Structural Methods for handling mode changes in multimode DAE systems

Albert Benveniste[1], Benoît Caillaud[1], Yahao Chen[1], Khalil Ghorbal[1] and Mathias Malandain[1]

***Abstract*— We present an approach to handle mode changes in nonlinear DAE systems with multiple modes having different dynamics. Mode changes can be time and/or state based. A symbolic *rescaling analysis* is proposed that automatically generates the conditions for the hot restart. By being *structural*, our method fits multiphysics modeling languages such as Modelica, where simulation code is generated based on the structure of the model, not the actual value of its parameters.**

I. Introduction

Since the pioneering work by Hilding Elmqvist [9], Differential Algebraic Equation (DAE)-based modeling has been strongly advocated for modeling based on first principles of physics. This led to the creation of the Modelica community in the 1990s, with the language of the same name being now industrially used for physical modeling. Since its release 3.3,[1] Modelica offers the possibility to specify hybrid DAE models, also known as *multimode* models: the model can have states in which the DAE dynamics differs. The mathematical form for such models is a collection of finitely many equations of the form

$$\texttt{if}\ b\ \texttt{then}\ f(\text{the } x_i\text{'s and their derivatives}) = 0\,, \qquad (1)$$

where $x_i, i = 1, \ldots, n$ are (time-indexed) numerical variables, f is a sufficiently differentiable numerical function, and b is a Boolean expression of predicates over system variables and time. Boolean b guards equation $f = 0$, meaning that this equation is enabled when b evaluates to true, disabled otherwise. Multimode DAE models exhibit two major difficulties:

1) Mathematical definitions exist for the solution of a multimode DAE only for specific subclasses of physics [4], not in general;
2) Modes can be numerous—the number of modes typically grows exponentially with the number of subsystems.

Consequently, existing tools do not implement mode and mode change-aware compilation. Some models, although clearly valid, fail to get correctly simulated. Such pathological models are by no means exceptional, nor are they difficult to exhibit [3]. The second point was adressed in [6]. This work addresses the first difficulty by:

1) Giving a mathematical definition of mode change events in multimode DAEs;
2) Proposing an effective algorithm computing the hot restart.

[1]Centre Inria de l'Univ. Rennes, France; albert.benveniste@inria.fr

[1]The most recent release to date is 4.1.0.

Related work: An extensive review of the literature on multimode DAEs is available in [1]. The reader is referred to this paper for a general review. In this introduction, we restrict ourselves to the works more closely related to ours. Key references for our work are the important contributions by Stephan Trenn and coworkers [18], [19], [12], [11], [8]. In [18], the inconsistent DAE initialization problem is investigated: a regular linear DAE $Ex'=Ax+f$ is considered for $t\geqslant 0$, with an inconsistent initialization $x_{(-\infty,0)}$, meaning that the left limit $\lim_{s\nearrow 0} x(s)$ does not satisfy the consistency constraints set by the DAE. This possibly causes impulsive behavior for some variables at instant $t = 0$. *Piecewise-smooth distributions* were proposed in [18, Chapter 2] to give an explicit solution for the inconsistent initialization of a linear DAE, by which the behavior of the impulsive variables is described as a linear combination of Dirac derivatives. The *quasi-Weierstrass form* for a linear DAE is used: by constructing a state basis and recombining equations, the impulsive order (differentiation degree of the Dirac function δ) of each variable can be identified, and pure constraints are separated from the ODE part. An interesting step forward is [11], in which the inconsistent initialization problem for DAE $Ex'=Ax+g(x)+f$ is investigated, where g is a smooth nonlinear function, under the assumption that the nonlinear part $g(x)$ remains "foreign to impulsions" [11]—the reason being that, for x impulsive and g nonlinear, $g(x)$ is undefined in general. An appropriate time-invariant change of coordinates and equations allows the authors to algebraically decompose the DAE system into three subsystems: 1) a nonlinear ODE, 2) a nonlinear algebraic constraint, and 3) a linear DAE carrying the impulsive part of the system. Using this decomposition, the authors propose an explicit solution to the inconsistent initialization problem. These results comply with the linear control systems vision: states are internal and are only defined up to a change of basis.

Our objectives and approach: Our work differs from the works of Trenn *et al.* in a number of aspects. A first difference is our overall objective, namely: *to provide mathematical soundness for the compilation of DAE-based modeling languages such as Modelica.* In Modelica models, parameter values are entered right before launching a simulation, i.e., after the simulation code was compiled. Therefore, compilation involves the structure of the model, not the actual values of its parameters. This applies, in particular, to *index reduction* and *structural analysis,* which transforms the DAE into an ODE-like system by suitably differentiating selected equations. The technique used in Modelica tools to perform index reduction uses the model structure, not the

numerical details. Our work complies with this:

1) *Changes of state basis are prohibited*[2] — such changes are little relevant in the context of physical modeling (unlike in black-box modeling).
2) Whereas, in classical linear dynamical systems theory, regularity (of matrices or linear pencils) plays a central role, *we give up numerical regularity and replace it with structural regularity* (also called structural non-singularity).

