma: F \rightarrow T \rightarrow F$, yielding

$$\begin{cases} 0 = x'' + \lambda x & (e_1) \\ 0 = y'' + \lambda y + g & (e_2) \\ 0 = y'^{\bullet} + (1 - \alpha)y' & (\tau_1) \\ 0 = L^2 - (x^2 + y^2) & (k_1) \\ 0 = \lambda + s & (k_2) \end{cases} \quad (28)$$

where $0 < \alpha < 1$ is a damping factor. We proceed again with the structural analysis. Variables x, y are the states, so that their values are set by the previous instants. Current equation (k_1) creates a conflict with the past. Hence, we discard it from System (28), which leaves us with the following system:

$$\begin{cases} 0 = x'' + \lambda x & (e_1) \\ 0 = y'' + \lambda y + g & (e_2) \\ 0 = y'^{\bullet} + (1 - \alpha)y' & (\tau_1) \\ 0 = \lambda + s & (k_2) \end{cases} \quad (29)$$

Model (29) is structurally nonsingular, recalling that y'' and y'^{\bullet} can be interchanged for the structural analysis. This refined model is therefore accepted by the structural analysis.

The impulse analysis proceeds as for the previous case of inelastic impact and effective code for restart can be generated.

Note that any impact law could be used instead of the one added in System (28), as long as it ensures uniqueness of the solution for a fixed state before the impact.

In (Benveniste et al., 2021), we also discuss the consequences, for modeling languages such as Modelica, of the need for stating as a side specification whether a mode is transient or not.

3 General Impulse Analysis

In this section, we explain how the reasoning used for the Cup-and-Ball example can be mechanized as a compilation stage following multimode structural analysis. Prior to developing this, we provide a simplified overview of said multimode structural analysis.

3.1 Overview of multimode structural analysis

We consider multimode DAE systems possessing *long modes* (having DAE-based dynamics for a positive duration) alternating with finite cascades of *transient modes* (having a zero duration, such as the straight rope mode in the Cup-and-Ball model with elastic impact).

We assume that the information regarding the type of a mode (long vs. transient) is known by the compiler—the two different Modelica primitives `if` and `when` should be used to declare long and transient modes, respectively.

In addition, we require that the current mode is defined by the left-limits of some predicates, see the reasoning leading to the corrected model (9) for the Cup-and-Ball.

For such models, the structural analysis proceeds according to the following steps:

1. The multimode model is mapped to its nonstandard expansion by using a first-order explicit Euler expansion for derivatives, with infinitesimal time step ∂ , and mapping left-limits to values at the previous instant. In particular, the mode at each nonstandard instant is known at the end of the previous instant.
2. The structural analysis for each specific mode is performed, depending on its long/transient type:
 - If the mode is long, then classical structural analysis applies: by, e.g., using Pryce’s Σ -method (Pryce, 2001), latent equations are added for the DAE system associated to each long mode;
 - Alternatively, if the mode is transient, a structural analysis of the *difference array* associated to the considered cascade of transient modes is performed.
3. Having done this, given the mode at the current instant:
 - If no mode change occurs, then the (classical) mode-specific structural analysis applies;
 - Otherwise, the conflict that may possibly exist between consistency equations of the current mode and leading equations of the previous mode is analyzed, using the Dulmage-Mendelsohn decomposition; conflicting subsystems are identified and the equations from the current instant that cause conflicts are erased.

Implementing the multimode structural analysis in the above described form would be very inefficient. Fortunately, (Caillaud et al., 2020) proposed a very efficient algorithm for handling all the long modes simultaneously without enumerating them, and extended the Σ -method in this “all-modes-at-once” framework. A similar extension of the Dulmage-Mendelsohn decomposition is being implemented.

