



SHOOTING METHODS FOR LOCATING GRAZING PHENOMENA IN HYBRID SYSTEMS*

VAIBHAV DONDE

*Lawrence Berkeley National Laboratory, 1 Cyclotron Road,
Berkeley, CA 94720, USA
VDonde@lbl.gov*

IAN A. HISKENS

*Department of Electrical and Computer Engineering,
University of Wisconsin-Madison,
Madison, WI 53706, USA
hiskens@engr.wisc.edu*

Received January 15, 2004; Revised April 11, 2005

Hybrid systems are typified by strong coupling between continuous dynamics and discrete events. For such piecewise smooth systems, event triggering generally has a significant influence over subsequent system behavior. Therefore, it is important to identify situations where a small change in parameter values alters the event triggering pattern. The bounding case, which separates regions of (generally) quite different dynamic behaviors, is referred to as *grazing*. At a grazing point, the system trajectory makes tangential contact with an event triggering hypersurface. The paper formulates conditions governing grazing points. Both transient and periodic behaviors are considered. The resulting boundary value problems are solved using shooting methods that are applicable for general nonlinear hybrid (piecewise smooth) dynamical systems. The grazing point formulation underlies the development of a continuation process for exploring parametric dependence. It also provides the basis for an optimization technique that finds the smallest parameter change necessary to induce grazing. Examples are drawn from power electronics, power systems and robotics, all of which involve intrinsic interactions between continuous dynamics and discrete events.

Keywords: Grazing; limit cycles; piecewise smooth dynamical systems; shooting methods; continuation methods.

1. Introduction

Many systems exhibit dynamic behavior that is best characterized by strong coupling between continuous dynamics and discrete events. Trajectories evolve smoothly through state-space until satisfying conditions that trigger an event. The event alters the system description and/or induces an impulsive change in the state. Smooth evolution then

continues until the next event is triggered. Such systems have come to be known as *hybrid systems* [van der Schaft & Schumacher, 2000; Liberzon, 2003] or *piecewise smooth* dynamical systems [di Bernardo *et al.*, 2003].¹ Examples can be drawn from a wide range of application areas, including process control [Lennartson *et al.*, 1996], constrained mechanical systems [Brogliato, 1999], robotics [Spong &

*Research supported by the National Science Foundation through grant ECS-0332777.

¹The term hybrid systems is preferred by the control community, whereas the bifurcation community is more familiar with piecewise smooth systems.

Vidyasagar, 1989; Piiroinen, 2002], power systems [Hiskens, 2004], and power electronics [Rajaraman *et al.*, 1996; Yuan *et al.*, 1998]. In fact, any physical device that exhibits hysteresis, or control loop with anti-wind-up limits [Goodwin *et al.*, 2001], is effectively a hybrid system.

Event triggering generally has a significant influence over subsequent system behavior. Therefore, it is important to identify situations where a small change in parameter values alters the event triggering pattern. Figure 1 provides an illustration. For a certain value of parameter θ_{hit} , the system trajectory encounters an event triggering hypersurface at a point x_{hit} . The event occurs, and the trajectory continues accordingly. However for a small change in parameter value to θ_{miss} , the trajectory misses (at least locally) the triggering hypersurface, and subsequently exhibits a completely different form of response. At a parameter value θ_g , lying between θ_{hit} and θ_{miss} , the continuous trajectory tangentially encounters (*grazes*) the triggering hypersurface. Behavior beyond the grazing point x_g is generally unpredictable, in the sense that without further knowledge of the system, it is impossible to determine whether or not the event triggers. This bounding case typifies grazing phenomena. The paper proposes a shooting method for locating grazing points x_g and the associated critical parameter value θ_g .

The significance of grazing can be illustrated by considering power system protection. For a

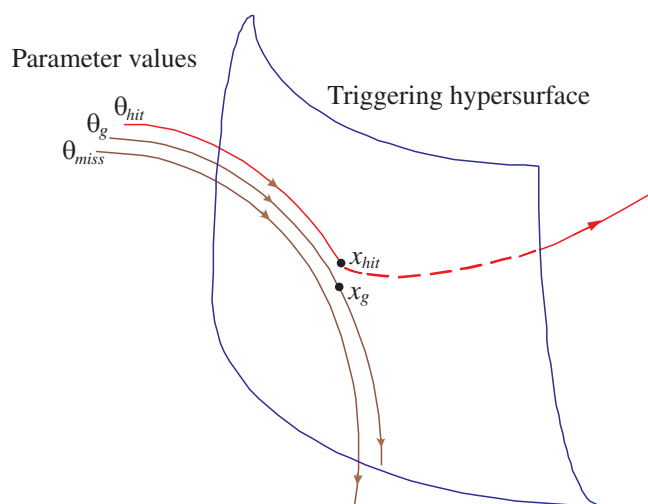


Fig. 1. Trajectory grazing triggering hypersurface.

particular set of parameters, a disturbance may cause the system trajectory to pass close to (but not encounter) a protection triggering characteristic [Singh & Hiskens, 2001]. The protection does not operate, no equipment is tripped, and the system recovers. However it may require only a small change in parameter values, such as loads, for protection to operate, perhaps leading to cascading outages and system fragmentation [U.S.-Canada Power System Outage Task Force, 2004]. Knowledge of the grazing point would highlight system vulnerability.

In a more general sense, grazing plays a fundamental role in dynamic embedded optimization of hybrid systems. Such optimization problems incorporate the system dynamic description into the constraint set [Galán & Barton, 1998; Hiskens, 2004]. As Galán and Barton [1998] and Piccoli [1998] have shown, even though hybrid system dynamics are nonsmooth, the cost function is smooth away from grazing points. This statement is expressed using different terminology though, with Piccoli [1998] requiring that all flows have the same *history*, and Galán and Barton [1998] requiring the existence of trajectory sensitivities (discussed later). Therefore, away from grazing points, gradient-based optimization techniques are appropriate. However, smoothness and even continuity of the cost function is often lost at grazing points. This introduces the need for branch-and-bound algorithms that search over the grazing-induced partitions of parameter space.

Grazing phenomena have been widely investigated recently, particularly with reference to periodic systems [Dankowicz *et al.*, 2002; Chin *et al.*, 1995; Fredriksson & Nordmark, 1997]. In that context, grazing is closely related to border-collision bifurcations [di Bernardo *et al.*, 2001; Nusse *et al.*, 1994; Yuan *et al.*, 1998], also known as C-bifurcations [di Bernardo *et al.*, 1999]. Transient grazing is considered in [Rajaraman *et al.*, 1996; Jalali *et al.*, 1996], where switching-time bifurcations are analyzed. These investigations have focused largely on classifying the (local) consequences of grazing through normal form analysis, particularly for periodic systems. Computation of actual grazing points has received less attention. With numerical packages such as AUTO [Doedel *et al.*, 1998] generally unsuited to nonsmooth systems,² *ad hoc* approaches have prevailed. This paper

²A related package called SlideCont [Dercole & Kuznetsov, 2005] does address nonsmooth systems, but in the context of using a Filippov approach to describe sliding mode behavior. Grazing conditions are not considered.

addresses that deficiency by establishing a shooting method that is applicable for general nonlinear hybrid (piecewise smooth) dynamical systems.

The paper is organized as follows. Section 2 considers the modeling of hybrid systems. Section 3 provides background material for trajectory sensitivities and limit cycles. Conditions governing transient grazing are developed in Sec. 4, and adapted to periodic systems in Sec. 5. The grazing point formulation is extended to a continuation process in Sec. 6. Section 7 develops an optimization technique for finding the smallest parameter change necessary to induce grazing. Conclusions are provided in Sec. 8.

2. Hybrid System Representation

2.1. Model

As mentioned previously, hybrid system behavior is characterized by continuous and discrete states, continuous dynamics, discrete events (triggers), and mappings that define the evolution of discrete states at events. Numerous formal models exist for rigorously describing hybrid system dynamics. Examples include petri nets [David & Alla, 1992] and hybrid automata [van der Schaft & Schumacher, 2000]. However, those representations are not immediately amenable to numerical implementation. A useful, nonrestrictive model formulation should be,

- capable of capturing the full range of continuous/discrete hybrid system dynamics,
- computationally efficient, and
- consistent with the development of shooting methods.³

As shown in [Hiskens & Pai, 2000; Hiskens, 2004], 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,

$$\dot{x} = f(x, y) + \sum_{j=1}^{m_r} \delta(y_{r[j]}) (h_j(x, y) - x) \quad (1)$$

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

where

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

and

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

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

where the notation x^+ denotes the value of x just after the reset event, while 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 $h'_j(x^+, x^-, y^-) = 0$.

- Subscripts $r[j]$, $s[i]$ index elements of y that trigger the j th state reset (impulsive) event and i th algebraic switching event, respectively. (This refers to the i th modeled event, not the event that occurs i th in the time sequence.)
- $f, h_j : \mathbb{R}^{n+m} \rightarrow \mathbb{R}^n$.
- $g^{(0)}, g^{(i\pm)} : \mathbb{R}^{n+m} \rightarrow \mathbb{R}^m$. Some elements of each $g^{(\cdot)}$ 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 .

A nonautonomous version of the DAIS model follows from explicit inclusion of time t in f , g , and h . Alternatively time can be modeled as a state, with dynamics $\dot{x}_t = 1$ and initial condition $x_t(0) = t_0$. This monotonic state x_t may complicate analysis of periodic systems. If so, periodic x_t can be achieved using a reset equation of the form (4),

$$x_t^+ = x_t^- - T \quad \text{when } x_t = t_0 + T,$$

where T is the period.