A square matrix is *structurally nonsingular* if all of its diagonal entries can be made nonzero by pre- and post-multiplying it by permutation matrices. An algebraic system of nonlinear equations is *structurally nonsingular* if so is its Jacobian at its solution. This is checked on the incidence graph of the system, which is the bipartite graph having variables x and equations f as vertices, and an edge (f, x) iff variable x occurs in equation f: **the system of equations is structurally nonsingular iff its incidence graph possesses a *perfect matching*, i.e., a subset of the incidence graph that defines a bijection between equations and variables.** Structural nonsingularity is checked on the incidence graph of the system (capturing which equation involves which variable). Structural methods are used in high-performance computing, and in all the compilers of DAE-based modeling tools [5]. By not relying on parameter values, they allow us to identify, prior to simulation, if the model is over-, under-, or well-determined, a useful information for model design.

The second difference is that we do not describe the full trajectories of all the variables (particularly impulsion events). Instead, *we detect impulsive behaviors, and rescale impulsive variables to make them non-impulsive.* Our rescaling algorithm bears similarities with J. Pryce's Σ-method for the structural analysis of DAEs [17]. Altogether, the method and algorithms proposed in this paper deeply differ from [11].

Based on the above approach, our contributions consist of: a general definition of *hot restart* and a procedure to generate it automatically; a system of *rescaling equations*; associated theorems and effective algorithms. Our procedure generates the system of equations by which restart values for the states are determined from the values just before the mode change. Uniqueness of the restart system of equations is guaranteed, when it exists, and its non-existence typically corresponds to a lack of determinism in specifying the hot restart.

In our previous work [1], a restricted class of multimode DAEs, possibly involving impulsive behaviors, was considered (since then known as *semi-linear* multimode DAEs [2]). For this subclass, two alternative approaches were investigated and shown to be equivalent: a Gear-Gupta-Leimkuhler method [10], implemented in the Julia package Modia, and a novel approach relying on nonstandard analysis [13]. The latter approach is the root of the one presented in this paper. In our previous work [2], the problem was addressed by building on top of nonstandard analysis [13]. The resulting algorithm, however, was not completely specified and was difficult to analyze.

The method introduced in the present paper fixes both issues. Most importantly, it handles general nonlinear systems; mode changes can be state-based, not only time-based. Hot restart succeeds only if impulsive variables are involved *linearly* in the system model—this mirrors the restriction in [11], where it was formulated in Def. 6 as a complex "condition (G_p)". In our case, there is no need for a separate check: our algorithm discovers by itself if this condition is satisfied or violated. Despite being physics-agnostic, our method is able to reveal hidden physical invariants: in our cup-and-ball running example, the hot restart we generate preserves angular momentum, although no such law was explicitly stated in the model.

In this paper, we only consider mode changes between two successive "long modes" (lasting for a positive duration with the same DAE dynamics). The case of finite cascades of successive transient modes (of zero duration) will be addressed elsewhere.

Due to length constraints, we are unable to both explain our approach on an example, and fully develop our approach. We chosed to detail a cup-and-ball example, and then we draw how the approach used properly generalizes.

II. A CUP-AND-BALL EXAMPLE

This example illustrates the main challenges to be addressed: mode changes are state-based, and they involve impulsive behaviors. Major tools fail to simulate this model.

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 length. A model for a 2D version of this example is:



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

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

Subsystem $(\kappa_1, \kappa_2, \kappa_3)$ expresses that the tension is non-negative, the distance between the ball and the origin is less than or equal to L, and one cannot have a nonzero tension and a distance less than L at the same time: (2) is a *complementarity system* [16]. This model is not of the form (1) yet because of unilateral constraints (κ_1, κ_2). Using a technique presented in [14], we rewrite $(\kappa_1, \kappa_2, \kappa_3)$ as:

$$\begin{cases} \gamma &= [s \leqslant 0] \\ 0 &= \texttt{if}\ \gamma\ \texttt{then}\ L^2 - (x^2+y^2)\ \texttt{else}\ \lambda \\ s &= \texttt{if}\ \gamma\ \texttt{then} - \lambda\ \texttt{else}\ L^2 - (x^2+y^2) \end{cases} \quad (3)$$

To break the fixpoint equation defining γ, we choose to base guards on left-limits of signals. This yields the modified model (4), where the modification is highlighted in red. For

[2] This prohibits the construction of the above mentioned quasi-Weierstrass form.

convenience, we also grouped the equations that are active in mode $\gamma = \text{T}$ (resp., $\gamma = \text{F}$).

$$\left\{\begin{array}{rll} & 0 = x'' + \lambda x & (e_1)\\ & 0 = y'' + \lambda y + g & (e_2)\\ & \gamma = [s^- \leqslant 0]; \gamma(0) = \text{F} & (k_0)\\ \texttt{if } \gamma \texttt{ then} & 0 = L^2 - (x^2+y^2) & (k_1)\\ \texttt{and} & 0 = \lambda + s & (k_2)\\ \texttt{if not } \gamma \texttt{ then} & 0 = \lambda & (k_3)\\ \texttt{and} & 0 = (L^2-(x^2+y^2)) - s & (k_4) \end{array}\right. \tag{4}$$