3.2 General Rules of Impulse Analysis

3.2.1 Problem setting

Restart systems of equations, as resulting from the structural analysis at mode changes, are nonstandard systems of equations of the following generic form:

$$\text{expand } X' \text{ as } \frac{X^\bullet - X}{\partial} \text{ in } 0 = \mathbf{H}(X', X^\bullet, V, X) \quad (30)$$

where V collects the algebraic variables, X collects the state variables, and $\frac{X^\bullet - X}{\partial}$ is the nonstandard semantics

of X' . $\mathbf{H}(\cdot, \cdot, \cdot, \cdot)$, seen as a vector function in its dotted arguments, is by itself standard, since the equations of system $0 = \mathbf{H}$ are obtained by shifting or differentiating equations specified by the user. The reason for (30) being nonstandard is indeed twofold:

1. Since X^\bullet is involved, the infinitesimal ∂ occurs in time; and
2. Since X' is involved, the infinitesimal ∂ occurs both in time and *space*, due to the expansion $X' \leftarrow \frac{X^\bullet - X}{\partial}$.

The occurrence of ∂ in time is not an issue: shifted state variables will correspond to restart values for states, whereas non-shifted ones correspond to values prior to the change. In contrast, the occurrence of ∂ in space is the root cause of possible impulsive behaviors. Identifying them is the subject of impulse analysis.

3.2.2 The rules of impulse analysis

We now develop the *impulse analysis* introduced in Definition 1. This analysis is useful as a postprocessing of structural analysis, prior to generating effective code for restarts. Note that Assumption 1 is still enforced in what follows.

Figures 2 and 3 display the rules defining the translation of a system of equations of the form (30) into its impulse analysis, for the restricted class where only rational expressions are involved.

Figure 2 describes the syntax of a mini-language specifying such systems of equations. The left column of Figure 3 gives the rules for mapping expressions to their corresponding impulse orders. The reason for the inequality in (R6) is that in the sum $e_1 + e_2$, the dominant terms in the expansion of e_i as a series over ∂ may cancel each other. For an example of this, see equation (e_2) in System (12): rewriting this equation as $-g = y'' + \lambda y$, we see a case of strict inequality for (R6) since gravity g has order zero, whereas it is equal to the difference of two terms of order one.

We will use Rule (R6) in the following way, thereby reinforcing it. Consider an equation

$$e : z = x + y .$$

We can rewrite e in the following equivalent ways: $0 = x + y - z$, $x = z - y$, or $y = z - x$. To each of them we apply the max rule. This yields the following system of constraints, called the *impulse analysis of equation e* :

$$\begin{cases} \llbracket z \rrbracket \leq \max\{\llbracket x \rrbracket, \llbracket y \rrbracket\} \\ \llbracket 0 \rrbracket \leq \max\{\llbracket x \rrbracket, \llbracket y \rrbracket, \llbracket z \rrbracket\} \\ \llbracket x \rrbracket \leq \max\{\llbracket z \rrbracket, \llbracket y \rrbracket\} \\ \llbracket y \rrbracket \leq \max\{\llbracket x \rrbracket, \llbracket z \rrbracket\} \end{cases} \quad (31)$$

Note that the constraint $\llbracket 0 \rrbracket \leq \dots$ is vacuously satisfied since $\llbracket 0 \rrbracket = -\infty$. Then, among the three nontrivial inequalities of (31), at least two of them must be saturated. We will use impulse analysis (31) for handling sums of terms. This reinforcement of the max rule is formalized by Rule

$$\begin{aligned}
e & ::= 0 \mid c \mid \partial \mid x \mid e^c \mid e+e \mid e \times e \\
E & ::= e = e \mid E \text{ and } E
\end{aligned}$$

Figure 2. Syntax: E is a system of one or several equations $e = e$. An expression e is 0, a nonzero (standard) real constant c , the infinitesimal ∂ , a variable x , the monomial e^c , a sum, or a product.

