Proceedings of the 42nd IEEE
Conference on Decision and Control
Maui, Hawaii USA, December 2003

Shooting for Border Collision Bifurcations in Hybrid Systems

Vaibhav Donde
Dept. of Elec. and Comp. Engineering
University of Illinois at UC
Urbana, IL 61801
donde@ece.uiuc.edu

Ian A. Hiskens
Dept. of Elec. and Comp. Engineering
University of Wisconsin-Madison
Madison, WI 53706
hiskens@engr.wisc.edu

***Abstract*— Border collision bifurcations refer to situations where a small parameter variation induces a change in the event sequence of a hybrid system. At such a bifurcation, the system trajectory makes tangential contact with an event triggering hypersurface. This bounding case separates regions of (generally) quite different dynamic behaviour. In the paper, the conditions governing bifurcation points are formulated as a boundary value problem. A shooting method is used to solve that problem. The approach is applicable for general nonlinear hybrid systems.**

I. Introduction

Hybrid systems are typified by strong coupling between continuous dynamics and discrete events. For such systems, event triggering generally has a significant influence over subsequent system behaviour. Therefore it is important to identify situations where a small change in parameters alters the event triggering pattern. Such situations include border collision bifurcations [1], also referred to as C-bifurcations [2] and switching time bifurcations [3], and grazing bifurcations [4], [5]. Subtle differences exist between border collision and grazing bifurcations. However both phenomena are induced by tangential contact between the system trajectory and an event triggering hypersurface.[1] Figure 1 provides an illustration.

Referring to Figure 1, for a certain value of parameter λ^+, the system trajectory encounters an event triggering hypersurface at a point x^+. The event occurs, and the trajectory continues accordingly. However for a small change in parameter value, to λ^-, the trajectory misses (at least locally) the triggering hypersurface, and subsequently exhibits a completely different form of response. At a parameter value λ^{bc}, lying between λ^+ and λ^-, the continuous trajectory tangentially encounters the triggering hypersurface. Behaviour beyond the point of touching x^{bc} is generally unpredictable, in the sense that without further knowledge of the system, it is impossible to determine whether or not the event triggers.[2] This bounding case describes a border collision bifurcation, with λ^{bc} referring to the critical value of the bifurcation parameter. This paper proposes a shooting method for locating such bifurcations.

Research supported by the National Science Foundation through grant ECS-0114725.

[1]The paper focuses on this common feature. Therefore for convenience, both phenomena will be referred to as border collision bifurcations.

[2]Under fairly strong assumptions, behaviour is predictable [2].

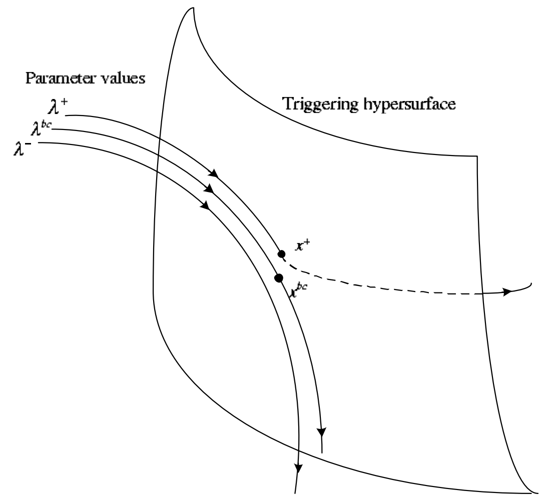

Fig. 1. Border collision bifurcation.

Previous investigations of border collision bifurcations have focused largely on classifying the (local) consequences of bifurcations through analysis of eigenvalue behaviour. Efforts have been directed primarily towards periodic systems, motivated in many cases by power electronic circuit applications. Computation of actual bifurcation points has generally received little attention. With numerical packages such as AUTO unable to handle non-smooth systems, *ad hoc* approaches have prevailed. This paper addresses that deficiency by establishing a shooting method that is applicable for general nonlinear hybrid systems.

II. Problem Formulation

A. Hybrid system model

A useful, non-restrictive model formulation should be,

- capable of capturing the full range of continuous/discrete hybrid system dynamics,
- computationally efficient, and
- consistent with the development of a shooting method.