Let us assume that the ball is initially under free motion ($\gamma = \text{F}$). Eventually, the rope will get straight (γ : F→T), with two possible continuations, depending on whether the impact is assumed *elastic* (the ball bounces inward when the rope gets straight) or *inelastic* (the rope remains straight). If the impact is inelastic, then the radial velocity of the ball becomes zero, hence the rope gets straight ($\gamma = \text{T}$) until the gravity pushes the ball back to free motion ($\gamma = \text{F}$). If the impact is elastic, then the radial velocity changes its sign (the simplest model is that it gets opposite and equal in magnitude) and the ball bounces back to free motion ($\gamma = \text{F}$); in this case, the ball spends zero time in $\gamma = \text{T}$ mode, we call it *transient*. We focus here on the inelastic case, where only long modes occur. An impulsive behavior is expected at the mode change from free motion to straight rope, γ:F→T. We illustrate our method on this mode change.

We first consider each mode separately in (4). Under the mode $\gamma = \text{F}$, free motion occurs, which yields an ODE. In contrast, under $\gamma = \text{T}$, the rope is straight, thus enabling the algebraic constraint (k_1): this yields a DAE. To make it ODE-like, *index reduction* is performed [7], [15], which consists here in adding to the model the two *latent equations* (k_1', k_1'') obtained by successive differentiations of (k_1). The resulting model splits into two parts [15]:

$$\begin{array}{rcl} \text{leading equations} & : & \left\{\begin{array}{ll} 0 = x'' + \lambda x & (e_1)\\ 0 = y'' + \lambda y & (e_2)\\ 0 = xx''+x'^2+y'^2+yy'' & (k_1'') \end{array}\right.\\ \text{consistency equations} & : & \left\{\begin{array}{ll} 0 = L^2-(x^2+y^2) & (k_1)\\ 0 = xx' + yy' & (k_1') \end{array}\right. \end{array} \tag{5}$$

In (5) and in the sequel, we highlight in blue an injective *matching* of equations to variables. Indeed, the leading equations generically determine the leading variables x'', y'', λ, while the consistency equation constraints the state variables x, y, x', y' at initialization, leaving two degrees of freedom.

The subsystem collecting the leading equations is called the *index reduced* version of the original system.

***Approach**:* Focus on mode change γ : F→T, and let t be the instant when it occurs. We wish to identify how the previous mode γ=F influences the restart of the new mode γ=T. Our approach unfolds as follows.

Approach 1: In a neighborhood of the mode change:

1) Each derivative is discretized using a simple forward Euler scheme with time step ε, i.e., any derivative x' is *identified* with $\varepsilon^{-1}(x(t+\varepsilon)-x(t))$; we call the added equation $x' = \varepsilon^{-1}(x(t+\varepsilon)-x(t))$ an *Euler identity*;
2) Using this interpretation of derivatives, we stack finitely many successive transitions of the resulting discrete-time system around the mode change, thus building an array of equations called *mode change array*; after deleting certain equations of it, the identification and compensation of impulsive behaviors is performed on this array, leading to its *rescaling*;
3) Finally, we let $\varepsilon := 0$ in the rescaled array.[3] The resulting *restart system* relates the latest state variables of this system, to the right limits of the trajectories of the previous mode; it is used for hot restart. ♦

To be properly used for the hot restart of the new mode, this restart system should satisfy the following requirements:

Requirement 1 (for the restart system):

1) It should be *deterministic,* i.e., it should uniquely determine the restart state variables of the new mode;
2) The restart state variables should satisfy the consistency conditions of the new mode.

Requirement 1.1 means that the mode change is determined. Requirement 1.2 expresses that a hot restart should at least satisfy the conditions for being a consistent start.

***Structural analysis**:* Performing Step 1 of Approach 1 discretizes the dynamics of the clutch in a neighborhood of the mode change. We zoom on it and we detail Step 2. To simplify the writing, we use the following notation where instant t is a reference: for any variable x, define

$$^\bullet x \equiv x(t-\varepsilon)\,,\ x \equiv x(t) \text{ and } x^\bullet \equiv x(t+\varepsilon)\,.$$

Using this notation, we display in Fig. 1 the dynamics of the new mode at instant t. We call A_0 a *mode change array* and regard it as a static system of equations, meaning that x and x' are considered independent variables. Variables x, y, x', y' are fully determined by the previous mode, due to the Euler identities $x = {}^\bullet x + \varepsilon \times {}^\bullet x'$ and $x' = {}^\bullet x' + \varepsilon \times {}^\bullet x''$ (and similarly for y): we call them *past variables*. The dependent variables of A_0 are x'', y'', λ (past variables are excluded).

The equation (k_1) is highlighted in green to indicate that it is satisfied up to $O(\varepsilon)$, despite it involving only past variables; the reason for this is that the mode change was detected at instant ${}^\bullet t$ by the *zero-crossing* event $0 = {}^\bullet(x^2+y^2-L^2)$, hence, $x^2+y^2-L^2 = O(\varepsilon)$. We call *fact* an equation that involves past variables only and is satisfied up to an $O(\varepsilon)$. Being nearly satisfied and involving no dependent variable, ***facts can be ignored***.[4]

This is in contrast to the next equation (k_1'): it also involves only past variables but it is not satisfied. *We decide to disable it at this instant*, which amounts to postponing its consideration for a while. We are left with the three equations (e_1, e_2, k_1''), which determine the leading variables x'', y'', λ. Hence, array A_0 uniquely determines the leading variables from the values of past variables: Requirement 1.1 is met.