A compact development of the equations describing grazing phenomena results from incorporating parameters $p \in \mathbb{R}^\ell$ into the dynamic states x . (Numerical implementation is also simplified.) This is achieved by introducing trivial differential

³Shooting methods couple Newton's method with numerical integration, see [Stoer & Bulirsch, 1993]. Such methods are discussed in detail later in the paper.

equations

$$\dot{p} = 0 \tag{5}$$

into (1), and results in the natural partitioning

$$x = \begin{bmatrix} \underline{x} \\ p \end{bmatrix}, \quad f = \begin{bmatrix} \underline{f} \\ 0 \end{bmatrix}, \quad h_j = \begin{bmatrix} \underline{h}_j \\ p \end{bmatrix} \tag{6}$$

where \underline{x} are the true dynamic states, and p are parameters.

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

$$\dot{x} = f(x, y) \tag{7}$$

$$0 = g(x, y) \tag{8}$$

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 $y_{s[i]}$. 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) (equivalently reset events (4)) force a discrete change in elements of \underline{x} . Algebraic variables may again step to ensure $g = 0$ is always satisfied.

The flows of x and y are defined as

$$x(t) = \phi(x_0, t) = \begin{bmatrix} \underline{\phi}(x_0, t) \\ p \end{bmatrix} \tag{9}$$

$$y(t) = \psi(x_0, t) \tag{10}$$

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

$$\phi(x_0, t_0) = x_0 \tag{11}$$

$$g(x_0, \psi(x_0, t_0)) = 0. \tag{12}$$

The partitioning of ϕ in (9) is in accordance with (6).

2.2. Example 1 (Modeling)

The simple static var compensator (SVC) of Fig. 2 provides a practical illustration of the DAIS model formulation. This example will be revisited later to examine various algorithms for locating grazing phenomena. The parameters of this model are taken from [Jalali *et al.*, 1996]. For clarity of presentation we consider only a single thyristor, whereas [Jalali *et al.*, 1996] used a back-to-back pair.

Resistor R_s and inductor L_s characterize the transformer and source impedance, while R_r and L_r are the resistance and inductance of the inductor coil in series with the thyristor switch. The thyristor conducts current i_{Lr} only in the forward direction. It turns on when triggered by a firing pulse at firing angle α , and continues to conduct while i_{Lr} remains positive. It turns off (commutates) when i_{Lr} becomes zero. The thyristor is assumed ideal, so detailed nonlinearities in the on/off process are neglected.

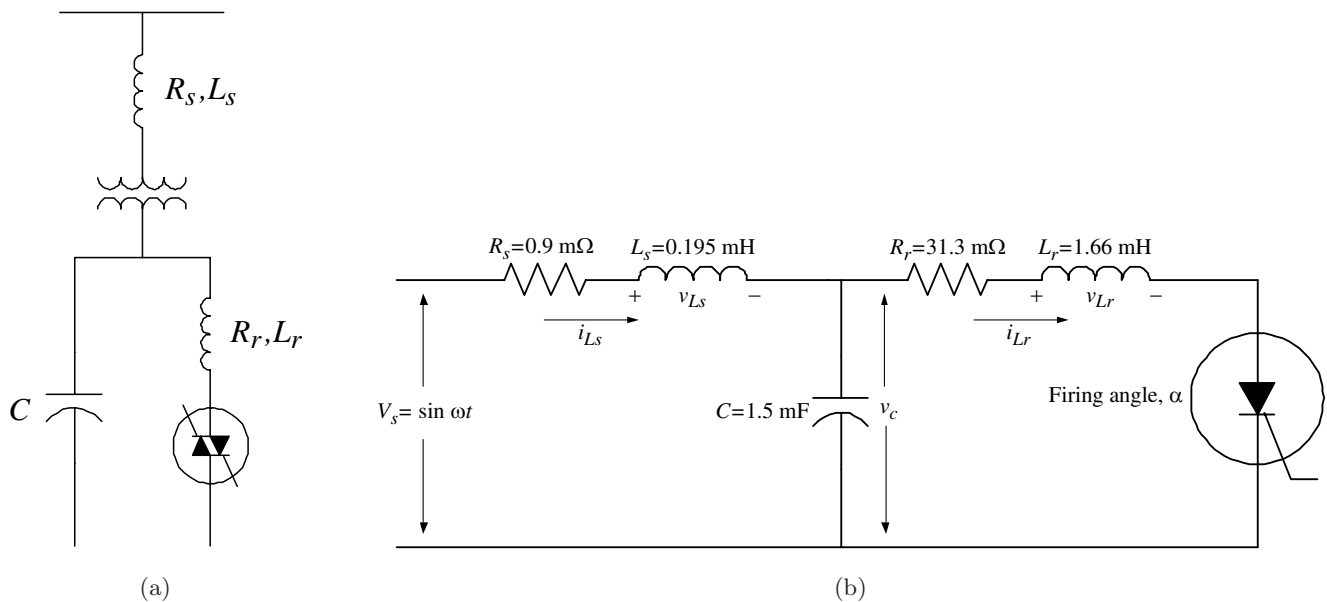


Fig. 2. Static var compensator (SVC). (a) One-line diagram. (b) Circuit representation.

The model consists of differential equations

$$\frac{d}{dt} \begin{bmatrix} x_t \\ i_{Ls} \\ i_{Lr} \\ v_c \\ x_{rr} \\ x_{stat} \end{bmatrix} = \begin{bmatrix} 1 \\ \frac{v_{Ls}}{L_s} \\ \frac{v_{Lr}}{L_r} \\ \frac{i_c}{C} \\ 0 \\ 0 \end{bmatrix} \quad (13)$$

together with algebraic equations

$$0 = R_s i_{Ls} + v_{Ls} + v_c - \sin(\omega x_t) \quad (14)$$

$$0 = i_{Ls} - i_c - i_{Lr} \quad (15)$$

$$y_{on} = \frac{\omega x_t}{\pi} - x_{rr} - \frac{\alpha}{180} \quad (16)$$

$$\left. \begin{array}{l} v_{Lr} = 0 \\ y_{off} = -1 \end{array} \right\} \begin{array}{l} x_{stat} < 0 \\ \text{(thyristor open circuit)} \end{array} \quad (17)$$

$$\left. \begin{array}{l} v_c - R_r i_{Lr} - v_{Lr} = 0 \\ y_{off} = i_{Lr} \end{array} \right\} \begin{array}{l} x_{stat} > 0 \\ \text{(thyristor conducting)} \end{array} \quad (18)$$

and state reset equations

$$\left. \begin{array}{l} x_{rr}^+ = x_{rr}^- + 2 \\ x_{stat}^+ = x_{stat}^- + 2 \end{array} \right\} \text{when } y_{on} = 0 \quad (19)$$

$$x_{stat}^+ = x_{stat}^- - 2 \quad \text{when } y_{off} = 0. \quad (20)$$

Model behavior can be best explained by referring to Fig. 3. Assume that initial time $x_t = 0$, and the thyristor is in blocking (open circuit) mode, that is $x_{stat} = -1$. Initially $x_{rr} = 0$, so according to (16), the turn-on indicator $y_{on} < 0$. Subsequently y_{on} increases linearly with time until a reset event (19) is triggered when $y_{on} = 0$. At that event, the thyristor status x_{stat} toggles from -1 to $+1$, i.e. from blocking to conducting. Also, a step occurs in x_{rr} , forcing $y_{on} < 0$ again. While the thyristor is conducting, the turn-off indicator y_{off} monitors the

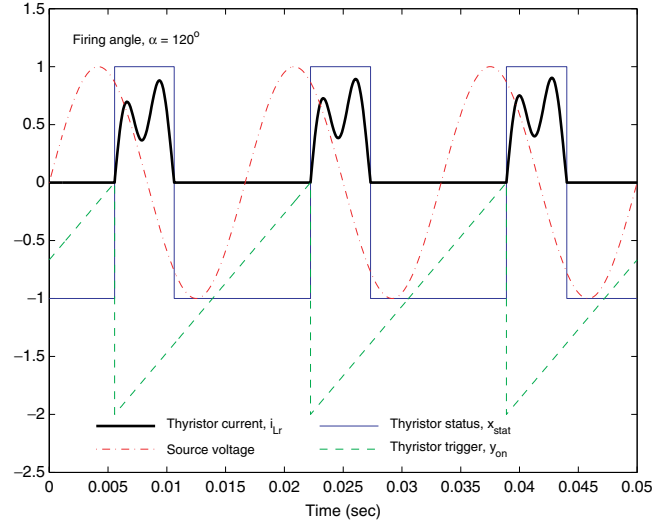


Fig. 3. SVC model behavior.

thyristor current i_{Lr} . If i_{Lr} drops to zero, reset event (20) toggles x_{stat} from $+1$ to -1 , and the thyristor self commutates (blocks).

In order to motivate later grazing analysis, Fig. 4 shows thyristor current i_{Lr} for various values of firing angle. It can be seen that as the firing angle increases from $\alpha = 120^\circ$ to 125° , the dip during the first conduction period distorts towards zero. Somewhere between 125° and 130° , the increasing distortion causes the current to fall to zero, and the thyristor prematurely commutates. The pivotal case, separating the two qualitatively different forms of behavior, corresponds to grazing. As the firing angle increases further, another grazing situation occurs between 130° and 135° , with premature commutation occurring during the second conduction period.