It is shown in [6], [7] that these specifications can be met completely by a model that consists of a set of differential-algebraic equations, adapted to incorporate impulsive (state reset) action and switching of the algebraic equations. This *DA Impulsive Switched* (DAIS) model can be written in the form,

$$\dot{x} = f(x,y) + \sum_{j=1}^{r} \delta(y_{r,j})\Big(h_j(x,y) - x\Big) \tag{1}$$

$$0 = g(x,y) \equiv g^{(0)}(x,y) + \sum_{i=1}^{s} g^{(i)}(x,y) \tag{2}$$

where

$$g^{(i)}(x,y) = \begin{cases} g^{(i-)}(x,y) & y_{s,i} < 0 \\ g^{(i+)}(x,y) & y_{s,i} > 0 \end{cases} \quad i = 1, ..., s \tag{3}$$

and

- $x \in R^n$ are dynamic states, and $y \in R^m$ are algebraic states;
- $\delta(.)$ is the Dirac delta. Each impulse term of the summation in (1) can be expressed in the alternative state reset form

  $$x^+ = h_j(x^-, y^-) \quad \text{when } y_{r,j} = 0 \tag{4}$$

  where the notation x^+ denotes the value of x just after the reset event, whilst x^- and y^- refer to the values of x and y just prior to the event. This form motivates a generalization to an implicit mapping $h'_j(x^+, x^-, y^-) = 0$.
- y_r, y_s are selected elements of y that trigger state reset (impulsive) and algebraic switching events respectively; y_r and y_s may share common elements.
- $f, h_j : R^{n+m} \rightarrow R^n$.
- $g^{(0)}, g^{(i\pm)} : R^{n+m} \rightarrow R^m$. Some elements of each $g^{(.)}$ will usually be identically zero, but no elements of the composite g should be identically zero. Each $g^{(i\pm)}$ may itself have a switched form, and is defined similarly to (2)-(3), leading to a nested structure for g.

Equations (1)-(4) are a reformulation (and slight generalization) of the model proposed in [7]. All deterministic hybrid systems can be described using this DAIS form.

A compact development of trajectory sensitivities[3] results from incorporating parameters λ into the dynamic states x. This is achieved by introducing trivial differential equations $\dot{\lambda} = 0$ into (1). This results in the natural partitioning

$$x = \begin{bmatrix} \underline{x} \\ \bar{x} \\ \lambda \end{bmatrix}, \qquad f = \begin{bmatrix} \underline{f} \\ 0 \\ 0 \end{bmatrix}, \qquad h_j = \begin{bmatrix} \underline{x} \\ \bar{h}_j \\ \lambda \end{bmatrix} \tag{5}$$

where $\underline{x}$ are the continuous dynamic states, $\bar{x}$ are discrete dynamic states, and λ are parameters. This partitioning of the differential equations f ensures that away from events, $\underline{x}$ evolves according to $\dot{\underline{x}} = \underline{f}(x,y)$, whilst $\bar{x}$ and λ remain constant. Similarly, the partitioning of the reset equations h_j ensures that $\underline{x}$ and λ remain constant at reset events, but the states $\bar{x}$ are reset to new values given by $\bar{x}^+ = \bar{h}_j(x^-, y^-)$.

[3]Details are summarized in the appendix.

Away from events, system dynamics evolve smoothly according to the familiar differential-algebraic model

$$\dot{x} = f(x,y) \tag{6}$$

$$0 = g(x,y) \tag{7}$$

where g is composed of $g^{(0)}$ together with appropriate choices of $g^{(i-)}$ or $g^{(i+)}$, depending on the signs of the corresponding elements of y_s. At switching events (3), some component equations of g change. To satisfy the new $g = 0$ constraints, algebraic variables y may undergo a step change. Impulse events (1) (alternatively reset events (4)) force a discrete change in elements of $\bar{x}$. Algebraic variables may again step to ensure $g = 0$ is always satisfied.