[3] Throughout this paper, symbol := means the assignment of a value to a variable or parameter.

[4] The reader is kindly advised against inferring anything about the human world from this statement.

$$A_0 : \left\{ \begin{array}{lll} 0 = x'' + \lambda x & (e_1) & \gamma{=}\mathrm{T}, t \\ 0 = y'' + \lambda y + g & (e_2) & \\ 0 = L^2{-}(x^2{+}y^2) & (k_1) & \text{fact} \\ 0 = xx'{+}yy' & (k_1') & \text{disabled} \\ 0 = xx''{+}x'^2{+}y'^2{+}yy'' & (k_1'') & \end{array} \right.$$

Fig. 1. **Cup-and-ball example:** Mode change array A_0. In the last column we point the facts and the disabled conflicting equations; black equations are enabled. Triple (e_1, e_2, k_1'') is structurally nonsingular, with a one-to-one matching $\mathcal{M} = \{(e_1, x''), (e_2, \lambda), (k_1'', y'')\}$ between enabled equations and variables.

$$A_1 : \left\{ \begin{array}{lll} 0 = x'' + \lambda x & (e_1) & \text{instant } t \\ 0 = y'' + \lambda y + g & (e_2) & \\ 0 = L^2{-}(x^2{+}y^2) & (k_1) & \text{fact} \\ 0 = xx'{+}yy' & (k_1') & \text{disabled} \\ 0 = xx''{+}x'^2{+}y'^2{+}yy'' & (k_1'') & \text{disabled} \\ 0 = x'' - \varepsilon^{-1}(x'^{\bullet} - x') & (E^{)}_{x''}, & \\ 0 = y'' - \varepsilon^{-1}(y'^{\bullet} - y') & (E^{)}_{y''}, & \\ 0 = (x'' + \lambda x)^{\bullet} & (e_1^{\bullet}) & \text{instant } t{+}\varepsilon \\ 0 = (y'' + \lambda y + g)^{\bullet} & (e_2^{\bullet}) & \\ 0 = (L^2{-}(x^2{+}y^2))^{\bullet} & (k_1^{\bullet}) & \text{fact} \\ 0 = (xx'{+}yy')^{\bullet} & (k_1'^{\bullet}) & \\ 0 = (xx''{+}x'^2{+}y'^2{+}yy'')^{\bullet} & (k_1''^{\bullet}) & \end{array} \right.$$

Fig. 2. **Cup-and-ball example:** Mode change array A_1. Facts and disabled conflicting equations are pointed. Euler identities $(E^{)}_{x''}, (E^{)}_{y''}$, relating second derivatives to shifts of first derivatives, were added. The enabled subsystem (in black) is structurally nonsingular, with a one-to-one matching $\mathcal{M}$ between equations and variables, highlighted in blue.

$$A_1 : \left\{ \begin{array}{ll} 0 = x'' + \lambda x & (f_1) \\ 0 = y'' + \lambda y + g & (f_2) \\ 0 = (xx'{+}yy')^{\bullet} & (f_3) \\ 0 = x'' - \varepsilon^{-1}(x'^{\bullet} - x') & (f_4) \\ 0 = y'' - \varepsilon^{-1}(y'^{\bullet} - y') & (f_5) \end{array} \right.$$

Fig. 3. **Cup-and-ball example:** The black subsystem of Fig. 2, in which leading equations of instant $t^{\bullet} = t{+}\varepsilon$ were omitted, and equations renumbered for convenience.

In contrast, Requirement 1.2 is not met since consistency equation (k_1') was violated.

Let us extend array A_0 by one more instant, by considering array A_1, shown in Fig. 2. Euler identities $(E^{)}_{x''}$ and $(E^{)}_{y''}$ were added, to make explicit the relations between x'' and $x'^{\bullet}$, and y'' and $y'^{\bullet}$. Variables x, y, x', y' are still determined by the previous mode. Equation (k_1) remains a fact, and so does $(k_1^{\bullet})$, for the same reasons. Let us disable the conflicting equations (k_1', k_1''). The remaining system is structurally non-singular, as evidenced by the perfect matching highlighted in blue. Array A_1 determines the values of all its dependent variables knowing the past variables: Requirement 1.1 is met. In addition, Requirement 1.2 is now satisfied since all the consistency equations at instant $t^{\bullet}$ are enabled. We solved the hot restart of the new mode, in discretized setting, i.e. Step 2 of Approach 1 was successful when using A_1.