2.3. Numerical integration

Predicting the dynamic response of a system, for a given set of initial conditions, generally requires numerical integration. The accuracy of the resulting approximate trajectory depends on factors such as integration technique, time step and round-off error [Parker & Chua, 1989; Shampine, 1994]. Gross errors, including numerical instability, can be avoided by using implicit integration with an appropriate time step. However, some inaccuracy will always remain. In cases where event triggering hypersurfaces are encountered transversally, slight inaccuracy in the predicted trajectory will have

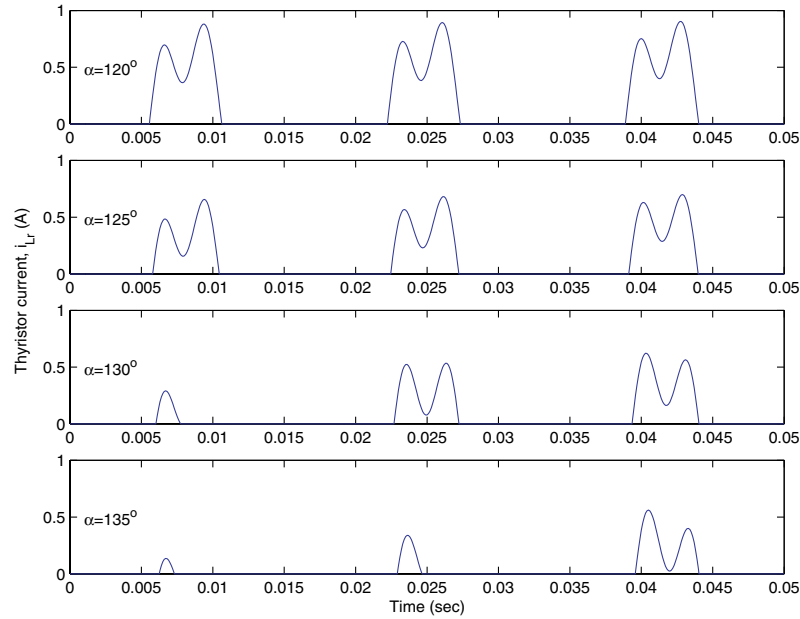


Fig. 4. Thyristor current waveforms for various firing angles.

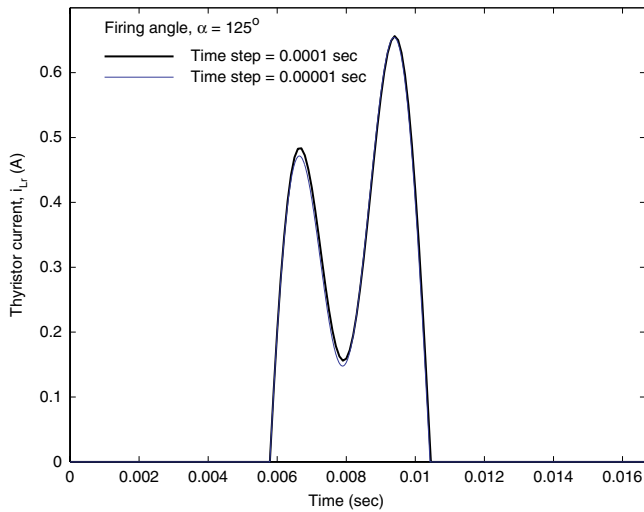


Fig. 5. Integration time step comparison.

only a minor effect on the triggering point.⁴ However in grazing situations, where the encounter is tangential, behavior will be as sensitive to simulation parameters as it is to true system parameters.

⁴The point where the trajectory encounters the triggering hypersurface can be accurately located by adapting the implicit integration solver. Normal progression from time t_k to $t_{k+1} = t_k + t_{step}$, for a given time-step t_{step} , is achieved by solving a (typically nonlinear) implicit algebraic equation of the form,

$$F_{int}(x_k, y_k, x_{k+1}, y_{k+1}, t_{step}) = 0 \tag{21}$$

for x_{k+1}, y_{k+1} . When locating a triggering point, t_{step} becomes a variable, and (21) is augmented by the triggering hypersurface equation $s(x_{k+1}, y_{k+1}) = 0$.

To illustrate, Fig. 5 shows the first cycle of SVC inductor current i_{Lr} , for a firing angle $\alpha = 125^\circ$. Trapezoidal integration was used to obtain this trajectory; the responses predicted using two different time steps are shown. The two trajectories are almost identical, except for slight divergence around the peaks and troughs. However referring to Fig. 4, it is the trough that undergoes grazing as the firing angle α is increased. The predicted grazing value of α will therefore depend (slightly) on the integration time step.

There are certainly integration techniques that are more accurate than trapezoidal, so numerical inaccuracies can be further reduced. However the tangential contact implicit in grazing suggests that any inaccuracy will (slightly) alter the grazing point prediction.

3. Background

3.1. Trajectory sensitivities

Shooting method algorithms, which form the basis for locating grazing phenomena, require the

sensitivity of a trajectory to perturbations in parameters and/or initial conditions [Stoer & Bulirsch, 1993]. To obtain the sensitivity of the flows ϕ and ψ to initial conditions x_0 , the Taylor series expansions of (9)–(10) are formed. Neglecting higher order terms gives

$$\Delta x(t) = \frac{\partial x(t)}{\partial x_0} \Delta x_0 \equiv \Phi(x_0, t) \Delta x_0 \quad (22)$$

$$\Delta y(t) = \frac{\partial y(t)}{\partial x_0} \Delta x_0 \equiv \Psi(x_0, t) \Delta x_0 \quad (23)$$

where Φ and Ψ are the *sensitivity transition matrices*, or *trajectory sensitivities*, associated with the x and y flows, respectively [Frank, 1978]. Equations (22)–(23) describe the changes $\Delta x(t)$ and $\Delta y(t)$ in a trajectory, at time t along the trajectory, for a given (small) change in initial conditions $\Delta x_0 = [\Delta \underline{x}_0^T \ \Delta p^T]^T$. In accordance with the partitioning (6), Φ can be decomposed as

$$\Phi = \begin{bmatrix} \Phi_x & \Phi_p \\ 0 & I_\ell \end{bmatrix} \quad (24)$$

where I_ℓ is the $\ell \times \ell$ identity matrix.

An overview of the variational equations describing the evolution of these sensitivities is provided in Appendix A. It should be emphasized that Φ and Ψ do not require smoothness of the underlying flows ϕ and ψ . As shown in Appendix A, they are well defined for nonsmooth and/or discontinuous flows generated by hybrid systems.

3.2. Limit cycles

3.2.1. Poincaré maps

Periodic behavior of limit cycles implies that the system state returns to its initial value every cycle. This can be expressed in terms of the flow as

$$x^* = \phi(x^*, T) \quad (25)$$

where T is the limit cycle period.⁵ For non-autonomous systems, the period T is a known quantity. However it is not known *a priori* for autonomous systems. The unknown period, or *return time*, can be found using Poincaré map concepts [Parker & Chua, 1989; Seydel, 1994]. These concepts are well known; the following summary is provided as it underlies later analysis of grazing bifurcations in periodic systems.

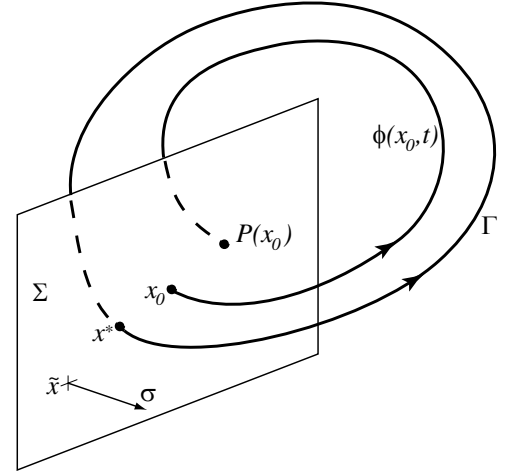


Fig. 6. Poincaré map.

Referring to Fig. 6, let Σ be a hyperplane that is transversal to the flow $\phi(x_0, t)$, and defined by

$$\Sigma = \{x : \sigma^T(x - \tilde{x}) = 0\} \quad (26)$$

where \tilde{x} is a point anchoring Σ , and σ is a vector normal to Σ . The return time τ_r for a trajectory emanating from $x_0 \in \Sigma$ is therefore given by

$$\sigma^T(\phi(x_0, \tau_r) - \tilde{x}) = 0. \quad (27)$$

Because the flow $\phi(x_0, t)$ transversally encounters Σ at τ_r , the tangent to the flow at that intersection point

$$\frac{\partial \phi}{\partial t}(\tau_r) = f(x(\tau_r), y(\tau_r)) \equiv f|_{\tau_r} \quad (28)$$

satisfies

$$\sigma^T f|_{\tau_r} \neq 0. \quad (29)$$

By differentiating (27) with respect to x_0 and τ_r ,

$$\sigma^T \Phi(x_0, \tau_r) dx_0 + \sigma^T f|_{\tau_r} d\tau_r = 0, \quad (30)$$

and taking account of the transversality condition (29), it can be seen that the implicit function theorem [Fleming, 1977] guarantees the existence of a function $\tau_r(x_0)$ which (locally) satisfies (27),

$$\sigma^T(\phi(x_0, \tau_r(x_0)) - \tilde{x}) = 0. \quad (31)$$

The flow ϕ and hyperplane Σ together describe a Poincaré map $P : \Sigma \rightarrow \Sigma$, defined by

$$P(x_0) = \phi(x_0, \tau_r(x_0)) \quad (32)$$

where $\tau_r(\cdot)$ is given (implicitly) by (27). Therefore from (25), a limit cycle of an autonomous system must satisfy

$$x^* = P(x^*) = \phi(x^*, \tau_r(x^*)). \quad (33)$$

The corresponding limit cycle is labeled Γ in Fig. 6.

⁵It is assumed throughout that the algebraic equations $g(x, y) = 0$ are always satisfied.