The *flows* of x and y are defined as

$$x(t) = \phi_x(x_0, t) \tag{8}$$

$$y(t) = \phi_y(x_0, t) \tag{9}$$

where $x(t)$ and $y(t)$ satisfy (1)-(3), along with initial conditions,

$$\phi_x(x_0, t_0) = x_0 \tag{10}$$

$$g\big(x_0, \phi_y(x_0, t_0)\big) = 0. \tag{11}$$

B. Border collision bifurcations

A border collision bifurcation is characterised by a trajectory (flow) of the system touching a triggering hypersurface tangentially. Let the target hypersurface be described by

$$b(x,y) = 0 \tag{12}$$

where $b : R^{n+m} \rightarrow R$. Vectors that are normal to b are therefore given by $\nabla b = \left[\frac{\partial b}{\partial x} \; \frac{\partial b}{\partial y}\right]^t \equiv [b_x \;\; b_y]^t$, and the tangent hyperplane is spanned by vectors $[u^t \; v^t]^t$ that satisfy

$$[b_x \; b_y] \begin{bmatrix} u \\ v \end{bmatrix} = 0. \tag{13}$$

The vector $[\dot{x}^t \; \dot{y}^t]^t$ is directed tangentially along the flow, so it must satisfy (13) at a border collision bifurcation. Furthermore, differentiating (7) and substituting (6) gives,

$$0 = \frac{\partial g}{\partial x}\dot{x} + \frac{\partial g}{\partial y}\dot{y} \tag{14}$$

$$\Rightarrow 0 = g_x f(x,y) + g_y v \tag{15}$$

where for notational convenience v replaces $\dot{y}$.

A single degree of freedom is available for varying parameters to find a bifurcation value. Recall from (5) that parameters λ are incorporated into the initial conditions x_0. Therefore the single degree of freedom can be achieved by parameterization $x_0(\theta)$, where θ is a scalar.

Border collision bifurcation points are therefore described by combining together the flow definition (8) (appropriately parameterized by θ), algebraic equations (7), target hypersurface (12), and tangency conditions (13),(15), to give

$$\phi_x(x_0(\theta), t) - x = 0 \quad (16)$$
$$g(x, y) = 0 \quad (17)$$
$$b(x, y) = 0 \quad (18)$$
$$\begin{bmatrix} b_x & b_y \\ g_x & g_y \end{bmatrix} \begin{bmatrix} f(x, y) \\ v \end{bmatrix} = 0. \quad (19)$$

This set of equations may be written compactly as

$$F(x, y, \theta, t, v) = F(z) = 0 \quad (20)$$

where $F : R^{n+2m+2} \to R^{n+2m+2}$ and $z = [x^t \; y^t \; \theta \; t \; v^t]^t$. Solution of (20) can be achieved using Newton's method. The solution process involves numerical simulation to obtain the flow (16). A discussion of the resulting shooting method follows.

III. Shooting Method

A. Algorithm

Numerical solution of (20) using Newton's method amounts to iterating on the standard update formula

$$z^{k+1} = z^k - \left(DF(z^k)\right)^{-1} F(z^k) \quad (21)$$

where DF is the Jacobian matrix

$$DF = \begin{bmatrix} -I & 0 & \Phi_x \frac{dx_0}{d\theta} & f & 0 \\ g_x & g_y & 0 & 0 & 0 \\ b_x & b_y & 0 & 0 & 0 \\ f^t b_{xx} + b_x f_x + v^t b_{yx} & f^t b_{xy} + b_x f_y + v^t b_{yy} & 0 & 0 & b_y \\ \hat{f}^t g_{xx} + g_x f_x + \hat{v}^t g_{yx} & \hat{f}^t g_{xy} + g_x f_y + \hat{v}^t g_{yy} & 0 & 0 & g_y \end{bmatrix}. \quad (22)$$

with

$$\hat{f} = \begin{bmatrix} f & & & & \\ & f & & & \\ & & f & & \\ & & & \ddots & \\ & & & & f \end{bmatrix} \in R^{mn \times m}$$

$$\hat{v} = \begin{bmatrix} v & & & & \\ & v & & & \\ & & v & & \\ & & & \ddots & \\ & & & & v \end{bmatrix} \in R^{m^2 \times m}$$