Rescaling analysis: What happens when performing Step 3, i.e., when letting $\varepsilon := 0$? We focus on the black subsystem of Fig. 2, which yields the system shown in Fig. 3, where the same perfect matching is highlighted in blue. Letting $\varepsilon := 0$ in A_1 causes trouble, due to the occurrence of ε^{-1} in equations (f_4, f_5). Multiplying these two equations by ε does not solve the problem either, since letting $\varepsilon := 0$ in this case erases y'' from equation (f_5), which makes A_1 singular. Only one action can overcome this: properly identifying impulsive variables and rescaling them. We propose a simple approach to identify and quantify impulsive behaviors:

1) Say that variable x has *rescaling offset* k, written $\mu_x{=}k$, if its value in the solution of system A_1 is $0(\varepsilon^{-k})$;
2) $\mu_x{=}k$ results in the rescaling $x^{\downarrow} =_{\mathsf{def}} \varepsilon^k{\times}x$.

(6)

Focus on equation (f_5). Since y' and $y'^{\bullet}$ are both non impulsive, equation (f_5) in array A_1 implies $\mu_{y''}{=}1{+}\max(\mu_{y'}, \mu_{y'^{\bullet}}){=}1$. In addition, we also associate, to function f_5, a rescaling offset indicating the magnitude order w.r.t. ε^{-1} of f_5 in the neighborhood of a solution of this equation. That is:

$$\mu_{f_5} = \max\big(\mu_{y''}, 1{+}\max(\mu_{y'}, \mu_{y'^{\bullet}})\big) = 1\,.$$

Performing this kind of reasoning for all the equations of array A_1 yields the following *rescaling analysis:*

$$\left\{ \begin{array}{rcl} \mu_{x''} = \mu_{f_1} & = & \max(\mu_{x''}, \mu_{\lambda}) \\ \mu_{\lambda} = \mu_{f_2} & = & \max(\mu_{y''}, \mu_{\lambda}) \\ \mu_{y'^{\bullet}} = \mu_{f_3} & = & \max(\mu_{x'^{\bullet}}, \mu_{y'^{\bullet}}) \\ 1 + \mu_{x'^{\bullet}} = \mu_{f_4} & = & \max(\mu_{x''}, 1 + \mu_{x'^{\bullet}}) \\ \mu_{y''} = \mu_{f_5} & = & \max(\mu_{y''}, 1 + \mu_{y'^{\bullet}}) \end{array} \right. \tag{7}$$

We search for solutions of the rescaling analysis, satisfying the following requirement, expressing that state variables for restart should not be impulsive:

Requirement 2 (on restart variables): State variables of the last instant possess rescaling offset zero. ♦

The only solution making restart variables non-impulsive is the following: $\mu_{x''} = \mu_{\lambda} = \mu_{y''} = 1 = \mu_{f_5} = \mu_{f_2} = \mu_{f_1}$, whereas other variables and functions have rescaling offset zero. Rescaling variables and equations using (6), Step 2, yields:

$$\begin{array}{rcl@{\quad;\quad}rcl} x''^{\downarrow} & =_{\mathsf{def}} & \varepsilon{\times}x'' = x'^{\bullet}{-}x' & f_1^{\downarrow} & =_{\mathsf{def}} & \varepsilon{\times}f_1 \\ y''^{\downarrow} & =_{\mathsf{def}} & \varepsilon{\times}y'' = y'^{\bullet}{-}y' & f_5^{\downarrow} & =_{\mathsf{def}} & \varepsilon{\times}f_5 \\ \lambda^{\downarrow} & =_{\mathsf{def}} & \varepsilon{\times}\lambda & f_2^{\downarrow} & =_{\mathsf{def}} & \varepsilon{\times}f_2 \end{array} \tag{8}$$

By using (8), we have

$$A_1^{\downarrow} : \left\{ \begin{array}{l} 0 = x''^{\downarrow} + \lambda^{\downarrow} x \\ 0 = y''^{\downarrow} + \lambda^{\downarrow} y \\ 0 = (xx'{+}yy')^{\bullet} \\ 0 = x''^{\downarrow} - (x'^{\bullet} - x') \\ 0 = y''^{\downarrow} - (y'^{\bullet} - y') \end{array} \right. \tag{9}$$

At this point, letting $\varepsilon := 0$ in $A_1^{\downarrow}$ preserves regularity, structurally — actually, ε already disappeared from $A_1^{\downarrow}$.

Hot restart: We are now ready to conclude. New positions were determined by the previous mode: $x^{\bullet} = x$ and $y^{\bullet} = y$, expressing that positions are continuous at mode change. The restart system is finally obtained by renaming, in (9):

1) the variables set by the previous mode x, y, x', y', by the left-limits at mode change x^-, y^-, x'^-, y'^-;

2) the tail variables $x^\bullet, y^\bullet, x'^\bullet, y'^\bullet$, by the restart values for the new mode x^+, y^+, x'^+, y'^+; whereas
3) other variables are auxiliary and are not renamed.