For a nonautonomous system, the Poincaré map $P : \mathbb{R}^n \rightarrow \mathbb{R}^n$ becomes simply

$$P(x_0) = \phi(x_0, T), \tag{34}$$

with limit cycles given by $x^* = P(x^*)$.

3.2.2. Locating limit cycles

Limit cycles can be located by solving (25) for nonautonomous systems or (33) for autonomous systems. In the case of nonautonomous systems, rewriting (25) gives

$$F_l(x^*) = \phi(x^*, T) - x^* = 0. \tag{35}$$

The solution x^* can be obtained using a shooting method,⁶ which solves the iterative scheme

$$x^{k+1} = x^k - (DF_l(x^k))^{-1} F_l(x^k) \tag{36}$$

where

$$DF_l(x^k) = \Phi(x^k, T) - I \tag{37}$$

and I is the n -dimensional identity matrix.

For autonomous systems, rewriting (33) gives

$$F_l(x^*) = \phi(x^*, \tau_r(x^*)) - x^* = 0. \tag{38}$$

Again (36) can be used to find the solution x^* . In this case

$$DF_l(x^k) = \Phi(x^k, \tau_r(x^k)) + f|_{\tau_r(x^k)} \frac{d\tau_r}{dx}(x^k) - I. \tag{39}$$

The term $d\tau_r/dx$ in (39) can be obtained from (30), giving

$$\begin{aligned} DF_l(x^k) &= \Phi(x^k, \tau_r(x^k)) \\ &\quad - f|_{\tau_r(x^k)} \frac{\sigma^T \Phi(x^k, \tau_r(x^k))}{\sigma^T f|_{\tau_r(x^k)}} - I \tag{40} \\ &= \left(I - \frac{f|_{\tau_r(x^k)} \sigma^T}{\sigma^T f|_{\tau_r(x^k)}} \right) \Phi(x^k, \tau_r(x^k)) - I. \tag{41} \end{aligned}$$

Recall from Appendix A that the sensitivity transition matrix Φ in (37) and (41) is well defined for hybrid (piecewise smooth) systems. Therefore, the proposed shooting method is suitable for piecewise smooth limit cycles. This will be illustrated in later examples.

3.2.3. Parameters within the state

It can be seen from (9) that the last ℓ states within x , those corresponding to parameters p , always remain constant. Therefore the last ℓ equations in (35) or (38) are always satisfied by default, and hence are redundant. Discarding those ℓ equations results in the under-constrained problem,

$$\underline{F}_l(x) = 0 \tag{42}$$

where $\underline{F}_l : \mathbb{R}^n \rightarrow \mathbb{R}^{n-\ell}$. Furthermore, (24) indicates that the last ℓ rows of Φ are identical to the corresponding rows of the identity matrix I . Therefore from (37),

$$DF_l = \begin{bmatrix} D\underline{F}_l \\ 0_{\ell \times n} \end{bmatrix} \tag{43}$$

where $0_{\ell \times n}$ refers to the $\ell \times n$ matrix of zeros. Likewise, it is straightforward to show that DF_l in (41) also has the structure (43).

It follows that solutions of (35) or (38) are well defined only when $\ell = 0$, i.e. no parameters are incorporated into x . In that case $F_l \equiv \underline{F}_l$. It will be shown in Sec. 5.1 though that periodic grazing requires $\ell = 1$, and in Sec. 6 that continuation requires $\ell = 2$. Closest grazing point concepts, developed in Sec. 7, require $\ell \geq 2$.

4. Transient Grazing

4.1. Mathematical description

Grazing is characterized by a trajectory (flow) of the system touching a triggering hypersurface tangentially. Let the target hypersurface be described by

$$b(x, y) = 0 \tag{44}$$

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

$$[b_x \ b_y] \begin{bmatrix} u \\ v \end{bmatrix} = 0. \tag{45}$$

The vector $[x^T \ y^T]^T$ is directed tangentially along the flow, so it must satisfy (45) at a grazing point. Furthermore, differentiating (8) and

⁶Reformulation as a multiple shooting method [Stoer & Bulirsch, 1993] is straightforward. Likewise for the grazing shooting methods discussed later.

substituting (7) gives,

$$0 = \frac{\partial g}{\partial x} \dot{x} + \frac{\partial g}{\partial y} \dot{y} \quad (46)$$

$$\Rightarrow 0 = g_x f(x, y) + g_y v \quad (47)$$

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

A single degree of freedom is available for varying parameters (and initial conditions) to find a grazing point. This single degree of freedom can be achieved by parametrization $x_0(\theta)$, where θ is a scalar. Grazing points are then described by combining together the flow definition (9) (appropriately parameterized by θ), algebraic equations (8), target hypersurface (44), and tangency conditions (45), (47), to give

$$F_{g1}(x_g, \theta, t_g) := \phi(x_0(\theta), t_g) - x_g = 0 \quad (48)$$

$$F_{g2}(x_g, y_g) := g(x_g, y_g) = 0 \quad (49)$$

$$F_{g3}(x_g, y_g) := b(x_g, y_g) = 0 \quad (50)$$

$$F_{g4}(x_g, y_g, v) := \begin{bmatrix} b_x & b_y \\ g_x & g_y \end{bmatrix}_{(x_g, y_g)} \begin{bmatrix} f(x_g, y_g) \\ v \end{bmatrix} = 0. \quad (51)$$

The grazing encounter occurs at time t_g along the trajectory, and its state-space location is given by x_g, y_g . This set of equations may be written compactly as

$$F_g(x_g, y_g, \theta, t_g, v) = F_g(z) = 0 \quad (52)$$

where $F_g : \mathbb{R}^{n+2m+2} \rightarrow \mathbb{R}^{n+2m+2}$ and $z = [x_g^T \ y_g^T \ \theta \ t_g \ v^T]^T$. Solution of (52) can be achieved using a shooting method, as discussed in the following section.

4.2. Shooting method

4.2.1. Algorithm

Numerical solution of (52) using Newton's method amounts to iterating on the standard update formula

$$z^{k+1} = z^k - (DF_g(z^k))^{-1} F_g(z^k) \quad (53)$$

where DF_g is the Jacobian matrix

$$DF_g = \begin{bmatrix} -I & 0 & \Phi \frac{dx_0}{d\theta} & f & 0 \\ g_x & g_y & 0 & 0 & 0 \\ b_x & b_y & 0 & 0 & 0 \\ f^t b_{xx} + b_x f_x + v^t b_{yx} & f^t b_{xy} + b_x f_y + v^t b_{yy} & 0 & 0 & b_y \\ \hat{f}^t g_{xx} + g_x f_x + \hat{v}^t g_{yx} & \hat{f}^t g_{xy} + g_x f_y + \hat{v}^t g_{yy} & 0 & 0 & g_y \end{bmatrix} \quad (54)$$

with

$$\hat{f} = \begin{bmatrix} f & & & & \\ & f & & & \\ & & f & & \\ & & & \ddots & \\ & & & & f \end{bmatrix} \in \mathbb{R}^{mn \times m}, \quad \hat{v} = \begin{bmatrix} v & & & & \\ & v & & & \\ & & v & & \\ & & & \ddots & \\ & & & & v \end{bmatrix} \in \mathbb{R}^{m^2 \times m} \quad (55)$$

$$g_{xx} = \begin{bmatrix} \frac{\partial^2 g_1}{\partial x^2} \\ \frac{\partial^2 g_2}{\partial x^2} \\ \vdots \\ \frac{\partial^2 g_m}{\partial x^2} \end{bmatrix} \in \mathbb{R}^{mn \times n}, \quad g_{yx} = \begin{bmatrix} \frac{\partial^2 g_1}{\partial y \partial x} \\ \frac{\partial^2 g_2}{\partial y \partial x} \\ \vdots \\ \frac{\partial^2 g_m}{\partial y \partial x} \end{bmatrix} \in \mathbb{R}^{m^2 \times n} \quad (56)$$

$$g_{yy} = \begin{bmatrix} \frac{\partial^2 g_1}{\partial y^2} \\ \frac{\partial^2 g_2}{\partial y^2} \\ \vdots \\ \frac{\partial^2 g_m}{\partial y^2} \end{bmatrix} \in \mathbb{R}^{m^2 \times m}, \quad g_{xy} = \begin{bmatrix} \frac{\partial^2 g_1}{\partial x \partial y} \\ \frac{\partial^2 g_2}{\partial x \partial y} \\ \vdots \\ \frac{\partial^2 g_m}{\partial x \partial y} \end{bmatrix} \in \mathbb{R}^{mn \times m}. \quad (57)$$

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_g 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 target hypersurface. The example of Sec. 4.4 provides an illustration. Two approaches have been used to obtain the second derivative terms:

1. Numerical differencing. Many simulators provide direct computation⁷ of g_x and g_y . 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 modeling structure [Hiskens & Sokolowski, 2001], 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.

Care must be taken in evaluating the terms of (52) and (54) that relate to trajectory solution. The flow term $\phi(x_0(\theta^k), t_g^k)$ in (48) evaluates, via numerical integration, to the value of x at time t_g^k along the trajectory that has initial value $x_0(\theta^k)$. Likewise, the terms Φ and f in the first row of DF_g should also be evaluated at time t_g^k along that trajectory. All other terms in DF_g should be evaluated at x_g^k, y_g^k .

4.2.2. Initialization of variables

As with all iterative procedures, solution of (53) requires a good initial guess z^0 . In terms of

the original system variables, initial (approximate) grazing point values of $x_g^0, y_g^0, \theta^0, t_g^0$, and v^0 are required. These can be obtained from regular simulation.