$$g_{xx} = \begin{bmatrix} \frac{\partial^2 g_1}{\partial x^2} \\ \frac{\partial^2 g_2}{\partial x^2} \\ \vdots \\ \frac{\partial^2 g_m}{\partial x^2} \end{bmatrix} \in R^{mn \times n}, \quad g_{yx} = \begin{bmatrix} \frac{\partial^2 g_1}{\partial y \partial x} \\ \frac{\partial^2 g_2}{\partial y \partial x} \\ \vdots \\ \frac{\partial^2 g_m}{\partial y \partial x} \end{bmatrix} \in R^{m^2 \times n}$$

$$g_{yy} = \begin{bmatrix} \frac{\partial^2 g_1}{\partial y^2} \\ \frac{\partial^2 g_2}{\partial y^2} \\ \vdots \\ \frac{\partial^2 g_m}{\partial y^2} \end{bmatrix} \in R^{m^2 \times m}, \quad g_{xy} = \begin{bmatrix} \frac{\partial^2 g_1}{\partial x \partial y} \\ \frac{\partial^2 g_2}{\partial x \partial y} \\ \vdots \\ \frac{\partial^2 g_m}{\partial x \partial y} \end{bmatrix} \in R^{mn \times m}.$$

The matrices g_{xx}, g_{yx}, g_{xy} and g_{yy} are usually extremely sparse. It has been found that often the error introduced into DF by ignoring them has negligible effect on convergence. However situations can arise where these terms are vital for reliable convergence. This is the case, for example, when the trajectory has multiple turning points (peaks and troughs) in the vicinity of the border hypersurface. The example of Section IV provides an illustration. Two approaches have been used to obtain the second derivative terms:

1) Numerical differencing. Many simulators allow direct computation of g_x and g_y.[4] Numerical differencing of g_x and g_y is straightforward, but not particularly efficient for high dimensional systems.
2) Direct computation. By utilizing an object oriented modelling structure [8], second derivative terms occur only within components. There are no terms introduced by inter-component dependencies. Explicit formulae for second derivative terms can be established for each component model. The sparse matrices can then be efficiently constructed.

The entry Φ_x in (22) gives the sensitivity of the flow (8) to perturbations in initial conditions x_0,

$$\Phi_x(t) \equiv \frac{\partial \phi_x}{\partial x_0}(x_0, t).$$

The variational equations describing the evolution of trajectory sensitivities Φ_x, Φ_y are given in Appendix A. Note that these quantities are defined for non-smooth trajectories generated by hybrid systems [7].

Care must be taken in evaluating the terms of (20) and (22) that relate to trajectory solution. The flow term $\phi_x(x_0(\theta^k), t^k)$ of

$$F_1(x^k, \theta^k, t^k) = \phi_x(x_0(\theta^k), t^k) - x^k$$

[4] These quantities are required for implicit numerical integration.

evaluates, via numerical integration, to the value of x at time t^k along the trajectory that has initial value $x_0(\theta^k)$. Likewise, the terms Φ_x and f in the first row of DF should also be evaluated at time t^k along that trajectory.

B. Initialization of variables

As with all iterative procedures, solution of (21) requires a good initial guess z^0. In terms of the original system variables, initial (approximate) border collision bifurcation values of x^0, y^0, θ^0, t^0, and v^0 are required. These can be obtained from normal simulation.

Referring to Figure 1, parameter values that are near the bifurcation value result in trajectories that either, 1) encounter the target hypersurface, or 2) just miss the hypersurface. Therefore the trajectory induced by parameter θ^0 should be monitored for,

1) the first point where $b(x, y) = 0$, i.e., an intersection with the target hypersurface, or
2) an appropriate local minimum of $b(x, y)$, i.e., a point where the trajectory passes close by the target hypersurface. This point is given by $\frac{db}{dt} = 0$, which implies $b_x\dot{x} + b_y\dot{y} = 0$. Substituting for $\dot{y}$ from (14), and using (6), gives

$$\frac{db}{dt} = (b_x - b_y g_y^{-1} g_x) f(x, y) = 0.$$

In both cases, the identified point directly provides initial values for x^0, y^0, and t^0. The corresponding value of v^0 can be obtained from (15) as

$$v^0 = -g_y^{-1} g_x f(x^0, y^0)$$

with all partial derivatives evaluated at x^0, y^0.