Finally, performing the above legitimate renaming yields the

$$\text{restart system}: \left\{ \begin{array}{l} 0 = x^+ - x^- \\ 0 = y^+ - y^- \\ 0 = x'^+ - x'^- + \lambda^\downarrow x^- \\ 0 = y'^+ - y'^- + \lambda^\downarrow y^- \\ 0 = x^- x'^+ + y^- y'^+ \end{array} \right. \tag{10}$$

*Note that eliminating the auxiliary variable $\lambda^\downarrow$ from (10) yields $x'^+ y^+ - y'^+ x^+ = x'^- y^- - y'^- x^-$, which expresses the **preservation of angular momentum**, a physical invariant, which our approach "discovered", despite it was not explicitly specified in the original modeling.*

III. A General Algorithm: Sketch

Due to lack of space, we can only sketch how the approach developed for our illustrative example can be made systematic. Details will be reported elsewhere.

A *mode change* is a triple (F^-, t, F), where F^- and F are square multimode DAE systems of equations of the form (1) and $t \in \mathbb{R}_+$. F^- is the *previous mode*, t is the mode change instant, and F is the *new mode*. We can assume that F^- is index-reduced and that F is completed with its latent equations (i.e., both consistency and leading equations are included). We denote by X the set of variables and derivatives occurring in F.

A. Mode change arrays and facts

In this section, we discretize the timeline using time step $\varepsilon > 0$: $\mathbb{T} = \{t + n\varepsilon \mid n \in \mathbb{Z}\}$. We use the following notations:

$$x =_{\mathsf{def}} x(t) \text{ and } x^{\bullet k} =_{\mathsf{def}} x(t + k\varepsilon), k \in \mathbb{Z}\,.$$

*1) Linking derivatives and shifts through **Euler identities**:* We denote by x'^n the n-th derivative of x, and its shifted versions are denoted by $x'^{n\bullet k}$, whereas the set of all $x'^{n\bullet k}$ is denoted by $\boldsymbol{\Xi}'^\bullet$. The first-order forward Euler scheme is enforced for all derivatives:

$$x' = \varepsilon^{-1}(x^\bullet - x)\,. \tag{11}$$

This *Euler identity* allows to define derivatives everywhere, including at the mode change instant, where discontinuities or even impulses can occur. See array A_1, Fig. 2.

*2) Identifying **past variables,** which are determined by the previous mode:* Dynamics F^- hold for $s \in \mathbb{T}, s < t$. By (11), some variables occurring in the new mode are determined by the previous mode, we call them *past variables.* Hot restart differs from cold start: past variables reduce the nondeterminism of consistency conditions for the new mode.

*3) Constructing **mode change arrays**:* Recalling that F is augmented with its latent equations, we form the K-array A_K for the new mode by stacking successive *blocks* $F, F^\bullet, \ldots, F^{\bullet K}$. We regard A_K as an algebraic system of equations in both the variables appearing at instant t or later, their derivatives (treated here as separate algebraic variables), and their shifted versions. Past variables are free variables of this array.

*4) Identifying **facts** and erasing them:* If the mode change is state-based, some properties are guaranteed at the mode change. If the mode change condition is the *zero-crossing* of some smooth function g (crossing zero upwards), then $g{=}0$ holds at the mode change instant, up to an $O(\varepsilon)$. If equation $(f) \in A_K$ is entailed by $g{=}0$ and involves past variables only, then $f{=}0$ holds up to $O(\varepsilon)$. Such *facts* can be safely erased from array A_K. This was illustrated in A_1, Fig. 2.

B. Rescaling analysis of mode change arrays

The set of functions possibly involved in arrays is the set of smooth functions on $\boldsymbol{\Xi}'^\bullet$, augmented with the set of Euler identities possibly relating variables belonging to $\boldsymbol{\Xi}'^\bullet$. This is subsumed by the class of functions of the form

$$f = a_f + \sum_{y \in \mathcal{L}_f} a_{f,y}\, \varepsilon^{-n_{f,y}} y + f_{n\ell}(\mathcal{N}_f) \quad : \quad a_f, a_{f,y} \in \mathbb{R}, \tag{12}$$

where the set of variables involved in f decomposes into the subset $\mathcal{L}_f$ entering linearly in f and its complement $\mathcal{N}_f$. For f of the form (12) and $\mathcal{M}_K$ a matching of array A_K, define $f_{\downarrow \mathcal{M}_K}$ by restricting, in (12), the sum to the subset of its terms involving the variable x_f matched with f in $\mathcal{M}_K$.

Call *rescaling offset* a function $f \mapsto \mu_f$, where we write μ_x when $f \equiv x$ and "f non-linear" means $\mathcal{L}_f = \varnothing$:

$$\begin{array}{rcl} \forall n \in \mathbb{N} & , & \mu_{\varepsilon^{-n}} = n \\ \forall f \text{ as in (12)} & , & \mu_f = \max\big(\max_{y \in \mathcal{L}_f}(n_{f,y} + \mu_y)\,,\ \mu_{f_{n\ell}}\big) \\ \forall f \text{ non-linear} & , & \mu_f = \textbf{ if } \mu_{\mathcal{X}_f} = 0 \textbf{ then } 0 \textbf{ else } +\infty \end{array}$$

where $\mu_{\mathcal{X}_f} = 0$ means $\mu_y = 0$ for every $y \in \mathcal{X}_f$. These axioms are supported by the following intuition. The rescaling offset aims to represent the "impulse order", namely $\mu_y = n$ captures that y is $O(\varepsilon^{-n})$. The first axiom expresses that intent, and the second one is immediate considering (12). The last axiom formalizes that, if f is nonlinear, we do not know to assign an "impulse order" to it, unless it is known that all the variables involved in f are not impulsive (this somehow parallels [11]). Note that rescaling offsets abstract away the real valued coefficients $a_f, a_{f,y}$; only their occurrence/absence plays a role in the above axioms.