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

1. a 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 $db/dt = 0$, which implies $b_x \dot{x} + b_y \dot{y} = 0$. Substituting for \dot{y} from (46), and using (7), 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_g^0, y_g^0 , and t_g^0 . The corresponding value of v^0 can be obtained from (47) as

$$v^0 = -g_y^{-1} g_x f(x_g^0, y_g^0)$$

with all partial derivatives evaluated at x_g^0, y_g^0 .

4.3. Example 2 (Switching-time bifurcation)

Referring to Fig. 4, variation of the thyristor firing angle α suggests that a switching-time bifurcation [Rajaraman *et al.*, 1996; Jalali *et al.*, 1996] occurs between 125° and 130° . The triggering hypersurface in this case is given by $i_{Lr} = 0$. The shooting method (53) was used to determine the conditions that gave rise to grazing. For illustration purposes, an initial parameter value of $\alpha = 120^\circ$ was chosen, even though it was known that the actual grazing value α^* lay in the range $125^\circ < \alpha^* < 130^\circ$. The

⁷These quantities are required for implicit numerical integration.

Table 1. Shooting method convergence, first conduction period.

Iteration	Grazing Values	
	Firing Angle, α	Time, t_g
0	120.00	0.007900
1	128.28	0.007954
2	129.39	0.007947
3	129.43	0.007946

solution process converged in three iterations, with convergence progress given in Table 1.

Figure 7 shows the behavior of i_{Lr} for the initial guess $\alpha = 120^\circ$ and the grazing value $\alpha^* = 129.43^\circ$. It can be seen that grazing occurred at time $t_g^* = 0.007946$ sec, when the dip during the first conduction period dropped to $i_{Lr} = 0$. Behavior is ill-defined at that point; i_{Lr} could continue as shown in Fig. 7, or the thyristor could commutate and i_{Lr} remain at zero. Exact behavior would be governed by unmodeled effects.

Figure 4 also indicates that a switching-time bifurcation can occur during the second conduction period, for a value of firing angle in the range $130^\circ < \alpha^* < 135^\circ$. The shooting method (53) was again used to find this grazing value. An initial guess of $\alpha = 120^\circ$ was used, with solution progress given in Table 2. Even though the initial guess for α was (deliberately) quite poor, convergence was obtained in three iterations. Grazing occurs at $t_g^* = 0.02499$ sec, with a firing angle $\alpha^* = 132.66^\circ$.

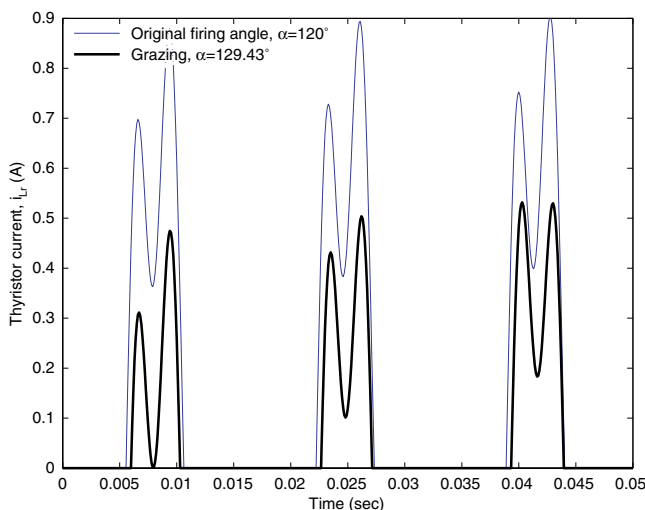

 Fig. 7. Comparison of original and grazing behaviors of i_{Lr} .

Table 2. Shooting method convergence, second conduction period.

Iteration	Grazing Values	
	Firing Angle, α	Time, t_g
0	120.00	0.02460
1	131.89	0.02487
2	132.47	0.02498
3	132.66	0.02499

4.4. Example 3 (Performance specification)

Performance specifications often place bounds on transient excursions of system quantities. Determining parameter values that ensure the specifications are satisfied can be formulated as a grazing problem (52). The single machine infinite bus power system of Fig. 8 will be used to illustrate this application.

For this example, the generator was accurately represented by a sixth-order machine model,⁸ and the generator excitation system was modeled according to Fig. 9. Note that the output limits on the field voltage E_{fd} are anti-wind-up limits, while the limits on the stabilizer output V_{PSS} are clipping limits [Goodwin *et al.*, 2001]. Therefore even though this example utilizes a simple network structure, it exhibits nonlinear, nonsmooth, hybrid system behavior. 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.

Generators are susceptible to over-voltage protection operation if their terminal voltage rises too high. This may occur during transients following a large disturbance. The field voltage maximum limit E_{fdmax} has a large influence on transient over-voltages. Therefore, this example considers the maximum value of E_{fdmax} that ensures the initial terminal voltage overshoot does not rise above a

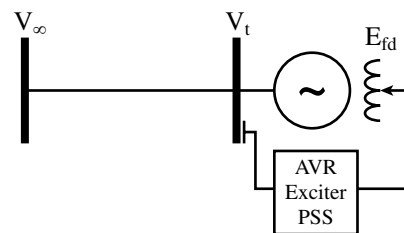


Fig. 8. Single machine infinite bus system.

⁸The sixth-order machine model consists of two axes, with two windings in each axis [Sauer & Pai, 1998].

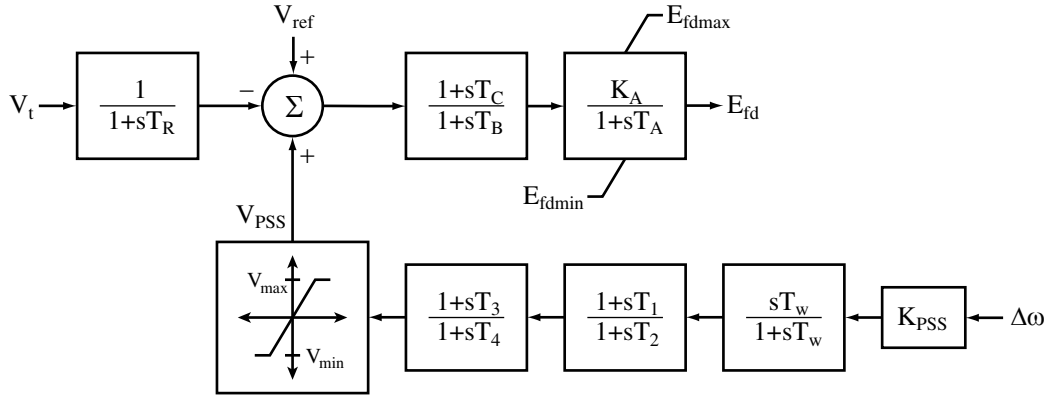


Fig. 9. Excitation system (AVR/PSS) representation.

specified value of 1.2 pu. The target hypersurface in this case is therefore $V_t - 1.2 = 0$.

Results of the iterative process are given in Table 3, and presented graphically in Figs. 10 and 11. Convergence of the shooting method was achieved in four iterations. This is an encouraging result, as an onerous test condition was chosen.

Table 3. Shooting method convergence.

Iteration	Grazing Values	
	Param, E_{fdmax}	Time, t_g
0	5.80	1.14
1	3.16	1.28
2	4.37	1.19
3	4.72	1.20
4	4.78	1.21

Referring to Fig. 10, it can be seen that the original voltage trajectory is quite flat over the first extended peak. The grazing formulation (52) not only describes peaks, but also troughs and points of inflection. It turns out that for this example, (52) actually has three solutions in close proximity. Accordingly, the Jacobian DF_g is quite ill-conditioned. It was found that if the second derivative terms in (52) were ignored, the shooting method converged, but to the wrong solution. This occurred for initial conditions over most of the extended peak. Clearly the directional information provided by the second derivative terms is important in cases such as this, where the encounter between the trajectory and the border is not unimodal.

It is evident from Figs. 10 and 11 that this system exhibits quite nonsmooth behavior. In fact

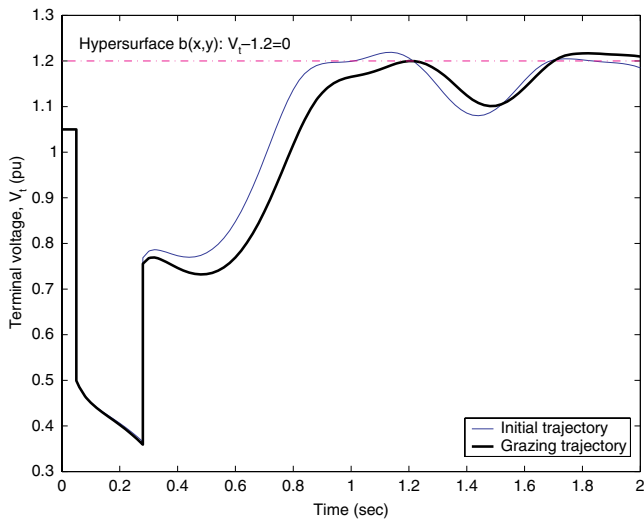


Fig. 10. Terminal voltage, V_t .