$$\begin{array}{ll}
\text{(R1)} & \llbracket 0 \rrbracket = -\infty \\
\text{(R2)} & \llbracket c \rrbracket = 0 \\
\text{(R3)} & \llbracket \partial \rrbracket = -1 \\
\text{(R4)} & \llbracket e^c \rrbracket = c \llbracket e \rrbracket \\
\text{(R5)} & \llbracket e_1 \times e_2 \rrbracket = \llbracket e_1 \rrbracket + \llbracket e_2 \rrbracket \\
\text{(R6)} & \llbracket e_1 + e_2 \rrbracket \leq \max\{\llbracket e_1 \rrbracket, \llbracket e_2 \rrbracket\}
\end{array}
\qquad
\begin{array}{l}
\frac{E \vdash e = e'}{\llbracket E \rrbracket \vdash \llbracket e \rrbracket = \llbracket e' \rrbracket} \quad \text{(R7)} \\
\frac{E \vdash x = y + e \text{ or } E \vdash 0 = y - x + e}{E \vdash E \text{ and } y = x - e} \quad \text{(R8)}
\end{array}$$

Figure 3. Rules: The left column displays the impulse order of the primitive expressions. Rule (R7) indicates that $\llbracket e \rrbracket = \llbracket e' \rrbracket$ is an equation of the impulse analysis $\llbracket E \rrbracket$ if $e = e'$ is an equation of E ; rule (R8) indicates that, if E involves the equation $x = y + e$ but not the equation $y = x - e$, then we augment E with the latter, i.e., we saturate E with the rule $x = y + e \implies y = x - e$.

(R8) of Figure 3, which mechanizes the association, to any equation, of its different rewritings.

Using the rules of Figures 2 and 3 in the numerical expressions, we map any system of rational equations of the form (30) into a system of constraints over impulse orders.

To cover functions beyond polynomials, we need to extend $\mathbb{R} \cup \{-\infty\}$ with $+\infty$. In this extension, we take the convention that $-\infty + \infty = -\infty$, justified by both Rules (R1,R5) and the equality $0 \times x = 0$ for any nonstandard x . For functions $f(x) = \sum_{k=0}^{\infty} a_k x^k$ that can be represented as absolutely converging power series, we then get

$$\llbracket f(x) \rrbracket = \llbracket \sum_{k=0}^{\infty} a_k x^k \rrbracket = \llbracket x \rrbracket \cdot \text{sup}(A) \text{ ,} \quad (32)$$

where $A = \{k \mid a_k \neq 0\}$ is the support of the series and $\text{sup}(A)$ is the supremum of set A . In particular, if $\llbracket x \rrbracket > 0$ and if the support of the series is infinite, we get $\llbracket f(x) \rrbracket = +\infty$.

3.2.3 Particularizing the impulse analysis to systems of equations for restarts

So far, Rules (R1)–(R8) of the impulse analysis apply to any system of nonstandard equations. Here we particularize the impulse analysis to systems of equations of the form (30), where the only reason for ∂ to occur is the expansion of derivatives using the Euler scheme:

$$0 = \mathbf{H} \left(\frac{X^\bullet - X}{\partial}, X^\bullet, V, X \right)$$

The dependent variables are X^\bullet, V . It will be convenient to introduce the auxiliary variables

$$U =_{\text{def}} X^\bullet - X \text{ ,}$$

so that the systems we consider take the following form, where X^\bullet, V, U are the dependent variables:

$$\begin{cases} 0 & = \mathbf{H} \left(\frac{U}{\partial}, X^\bullet, V, X \right) \\ U & = X^\bullet - X \end{cases} \quad (33)$$

The following condition for System (33) can be assumed, based on physical considerations (restart values for an ODE or a DAE cannot be impulsive):

Assumption 2 *Since X is a state, both X (a known value) and X^\bullet must be finite.*

First, the impulse orders $\llbracket X \rrbracket$ are all known, from previous nonstandard instants. Next, from Assumption 2 we deduce the inequalities:

$$\llbracket X^\bullet \rrbracket \leq 0 \text{ and } \llbracket U \rrbracket \leq 0. \quad (34)$$

The impulse orders $\llbracket V \rrbracket$ are a priori unknown. We have, however, more prior information, thanks to the structural analysis. From the structural analysis at the considered mode change, we know which consistency equations of the new mode were conflicting with the dynamics of previous mode. Formally, call $G=0$ the subsystem collecting all the equations that were erased while solving this conflict—for the Cup-and-Ball model (10), at the instant of mode change $\bullet\gamma=F, \gamma=T$, G collects the bodies of the two violated consistency constraints (k_1) and (k_1^\bullet).