Using μ_f, function f is *rescaled* as follows:

$$f^\downarrow =_{\mathsf{def}} \textbf{ if } \mu_f < \infty \textbf{ then } \varepsilon^{\mu_f} f \textbf{ else } \text{"undefined"} \tag{13}$$

Rescaling Problem (RP): Find:

1) a $K \geqslant 0$ and a variable-complete matching $\mathcal{M}_K$ for A_K, such that every equation of the Kth block of A_K is matched;
2) a rescaling offset μ solving the system of equations

$$\forall f \in \mathcal{M}_K \quad , \quad \mu_f = \mu_{f_{\downarrow \mathcal{M}_K}} \tag{14}$$

and satisfying the following additional conditions:

$$\forall y \in \mathcal{X}_{A_K} \quad , \quad \mu_y < \infty\,; \tag{15}$$
$$\forall f \in \mathcal{M}_K \quad , \quad \mu_f < \infty\,; \tag{16}$$
$$\forall y \in \mathcal{X}^{\mathsf{tail}}_{A_K} \quad , \quad \mu_y = 0\,, \tag{17}$$

where $\mathcal{X}^{\mathsf{tail}}_{A_K}$ is the set of all the K-shifted variables of A_K. Triple $(K, \mathcal{M}_K, \mu)$ is a *solution* of (RP). Using $(K, \mathcal{M}_K, \mu)$

and denoting by A_ε the subarray of A_K collecting the equations matched in $\mathcal{M}_K$, apply the following procedure:

1) erase from A_ε all equations not matched in $\mathcal{M}_K$;
2) rescale the remaining equations and variables using (13): the array that results from this rescaling is denoted by $A_\varepsilon^\downarrow$;
3) set $\varepsilon := 0$ in $A_\varepsilon^\downarrow$, and denote the result by $A_0^\downarrow$;
4) rename as follows all the variables occurring in $A_0^\downarrow$:

$$\begin{array}{rcll} x'^{m\bullet k} \in \mathcal{X}^- & \leftarrow & (x'^m)^- & \text{(past variables) ;} \\ x'^{m\bullet k} \in \mathcal{X}^{\mathsf{tail}}_{A_K} & \leftarrow & (x'^m)^+ & \text{(restart variables) ;} \\ x'^{m\bullet k} \notin \mathcal{X}^{\mathsf{tail}}_{A_K} & \leftarrow & x'^{m\bullet k} & \text{(auxiliary variables) .} \end{array} \tag{18}$$

Let $A_\star$ be the array obtained from A_ε by performing the above sequence of operations: we call it the *restart system*. See (7–10) for an illustration on the cup-and-ball example.

Theorem 1: $A_\star$ satisfies Requirements 1 and 2. ♦

Proof: Let f be as in (12) and such that $\mu_f < \infty$. Using the axioms of rescaling offsets, we get $\mu_{f_{n\ell}} = 0$, and:

$$\begin{array}{rcl} f^\downarrow & = & \varepsilon^{\mu_f}\left(\star + \sum_{y\in\mathcal{L}_f} \star\, \varepsilon^{-n_{f,y}}\; y + f_{n\ell}(\mathcal{N}_f)\right) \\ & = & \varepsilon^{\mu_f}\left(\star + \sum_{y\in\mathcal{L}_f} \star\, \varepsilon^{-n_{f,y}}\; \varepsilon^{-\mu_y} y^\downarrow + f_{n\ell}(\mathcal{N}_f)\right) \\ & = & \left(\sum_{y\in\mathcal{L}_f} \star\, \varepsilon^{\mu_f-\tilde{\mu}_{f,y}}\; y^\downarrow\right) + \varepsilon^{\mu_f} f_{n\ell}(\mathcal{N}_f) \end{array}$$

where $\tilde{\mu}_{f,y} =_{\mathsf{def}} n_{f,y}+\mu_y$. Using the second axiom of rescaling offsets and Problem (RP), we get $\mu_f - \tilde{\mu}_{f,y} \geqslant 0$, with equality for the term in the sum involving the unique variable y such that $(f,y)\in\mathcal{M}_K$. Hence, when setting $\varepsilon:=0$ in the rescaled array $A_\varepsilon^\downarrow$, the presence of term $y^\downarrow$ remains guaranteed for $(f,y)\in\mathcal{M}_K$. Therefore, rescaling A_ε and then setting $\varepsilon := 0$ ensures that $\mathcal{M}_0^\downarrow =_{\mathsf{def}} \{(f^\downarrow, y^\downarrow) \mid (f,y)\in\mathcal{M}_K\}$ is a variable-complete matching for $A_0^\downarrow$, hence $A_0^\downarrow$ is structurally nonsingular. The theorem follows easily. ■

Additional results are only briefly reported here, due to lack of space:

1) Rescaling analysis with given K: When K is fixed, Problem (RP) is shown to be equivalent to a pair of dual Linear Programming problems, reinforced by the requirement that tail variables of the array are all non-impulsive (possess zero rescaling offset). If a solution of these LP exists, a unique componentwise minimal solution exists and is integer valued. Furthermore, it does not depend on the particular choice of $\mathcal{M}_K$ in the solution of (RP).