IV. Example

The single machine infinite bus power system of Figure 2 was used to test the algorithm developed in Section III. The generator was accurately represented by a sixth order machine model, viz., a two axis model with two windings in each axis [9], and the generator excitation system was modelled according to Figure 3. Note that the output limits on the field voltage E_{fd} are anti-wind-up limits, whilst the limits on the stabilizer output V_{PSS} are clipping limits. Therefore even though this example utilizes a simple network structure, it exhibits nonlinear, non-smooth, hybrid system behaviour. Larger systems are no more challenging.

A single phase fault was applied at the generator terminal bus at 0.05 sec. The fault was cleared, without line tripping, at 0.28 sec. The aim of the exercise was to determine the maximum value of the field voltage limit E_{fdmax} that ensured the initial terminal voltage overshoot did not rise above the threshold value of 1.2 pu. (The target hypersurface was therefore $V_t - 1.2 = 0$.) This threshold is indicative of over-voltage protection. Excursions above that value could trigger protection and ultimately trip the generator.

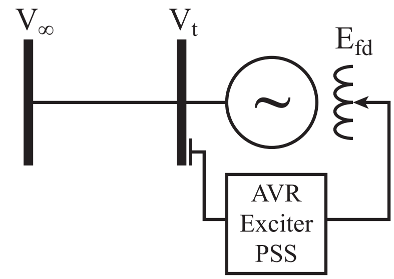

Fig. 2. Single machine infinite bus system.

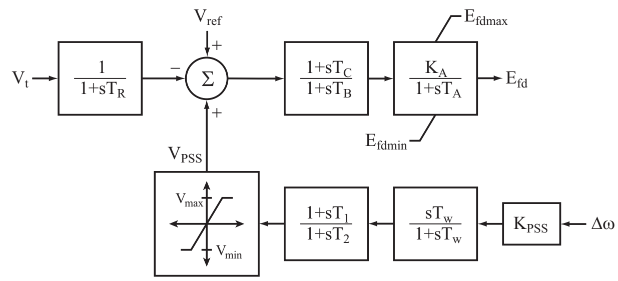

Fig. 3. Excitation system (AVR/PSS) representation.

Results of the iterative process are given in Table I, and presented graphically in Figures 4 and 5. It can be seen that convergence was effectively achieved after four iterations. This is an encouraging result, as an onerous test condition was chosen. Referring to Figure 4, it can be seen that the voltage trajectory is quite flat over the first extended peak, and actually consists of two peaks separated by a shallow trough. The solution set formulation (16)-(19) also describes troughs, so this example has three solutions in quite close proximity. This situation indicates proximity to a saddle-node bifurcation [10], and the Jacobian DF is accordingly quite ill-conditioned. However convergence was still reliable.

V. Conclusions

Hybrid systems, where discrete events have a significant influence over system behaviour, are susceptible to border collision bifurcations. This form of bifurcation refers to the situation where the system trajectory is tangential to an event triggering hypersurface, i.e., the encounter is not transversal, as required for well-defined behaviour.

TABLE I

ITERATION RESULTS.

Iteration	Bifurcation Values	
	Param, E_{fdmax}	Time, t^*
0	5.80	1.12
1	3.15	1.28
2	4.37	1.19
3	4.72	1.20
4	4.78	1.21
5	4.78	1.21

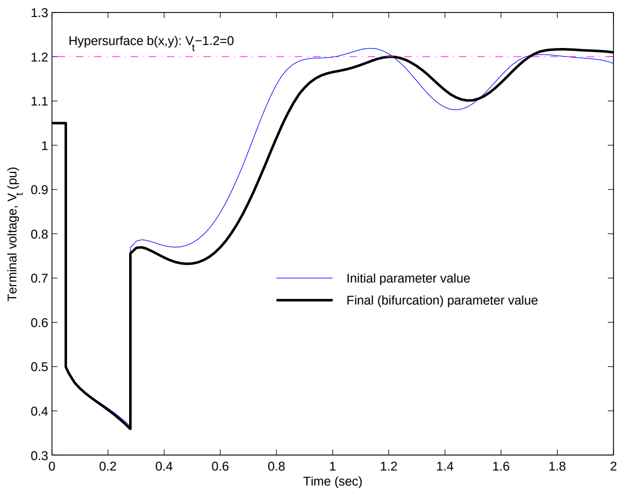

Fig. 4. Terminal voltage, V_t.