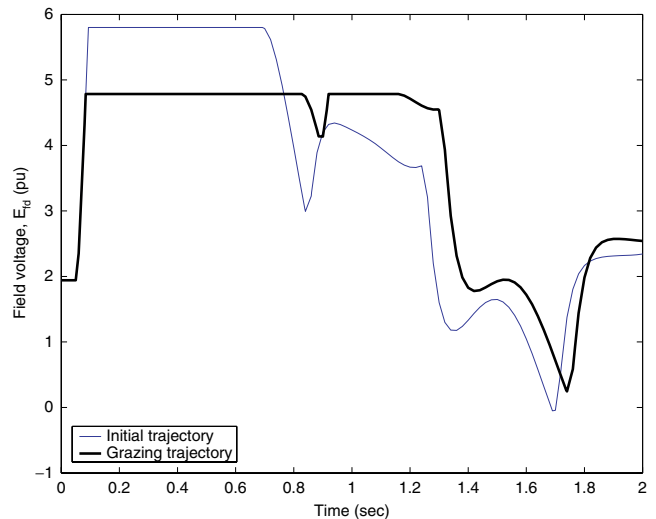


Fig. 11. Generator field voltage, E_{fd} .

fifteen events occur over the initial 2sec transient, primarily V_{PSS} banging on maximum and minimum limits. Discrete events clearly exert a strong influence on system dynamics. However because trajectory sensitivities, and hence the Jacobian DF_g , take those events into account, shooting method convergence is unaffected.

Notice in Fig. 10 that enforcing the performance specification at the first peak has a detrimental effect on the second peak. This reflects the fact that the grazing formulation (52) solves for local encounters between the trajectory and the tangent hypersurface. However the specification could be enforced at both peaks by freeing a second parameter, and coupling together two sets of equations of the form (48)–(51).

5. Periodic Grazing

5.1. Algorithm

For a grazing bifurcation to occur in a periodic system, the conditions governing limit cycles, (35) or (38), and grazing (52) must be jointly satisfied. This requires more degrees of freedom than was the case for transient grazing. In order to locate a limit cycle, the initial point x_0 must be free to vary. Also, grazing point solution requires one free parameter. Computational and notational advantages follow from a consistent treatment of all variables. Therefore, throughout subsequent developments, parameters

$p \in \mathbb{R}^\ell$ will be incorporated into x_0 in accordance with (6).

To allow for the extra degrees of freedom, (48) should be restated,

$$F'_{g1}(x_g, x_0, t_g) := \phi(x_0, t_g) - x_g = 0. \tag{58}$$

Also, recall from Sec. 3.2.3 that with parameters p incorporated into x , the equations describing limit cycles take the form (42). Assembling the full set of equations governing grazing and limit cycles gives

$$F_{gl}(z) \equiv F_{gl}(x_g, y_g, x_0, t_g, v) := \begin{bmatrix} F'_{g1}(x_g, x_0, t_g) \\ F_{g2}(x_g, y_g) \\ F_{g3}(x_g, y_g) \\ F_{g4}(x_g, y_g, v) \\ \underline{F}_l(x_0) \end{bmatrix} = 0 \tag{59}$$

where $z = [x_g^T \ y_g^T \ x_0^T \ t_g \ v^T]^T$ and $F_{gl} : \mathbb{R}^{2n+2m+1} \rightarrow \mathbb{R}^{2n+2m+2-\ell}$. Point solutions of (59) require a match between the numbers of equations and variables. This is achieved when $\ell = 1$, i.e. a single free parameter. When $\ell > 1$, (59) describes a continuum of solutions. That situation arises in Secs. 6 and 7.

As in the case of transient grazing, numerical solution of (59) using Newton’s method amounts to iterating on the standard update formula

$$z^{k+1} = z^k - (DF_{gl}(z^k))^{-1} F_{gl}(z^k) \tag{60}$$

where now DF_{gl} is the Jacobian matrix

$$DF_{gl} = \begin{bmatrix} -I & 0 & \Phi & f & 0 \\ g_x & g_y & 0 & 0 & 0 \\ b_x & b_y & 0 & 0 & 0 \\ f^t b_{xx} + b_x f_x + v^t b_{yx} & f^t b_{xy} + b_x f_y + v^t b_{yy} & 0 & 0 & b_y \\ \hat{f}^t g_{xx} + g_x f_x + \hat{v}^t g_{yx} & \hat{f}^t g_{xy} + g_x f_y + \hat{v}^t g_{yy} & 0 & 0 & g_y \\ 0 & 0 & \underline{DF}_l & 0 & 0 \end{bmatrix} \tag{61}$$

with \hat{f} , \hat{v} , and the second-order partial derivatives defined by (55)–(57). The term \underline{DF}_l is defined by (43).

As noted in Sec. 4.2.1, care must be given to evaluating terms of (59) and (61) that relate to trajectory solution. Referring to (59), the flow term $\phi(x_0^k, t_g^k)$ in F'_{g1} evaluates, via numerical integration, to the value of x at time t_g^k along the trajectory initialized at x_0^k . The associated Jacobian entries Φ

and f in the first row of DF_{gl} should also be evaluated at t_g^k along that trajectory. Similarly, \underline{F}_l in (59) involves the flow term $\phi(x_0^k, T)$ or $\phi(x_0^k, \tau_r(x_0^k))$ for nonautonomous or autonomous systems respectively. The former evaluates to x at time T along the trajectory initiated from x_0^k . The latter gives x at the point where the trajectory, starting from $x_0^k \in \Sigma$, re-encounters Σ .⁹ The associated Jacobian

⁹This occurs at return time $\tau_r(x_0^k)$, though that time is not explicitly determined.

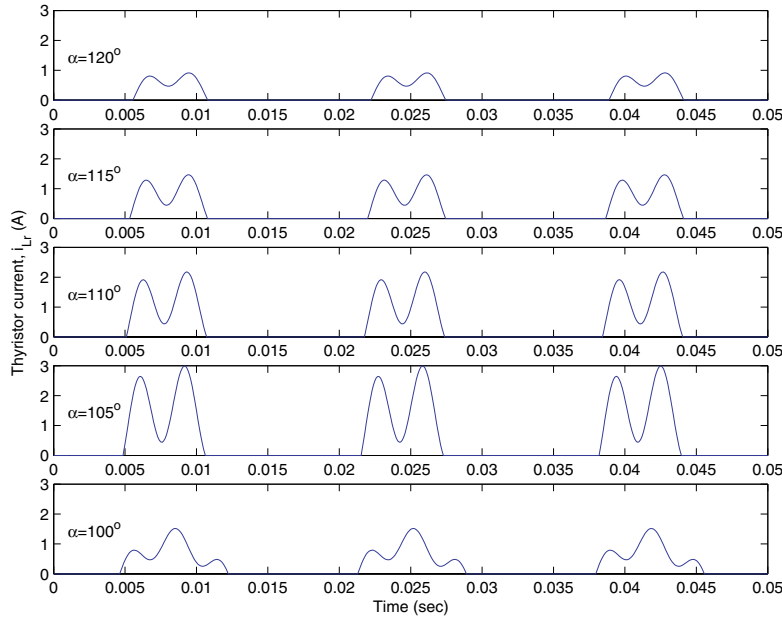


Fig. 12. SVC steady state behavior over a range of firing angles.

entry DF_l , which involves Φ , should be evaluated at time T , or the return point, accordingly.

5.2. Example 4 (Nonautonomous limit cycles)

The SVC of Sec. 2.2 was previously used (in Sec. 4.3) to illustrate transient grazing. In steady state, the SVC exhibits (nonautonomous) limit cycle behavior.¹⁰ Figure 12 shows steady state operation for a range of firing angles α . These limit cycles were located using the shooting method (36).

It is apparent from Fig. 12 that the limit cycles for $\alpha = 100^\circ$ and $\alpha = 105^\circ$ have structurally different forms. A grazing bifurcation occurs somewhere between these two cases. The shooting method of Sec. 5.1 was used to determine the value of α^* corresponding to this grazing bifurcation. Initial conditions were given by the limit cycle with $\alpha = 100^\circ$. Convergence progress is reported in Table 4. The bifurcation occurs at $\alpha^* = 102.16^\circ$, and is shown in Fig. 13, along with the initial condition limit cycle.

Figure 13 also shows the post-bifurcation limit cycle that develops when α is incrementally perturbed above the bifurcation value. Interestingly, this single-trough limit cycle coexists with the double-trough version over the range $97.73^\circ < \alpha < 102.16^\circ$. Both coexisting limit cycles are stable,

Table 4. Shooting method convergence to grazing bifurcation.

Iteration	Limit Cycle Initial Point, x_0		Grazing Values	
	i_{Ls}	v_c	Firing Angle, α	Time, t_g
0	3.8462	-0.5853	100.00	0.01050
1	4.6489	-0.9602	102.13	0.01043
2	4.6628	-0.9640	102.16	0.01043
3	4.6627	-0.9640	102.16	0.01043

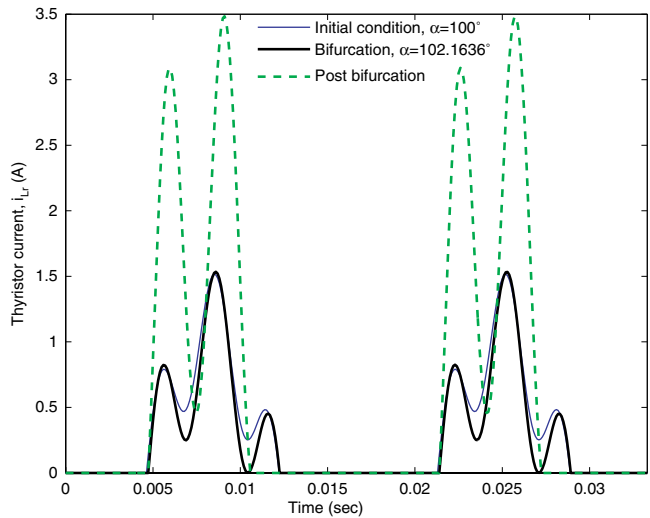


Fig. 13. SVC grazing bifurcation.

¹⁰The SVC model is nonautonomous, due to the time-dependent source.

though the region of attraction for the single-trough one is quite small. At $\alpha = 97.73^\circ$, the single-trough limit cycle undergoes its own grazing bifurcation and vanishes.