As a result, $G=0$ no longer holds at the considered mode change, and thus, G defines a tuple R of variables (one per entry of G) called *residuals*, by setting

$$R = G, \quad (35)$$

which are all finite and nonzero. An example of residual in the Cup-and-Ball is $r = L - (x^2 + y^2)$, which is both finite and nonzero at mode change $\bullet\gamma=F, \gamma=T$. The residuals are found by the structural analysis.

Finally, the system of equations that we need to solve collects all the above items, namely:

$$\begin{cases} 0 & = \mathbf{H} \left(\frac{U}{\partial}, X^\bullet, V, X \right) \\ U & = X^\bullet - X \\ R & = G \end{cases} \quad (36)$$

with dependent variables X^\bullet, V, U, R , and the following prior information on impulse orders is known:

$$\llbracket \frac{1}{\sigma} \rrbracket = 1; \llbracket X^\bullet \rrbracket \leq 0; \llbracket U \rrbracket \leq 0; \llbracket R \rrbracket = 0. \quad (37)$$

System (36) is then mapped to its impulse analysis by using Rules (R1–R8) of Figures 2 and 3. A suitable constraint solver is then used to solve the resulting set of constraints on impulse orders, by using side information (37). The choice of an appropriate constraint solver remains to be done.

4 Conclusion

The correct handling of truly multimode Modelica models (in which index and structure may vary with the mode) requires significant add-ons to the existing structural analyses. In the companion paper (Benveniste et al., 2021), we introduce, by means of two small but representative examples, a truly multimode structural analysis that applies both in modes and at mode changes. One important difficulty is the correct handling of impulsive behaviors for some variables.

In this paper, we introduced the *impulse analysis* of multimode DAE systems, a complement to multimode structural analysis for Modelica models. Impulse analysis is performed at compile time, prior to generating simulation code. It allows to identify impulsive variables, along with the mode changes at which impulsive behavior occurs. When impulsive behaviors occur in a model, then the conditions for restart at the impulsive mode change are generally known implicitly, not explicitly. Generating simulation code for restarts can thus be problematic. Using our approach based on impulse analysis, impulsive variables can be properly rescaled, so that correct explicit code for restarts can be generated.

In this paper, we did not consider the computational cost of performing true multimode structural analysis at compile time: unfortunately, the number of modes tends to be roughly exponential in the size of the model, and the *a priori* number of mode changes is at least proportional to the square of the number of modes. This is a limitation of a model representation in which one characterizes the subset of equations and variables active in any given mode.

A possible way of alleviating this issue is by shifting to a dual representation, that provides predicates characterizing the set of modes in which each equation and each variable is active. In practice, not only does this approach lead to a much more compact representation, but it also allows for the design of efficient structural analysis methods for multimode DAE systems, working in an ‘all-modes-at-once’ fashion. Such a method was implemented in the IsamDAE tool, and first results are reported in (Caillaud et al., 2020). The examples coming with this tool already include thermodynamical, electrical and pneumatic models. Although only the structural analysis of long modes is currently performed, the implementation of the structural analysis of mode changes is in progress.

Acknowledgements

The authors are indebted to several colleagues and friends for valuable discussions: Hilding Elmqvist and Martin Otter, John Pryce, and Vincent Acary. Khalil Ghorbal contributed to the first version of this approach.

This work was supported by the FUI ModeliScale DOS0066450/00 French national grant (2018-2021) and the Inria IPL ModeliScale large scale initiative (2017-2021, <https://team.inria.fr/modeliscale/>).