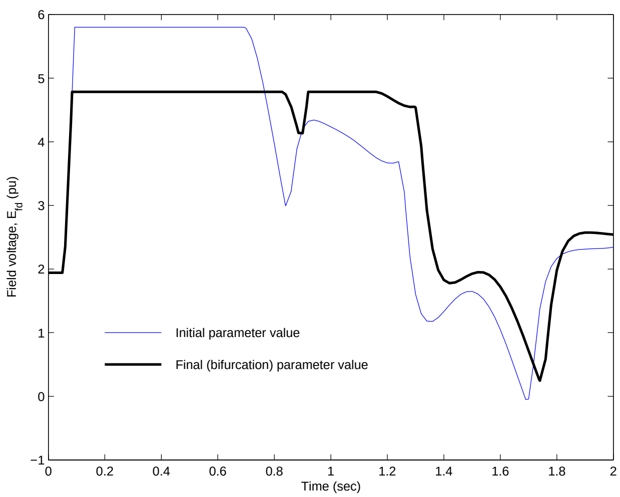

Fig. 5. Generator field voltage, E_{fd}.

Border collision bifurcation points can be described by a set of nonlinear, algebraic equations. Iterative solution via Newton's method requires numerical integration of the system trajectory, and therefore has the form of a shooting method. The associated Jacobian incorporates trajectory sensitivities, which can be efficiently computed along with the trajectory. The shooting method is therefore practical for arbitrarily large hybrid systems.

VI. REFERENCES

[1] H. Nusse, E. Ott, and J. Yorke, "Border-collision bifurcations: An explanation for observed bifurcation phenomena," *Physical Review E*, vol. 49, no. 2, pp. 1073–1077, February 1994.

[2] M. di Bernardo, M. Feigen, S. Hogan, and M. Homer, "Local analysis of C-bifurcations in n-dimensional piecewise-smooth dynamical systems," *Chaos, Solitons and Fractals*, vol. 10, no. 11, pp. 1881–1908, 1999.

[3] R. Rajaraman, I. Dobson, and S. Jalali, "Nonlinear dynamics and switching time bifurcations of a thyristor controlled reactor circuit," *IEEE Transactions on Circuits and Systems I*, vol. 43, no. 12, pp. 1001–1006, December 1996.

[4] W. Chin, E. Ott, H. Nusse, and C. Grebogi, "Universal behavior of impact oscillators near grazing incidence," *Physical Letters A*, vol. 201, no. 2, March 1995.

[5] H. Dankowicz, P. Piiroinen, and A. Nordmark, "Low-velocity impacts of quasiperiodic oscillations," *Chaos, Solitons and Fractals*, vol. 14, pp. 241–255, 2002.

[6] I. Hiskens, "Power system modeling for inverse problems," *IEEE Transactions on Circuits and Systems I*, to appear.

[7] I. Hiskens and M. Pai, "Trajectory sensitivity analysis of hybrid systems," *IEEE Transactions on Circuits and Systems I*, vol. 47, no. 2, pp. 204–220, February 2000.

[8] I. Hiskens and P. Sokolowski, "Systematic modeling and symbolically assisted simulation of power systems," *IEEE Transactions on Power Systems*, vol. 16, no. 2, pp. 229–234, May 2001.

[9] P. Sauer and M. Pai, *Power System Dynamics and Stability*. Upper Saddle River, NJ: Prentice Hall, 1998.

[10] R. Seydel, *Practical Bifurcation and Stability Analysis*, 2nd ed. New York: Springer-Verlag, 1994.

[11] W. Feehery, J. Tolsma, and P. Barton, "Efficient sensitivity analysis of large-scale differential-algebraic systems," *Applied Numerical Mathematics*, vol. 25, pp. 41–54, 1997.