5.3. Example 5 (Autonomous limit cycles)

5.3.1. Robot model

A model of the compass gait biped robot is discussed in detail in [Goswami *et al.*, 1998]. A summary is included here for completeness. The biped robot can be treated as a double pendulum, with point masses m_H and m concentrated at the hips and legs, respectively. Figure 14 provides a schematic representation and identifies other important parameters: lengths a and b , and incline angle γ . The robot configuration is described by the *support angle* θ_s and the *nonsupport angle* θ_{ns} . The dynamic equations describing the robot can be written

$$M(\theta)\ddot{\theta} + N(\theta, \dot{\theta})\dot{\theta} + \frac{1}{a}g(\theta) = 0 \quad (62)$$

where $\theta = [\theta_{ns} \ \theta_s]^T$. The state vector is therefore $x = [\theta_{ns} \ \theta_s \ \dot{\theta}_{ns} \ \dot{\theta}_s]^T \in \mathbb{R}^4$. The matrix coefficients of (62) are given by,

$$M(\theta) = \begin{bmatrix} \beta^2 & -(1+\beta)\beta \cos \theta_d \\ -(1+\beta)\beta \cos \theta_d & (1+\beta)^2(\mu+1) + 1 \end{bmatrix}$$

$$N(\theta, \dot{\theta}) = \begin{bmatrix} 0 & (1+\beta)\beta \dot{\theta}_s \sin \theta_d \\ -(1+\beta)\beta \dot{\theta}_{ns} \sin \theta_d & 0 \end{bmatrix}$$

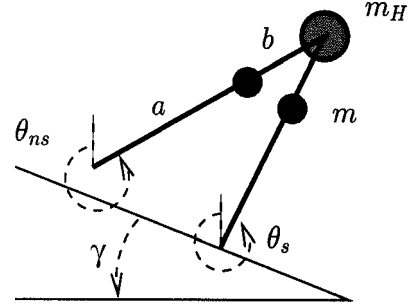


Fig. 14. Compass gait biped robot.

$$g(\theta) = \begin{bmatrix} g\beta \sin \theta_{ns} \\ -((\mu+1)(1+\beta) + 1)g \sin \theta_s \end{bmatrix}$$

where $\theta_d = \theta_s - \theta_{ns}$, $\beta = b/a$, $\mu = m_H/m$, and $g = 9.8$ is the gravitational constant. The model can be simply manipulated into the form (1)–(2), though the impulse effects are added below.

An event occurs when the nonsupport (swing-ing) leg collides with the ground. This establishes the triggering condition

$$\theta_{ns} + \theta_s + 2\gamma = 0. \quad (63)$$

At the event, the nonsupport leg becomes the support leg, and vice-versa. Velocities $\dot{\theta}_{ns}$ and $\dot{\theta}_s$ undergo step changes to ensure conservation of momentum through the collision. The resulting reset equations can be written

$$\begin{bmatrix} \theta \\ \dot{\theta} \end{bmatrix}^+ = \begin{bmatrix} \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} & 0 \\ 0 & Q^+(\theta_d^-)^{-1}Q^-(\theta_d^-) \end{bmatrix} \begin{bmatrix} \theta \\ \dot{\theta} \end{bmatrix}^- \quad (64)$$

where

$$Q^+(\theta_d) = \begin{bmatrix} \beta(\beta - (1+\beta)\cos \theta_d) & (1+\beta)((1+\beta) - \beta \cos \theta_d) + 1 + \mu(1+\beta)^2 \\ \beta^2 & -\beta(1+\beta)\cos \theta_d \end{bmatrix}$$

$$Q^-(\theta_d) = \begin{bmatrix} -\beta & -\beta + (\mu(1+\beta)^2 + 2(1+\beta))\cos \theta_d \\ 0 & -\beta \end{bmatrix}.$$

Equation (64) matches the form (4). The event triggering state y_r follows from (63) as $y_r = \theta_{ns} + \theta_s + 2\gamma$.

5.3.2. Results

To illustrate the shooting algorithm, a triggering hypersurface (border) was established by $b(x, y) = x_3 - 2.5 = 0$ where $x_3 \equiv \dot{\theta}_{ns}$. This could be interpreted as a maximum allowable nonsupport leg velocity. (Perhaps the robot falls apart at higher

velocities!) Walking motion that just satisfied this constraint was achieved by varying the incline angle γ . Solution progress is given in Table 5, and illustrated in Fig. 15. The figure shows the initial trajectory, and final grazing limit cycle.

The Poincaré hyperplane Σ is defined by normal vector $\sigma = [0 \ 0 \ 1 \ 0]^T$ and anchor point \tilde{x} equal to the initial guess for x_0 . This gave $\Sigma = \{x : x_3 = 0.1\}$. A projection of this hyperplane is shown in Fig. 15. The initial trajectory started

Table 5. Shooting method convergence to grazing bifurcation.

Iteration	Limit Cycle Initial Point, x_0				Grazing Values	
	θ_{ns}	θ_s	$\dot{\theta}_{ns}$	$\dot{\theta}_s$	Slope (deg), γ	Time, t_g
0	-0.3500	0.2100	0.1	-1.0000	3.00	0.2800
1	-0.4076	0.2253	0.1	-1.1408	4.66	0.2742
2	-0.4107	0.2223	0.1	-1.1278	5.07	0.2762
3	-0.4095	0.2220	0.1	-1.1215	4.99	0.2764

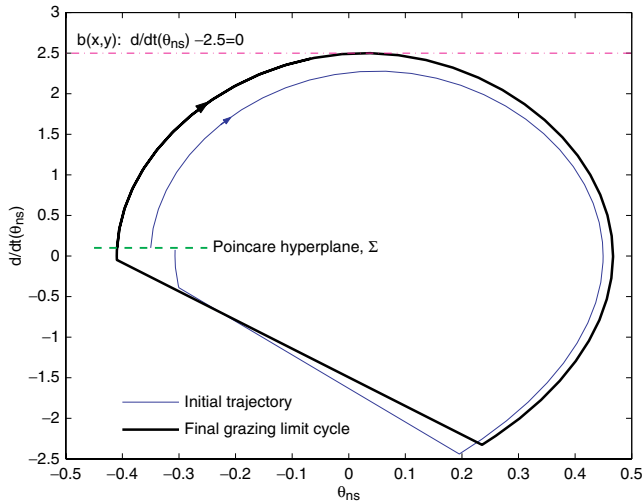


Fig. 15. Nonsupport leg angle θ_{ns} versus velocity $\dot{\theta}_{ns}$.

from, and returned to, this hyperplane. It was not a limit cycle. The final solution started from a point on the hyperplane and returned to that same point. Along the way it grazed the surface $b(x, y) = 0$.

The characteristic multipliers for the grazing limit cycle are: -1.549 , -0.1873 and 0.0970 . Because one of the eigenvalues lies outside the unit circle, this limit cycle is unstable. Even so, shooting method convergence was fast and reliable. On the other hand, locating this limit cycle by repeated simulations would be extremely difficult.

6. Continuation Method for Grazing Limit Cycles

6.1. Algorithm

When a single parameter is free to vary, (59) describes isolated grazing limit cycles. If two parameters are free, then $F_{gl} : \mathbb{R}^{2n+2m+1} \rightarrow \mathbb{R}^{2n+2m}$, so

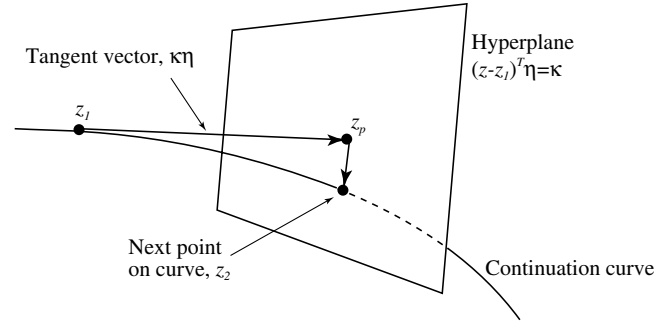


Fig. 16. Predictor-corrector process.

$F_{gl}(z) = 0$ describes a 1-manifold, or curve, of solutions. Such curves are useful for exploring parameter dependencies, but cannot be computed directly. Rather, a continuation process is required to generate successive points along the curve. This can be achieved robustly using an Euler homotopy algorithm [Garcia & Zangwill, 1981]. A brief description of this predictor-corrector process follows.

Assume the algorithm starts from a point z_1 on the curve.¹¹ The first step of the algorithm, as illustrated in Fig. 16, is the prediction of the next point on the curve. This is achieved by determining the unit vector that is tangent to the curve at z_1 , and moving along that vector at predefined distance κ . This κ is a (scalar) control parameter that effectively determines the distance between successive points along the curve. In regions of high curvature, κ may need to be small. When the curve is almost linear, a large value of κ would suffice. The unit vector $\eta \in \mathbb{R}^{2n+2m+1}$ that is tangent to the curve at z_1 is given by,

$$DF_{gl}(z_1)\eta = 0 \tag{65}$$

$$\|\eta\|_2 = 1 \tag{66}$$

where DF_{gl} is the Jacobian (61) which, because of the two free parameters, now has dimension

¹¹This point would be obtained by solving (59), as discussed in Sec. 5.1.