References

- Vincent Acary and Bernard Brogliato. *Numerical Methods for Nonsmooth Dynamical Systems. Applications in Mechanics and Electronics*, volume 35 of *Lecture Notes in Applied and Computational Mechanics*. Springer-Verlag, 2008. ISBN 978-90-481-9680-7.
- Albert Benveniste, Timothy Bourke, Benoît Caillaud, and Marc Pouzet. Nonstandard semantics of hybrid systems modelers. *J. Comput. Syst. Sci.*, 78(3):877–910, 2012.
- Albert Benveniste, Benoît Caillaud, Hilding Elmqvist, Khalil Ghorbal, Martin Otter, and Marc Pouzet. Multi-mode DAE models - challenges, theory and implementation. In Bernhard Steffen and Gerhard J. Woeginger, editors, *Computing and Software Science - State of the Art and Perspectives*, volume 10000 of *Lecture Notes in Computer Science*, pages 283–310. Springer, 2019. ISBN 978-3-319-91907-2. doi:10.1007/978-3-319-91908-9_16.
- Albert Benveniste, Benoît Caillaud, and Mathias Malandain. The mathematical foundations of physical systems modeling languages. *Annual Reviews in Control*, 50:72–118, 2020. ISSN 1367-5788. doi:10.1016/j.arcontrol.2020.08.001.
- Albert Benveniste, Benoît Caillaud, and Mathias Malandain. Handling Multimode Models and Mode Changes in Modelica. In *Proceedings of the 14th International Modelica Conference*. Linköping University Electronic Press, September 2021.
- Benoît Caillaud, Mathias Malandain, and Joan Thibault. Implicit structural analysis of multimode DAE systems. In *23rd ACM International Conference on Hybrid Systems: Computation and Control (HSCC 2020)*, Sydney, Australia, April 2020. doi:10.1145/3365365.3382201.
- Stephen L. Campbell and C. William Gear. The index of general nonlinear DAEs. *Numer. Math.*, 72:173–196, 1995.
- Francesco Casella. Simulation of Large-Scale Models in Modelica: State of the Art and Future Perspectives. In Hilding Elmqvist and Peter Fritzson, editors, *Proc. of the Int. Modelica Conference*, Versailles, France, September 2015. Modelica Association.
- François Cellier and Ernesto Kofman. *Continuous System Simulation*. Springer, 2006. ISBN 9780387261027.
- Nigel Cutland. *Nonstandard analysis and its applications*. Cambridge Univ. Press, 1988.
- Andrew L. Dulmage and Nathan S. Mendelsohn. Coverings of bipartite graphs. *Canadian Journal of Mathematics*, 10: 517–534, 1958. doi:10.4153/CJM-1958-052-0.

- Hilding Elmqvist, Fabien Gaucher, Sven Erik Mattsson, and Dupont Francois. State machines in Modelica. In Martin Otter and Dirk Zimmer, editors, *Proc. of the Int. Modelica Conference*, pages 37–46, Munich, Germany, September 2012. Modelica Association.
- Daniel Liberzon and Stephan Trenn. Switched nonlinear differential algebraic equations: Solution theory, Lyapunov functions, and stability. *Automatica*, 48(5):954–963, 2012. doi:10.1016/j.automatica.2012.02.041.
- Tom Lindstrøm. An invitation to nonstandard analysis. In N.J. Cutland, editor, *Nonstandard Analysis and its Applications*, pages 1–105. Cambridge Univ. Press, 1988.
- Sven Erik Mattsson and Gustaf Söderlind. Index reduction in Differential-Algebraic Equations using dummy derivatives. *Siam J. Sci. Comput.*, 14(3):677–692, 1993.
- Sven Erik Mattsson, Martin Otter, and Hilding Elmqvist. Modelica Hybrid Modeling and Efficient Simulation. In IEEE, editor, *38th IEEE Conference on Decision and Control*, pages 3502–3507, 1999.
- Constantinos C. Pantelides. The consistent initialization of differential-algebraic systems. *SIAM J. Sci. Stat. Comput.*, 9(2):213–231, 1988.
- John D. Pryce. A simple structural analysis method for DAEs. *BIT*, 41(2):364–394, 2001.
- Abraham Robinson. *Nonstandard Analysis*. Princeton Landmarks in Mathematics, 1996. ISBN 0-691-04490-2.
- Stephan Trenn. Regularity of distributional differential algebraic equations. *MCSS*, 21(3):229–264, 2009a. doi:10.1007/s00498-009-0045-4.
- Stephan Trenn. *Distributional Differential Algebraic Equations*. PhD thesis, Technischen Universität Ilmenau, 2009b.