2) Letting K vary: If a minimal solution of the above LP exists but some of its tail variables are impulsive, then there is no point trying a larger K.

3) Bounds on K: There exist two heights $K_* \leqslant K^*$, depending on the equation offsets of the new mode S and the set $\mathcal{X}^-$ of past variables, such that no variable-complete matching $\mathcal{M}_K$ in which all tail equations are matched, can be found for $K < K_*$; such a matching, however, can always be found for some $K \leqslant K^*$.

4) Preserving invariants: Every invariant of the multi-mode system yields, after the rescaling found by our analysis, a corresponding invariant at the mode change. Hence, the preservation of angular momentum in the cup-and-ball example was by no means incidental.

IV. Conclusion and perspectives

We have presented a fully sound algorithmic solution to the hot restart at a mode change separating two long modes. Our method is computationally simple and is expected to scale up nicely. General nonlinear systems and state-triggered mode changes are supported.

Unlike previous approaches by the control community, ours is structural, which fits the generation of simulation code in DAE-based modeling languages such as Modelica.

Extending this approach to *cascades of transient modes* (of zero duration) is the next research step, as illustrated by the cup-and-ball example in the case of elastic impact.

References

[1] 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, *Comp. Software Sci. – SOTA Persp.*, volume 10000 of *Lecture Notes in Comp. Sci.*, pages 283–310. Springer, 2019.

[2] Albert Benveniste, Benoît Caillaud, and Mathias Malandain. The mathematical foundations of physical systems modeling languages. *Annual Reviews in Control*, 50:72–118, 2020.

[3] Albert Benveniste, Benoît Caillaud, Mathias Malandain, and Joan Thibault. Algorithms for the structural analysis of multimode Modelica models. *Electronics*, 11(17), 2022.

[4] B. Brogliato. *Nonsmooth Mechanics: Models, Dynamics and Control*. Communications and Control Engineering. Springer London, 2012.

[5] Peter Bunus and Peter Fritzson. Methods for structural analysis and debugging of modelica models. *Proc. 2nd Int. Modelica Conf.*, 05 2002.

[6] Benoît Caillaud, Mathias Malandain, and Joan Thibault. Implicit structural analysis of multimode DAE systems. In *23rd ACM Int. Con. on Hybrid Syst.: Comp. and Control (HSCC 2020)*, Sydney, Australia, April 2020.

[7] Stephen L. Campbell and C. William Gear. The index of general nonlinear DAEs. *Numerische Mathematik*, 72:173–196, 1995.

[8] Yahao Chen and Stephan Trenn. On impulse-free solutions and stability of switched nonlinear differential-algebraic equations. *Automatica*, 156(111208):1–14, 2023.

[9] Hilding Elmqvist. *A structured model language for large continuous systems*. PhD thesis, Universiteit i Lund, 1978.

[10] Charles William Gear, Ben Leimkuhler, and Gopal K Gupta. Automatic integration of Euler-Lagrange equations with constraints. *J. Comput. Appl. Math.*, 12:77–90, 1985.

[11] Rukhsana Kausar and Stephan Trenn. Impulses in structured nonlinear switched DAEs. In *Proc. 56th IEEE Conf. Decis. Control*, pages 3181 – 3186, Melbourne, Australia, 2017.

[12] Daniel Liberzon and Stephan Trenn. Switched nonlinear differential algebraic equations: Solution theory, Lyapunov functions, and stability. *Automatica*, 48(5):954–963, 2012.

[13] T. Lindstrøm. An invitation to nonstandard analysis. In N.J. Cutland, editor, *Nonstandard Analysis and its Applications*, pages 1–105. Cambridge Univ. Press, 1988.

[14] Sven Erik Mattsson, Martin Otter, and Hilding Elmqvist. Modelica hybrid modeling and efficient simulation. In *Proc. 38th IEEE Conf. on Decis. Control*, pages 3502–3507, 1999.

[15] C. C. Pantelides. The consistent initialization of differential-algebraic systems. *SIAM J. Sci. Stat. Comput.*, 9(2):213–231, 1988.

[16] Friedrich Pfeiffer. On non-smooth multibody dynamics. *Proc. Inst. Mech. Eng., Part K: J. Multi-body Dyn.*, 226(2):147–177, 2012.

[17] J. D. Pryce. A simple structural analysis method for DAEs. *BIT Numer. Math.*, 41(2):364–394, 2001.

[18] Stephan Trenn. *Distributional Differential Algebraic Equations*. PhD thesis, Technischen Universität Ilmenau, 2009.

[19] Stephan Trenn. Regularity of distributional differential algebraic equations. *Math. Control Signals Syst.*, 21(3):229–264, 2009.