[12] S. Li, L. Petzold, and W. Zhu, "Sensitivity analysis of differential-algebraic equations: A comparison of methods on a special problem," *Applied Numerical Mathematics*, vol. 32, pp. 161–174, 2000.

APPENDIX
VARIATIONAL EQUATIONS

Away from events, where system dynamics evolve smoothly, the sensitivities Φ_x and Φ_y are obtained by differentiating (6)-(7) with respect to x_0. This gives

$$\dot{\Phi}_x = f_x(t)\Phi_x + f_y(t)\Phi_y \tag{23}$$
$$0 = g_x(t)\Phi_x + g_y(t)\Phi_y \tag{24}$$

where $f_x \equiv \partial f/\partial x$, and likewise for the other Jacobian matrices. Note that f_x, f_y, g_x, g_y are evaluated along the trajectory, and hence are time-varying matrices. It is shown in [7], [11], [12] that the solution of this (potentially high order) linear, time-varying DAE system can be obtained as a by-product of solving the original DAE system (6)-(7).

Initial conditions for Φ_x are obtained from (10) as

$$\Phi_x(t_0) = I$$

where I is the identity matrix. Initial conditions for Φ_y follow directly from (24),

$$0 = g_x(t_0) + g_y(t_0)\Phi_y(t_0).$$

Equations (23)-(24) describe the evolution of the sensitivities Φ_x and Φ_y between events. However at an event, the sensitivities are often discontinuous. It is necessary to calculate *jump conditions* describing the step change in Φ_x and Φ_y. For clarity, consider a single switching/reset event, so the model (1)-(4) effectively reduces to the form

$$\dot{x} = f(x, y) \tag{25}$$

$$0 = \begin{cases} g^-(x, y) & s(x, y) < 0 \\ g^+(x, y) & s(x, y) > 0 \end{cases} \tag{26}$$

$$x^+ = h(x^-, y^-) \qquad s(x, y) = 0. \tag{27}$$

(The switching hypersurface s has been made explicit to clearly identify its role in the jump conditions.)

Let $(x(\tau), y(\tau))$ be the point where the trajectory encounters the hypersurface $s(x, y) = 0$, i.e., the point where an event is triggered. This point is called the *junction point* and τ is the *junction time*. Assume that the trajectory encounters this triggering hypersurface transversally.

Just prior to event triggering, at time τ^-, x and y are given by

$$x^- \equiv x(\tau^-) = \phi_x(x_0, \tau^-)$$
$$y^- \equiv y(\tau^-) = \phi_y(x_0, \tau^-)$$

where

$$g^-(x^-, y^-) = 0.$$

Similarly, x^+, y^+ are defined for time τ^+, just after the event has occurred. It is shown in [7] that the jump conditions for the sensitivities Φ_x are given by

$$\Phi_x(\tau^+) = h_x^* \, \Phi_x(\tau^-) - \left(f^+ - h_x^* \, f^-\right) \tau_{x_0} \tag{28}$$

where

$$\begin{aligned} h_x^* &= \left(h_x - h_y (g_y^-)^{-1} g_x^-\right)\Big|_{\tau^-} \\ \tau_{x_0} &= -\frac{\left(s_x - s_y (g_y^-)^{-1} g_x^-\right)\Big|_{\tau^-} \Phi_x(\tau^-)}{\left(s_x - s_y (g_y^-)^{-1} g_x^-\right)\Big|_{\tau^-} f^-} \\ f^- &\equiv f(x(\tau^-), y^-(\tau^-)) \\ f^+ &\equiv f(x(\tau^+), y^+(\tau^+)). \end{aligned}$$

The sensitivities Φ_y immediately after the event are given by

$$\Phi_y(\tau^+) = -\left(g_y^+(\tau^+)\right)^{-1} g_x^+(\tau^+)\Phi_x(\tau^+).$$

Following the event, i.e., for $t > \tau^+$, calculation of the sensitivities proceeds according to (23)-(24), until the next event is encountered. The jump conditions provide the initial conditions for the post-event calculations.