$(2n + 2m) \times (2n + 2m + 1)$.¹² The prediction of the next point on the curve is given by,

$$z_p = z_1 + \kappa\eta. \quad (67)$$

Having found the prediction point, the next step is to correct to a point z_2 on the curve. The Euler homotopy method does this by solving for the point of intersection of the curve and a hyperplane which passes through z_p and that is orthogonal to η . Points z on this hyperplane are given by,

$$(z - z_p)^T \eta = 0 \quad (68)$$

or alternatively

$$(z - z_1)^T \eta = \kappa. \quad (69)$$

Either (68) or (69) can be used. The point of intersection of the curve and the hyperplane is then given by,

$$F_{gl}^{\text{cont}}(z) = \begin{bmatrix} F_{gl}(z) \\ (z - z_1)^T \eta - \kappa \end{bmatrix} = 0. \quad (70)$$

Note that in (70), z_1 , η and κ are fixed, with z being the only unknown. The first $2n + 2m$ equations ensure the point is on the desired curve. The last scalar equation ensures the point is on the hyperplane. Altogether (70) forms a set of $2n + 2m + 1$ equations in the same number of unknowns. They can be solved for z_2 using a standard Newton-based shooting method, of the form (60), with

$$DF_{gl}^{\text{cont}} = \begin{bmatrix} DF_{gl} \\ \eta^T \end{bmatrix}. \quad (71)$$

The complete predictor–corrector process is illustrated in Fig. 16.

After the second point z_2 on the curve has been determined, an approximate tangent vector can generally be used for obtaining successive points. The approximate tangent vector at the i th point, which is used to calculate the $(i + 1)$ th point, is given by,

$$\eta_i = \frac{z_i - z_{i-1}}{\|z_i - z_{i-1}\|_2}. \quad (72)$$

Obtaining the approximate tangent vector involves much less computation than finding the exact tangent vector using (65), (66). However the

approximation may not be adequate in regions of high curvature.

6.2. Example 6 (Continuation)

Illustration of the continuation process again utilizes the SVC model in Sec. 2.2. The aim is to explore the relationship between values of L_r and C that give rise to grazing. In Sec. 5.2, an isolated grazing limit cycle was found by allowing the firing angle α to vary, while holding all other parameters constant. That case, corresponding to $\alpha = 102.16^\circ$, $L_r = 1.66$ mH and $C = 1.5$ mF, provides an initial point for the continuation process. Now though, the firing angle is fixed at $\alpha = 102.16^\circ$, and L_r and C are allowed to vary. The Euler homotopy algorithm provides a sequence of points along the resulting 1-manifold. Figure 17 shows the outcome of this continuation process.

Notice that the initial point is located quite close to a sharp turn in the curve. Even so, the continuation process proceeded reliably.

Grazing trajectories typically separate different forms of dynamic behavior. The curve of grazing points in Fig. 17 therefore separates regions where behaviors are structurally different. To illustrate, Fig. 18 shows limit cycles corresponding to parameter values $C = 1.51$ mF, and $L_r = 1.3$ mH, 1.4 mH and 1.55 mH. The parameter space locations of these three cases are identified in Fig. 17. Notice

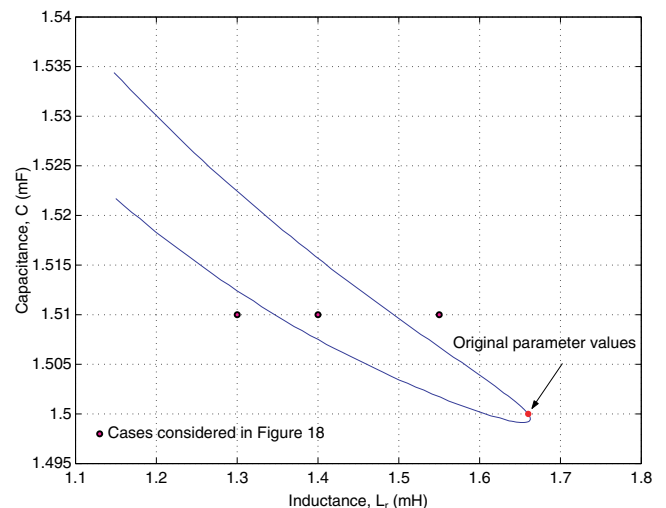


Fig. 17. Curve relating L_r and C , given by the continuation process.

¹²A numerically robust method of obtaining η follows from the factorization $DF_{gl}^T = QR$, where Q is orthogonal and R upper triangular. The last column of Q is exactly the desired η .

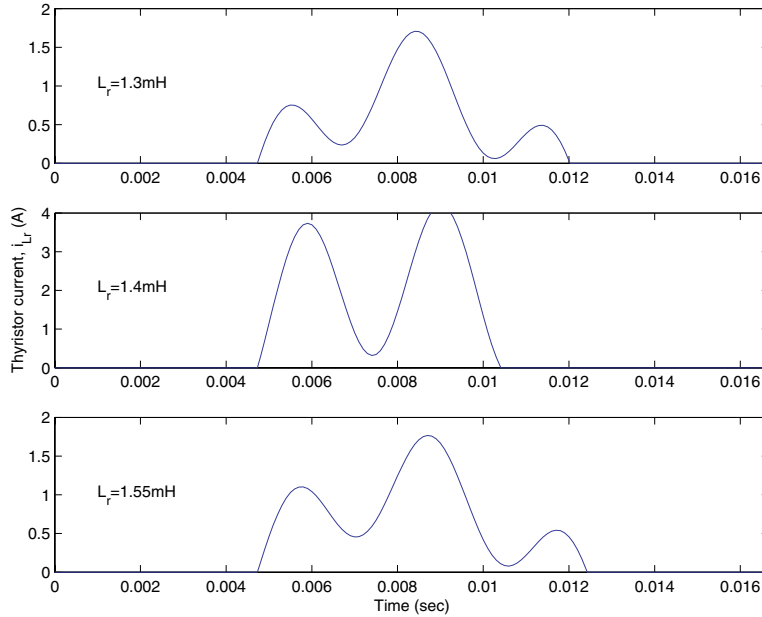


Fig. 18. Limit cycles for $C = 1.51$ mF and three values of L_r .

that the parameter set ($L_r = 1.4$ mH) from the region inside the curve gives rise to a limit cycle that differs greatly from the two cases that lie outside the curve.

7. Closest Grazing

It has been established previously that when $\ell \geq 2$, i.e. two or more parameters are free to vary, $F_{gl} : \mathbb{R}^{2n+2m+1} \rightarrow \mathbb{R}^{2n+2m+2-\ell}$ describes an $(\ell - 1)$ -manifold, or hypersurface. Let \mathcal{P} denote the projection of that manifold onto parameter space. As indicated in Fig. 18, limit cycles corresponding to parameters $p_0 \notin \mathcal{P}$ do not undergo grazing. However the parameter space “distance” between p_0 and \mathcal{P} provides an indication of the closeness, or robustness, of that limit cycle to grazing phenomena. A small distance would suggest that the system was vulnerable to grazing. Therefore identifying points on \mathcal{P} that are (locally) closest to a nominal parameter set p_0 provides important design information.

7.1. Mathematical formulation

In order to establish the closest grazing minimization problem, it is helpful to distinguish the parameters in (59). It follows from the partitioning (6), which specifies $x^T = [\underline{x}^T \ p^T]$, that (59) can be restated

$$F_{gl}(\tilde{z}, p) = 0 \tag{73}$$

where $\tilde{z} = [x_g^T \ y_g^T \ \underline{x}_0^T \ t_g \ v^T]^T \in \mathbb{R}^k$ and $k = 2n + 2m + 1 - \ell$. The minimization therefore takes the form

$$\min_p \frac{1}{2} \|p - p_0\|_A^2 \tag{74}$$

$$\text{s.t. } F_{gl}(\tilde{z}, p) = 0 \tag{75}$$

where $\|p - p_0\|_A^2$ denotes the A -norm $(p - p_0)^T A (p - p_0)$. For this to be a valid norm, A must be (symmetric) positive definite. It is common for A to be diagonal, with entries taking account of scaling differences between parameters.

The Lagrangian corresponding to (74)–(75) is

$$\mathcal{L}(\tilde{z}, p, \lambda) = \frac{1}{2} \|p - p_0\|_A^2 + \lambda^T F_{gl}(\tilde{z}, p), \tag{76}$$

and therefore optimal solutions are given by

$$\nabla \mathcal{L}(\tilde{z}, p, \lambda) = \begin{bmatrix} \frac{\partial F_{gl}^T}{\partial \tilde{z}} \lambda \\ A(p - p_0) + \frac{\partial F_{gl}^T}{\partial p} \lambda \\ F_{gl}(\tilde{z}, p) \end{bmatrix} = 0 \tag{77}$$

where $\partial F_{gl} / \partial \tilde{z}$ is the $(k + 1) \times k$ matrix constructed from DF_{gl} in (61) by removing the columns corresponding to parameters p . Those columns form the $(k + 1) \times \ell$ matrix $\partial F_{gl} / \partial p$. Well defined solutions of (77) require $\text{rank}\{\partial F_{gl} / \partial \tilde{z}\} = k$.

To understand the significance of (77), consider the hyperplane that is tangent to the manifold described by (73). Vectors $(d\tilde{z}, dp)$ on that tangent hyperplane satisfy

$$\frac{\partial F_{gl}}{\partial \tilde{z}} d\tilde{z} + \frac{\partial F_{gl}}{\partial p} dp = 0. \tag{78}$$

Premultiplying by the Lagrangian multiplier λ^T results in

$$\lambda^T \frac{\partial F_{gl}}{\partial \tilde{z}} d\tilde{z} + \lambda^T \frac{\partial F_{gl}}{\partial p} dp = 0. \tag{79}$$

But it follows from (77) that the coefficient of $d\tilde{z}$ in (79) equals zero. Therefore $\lambda^T (\partial F_{gl} / \partial p) dp = 0$, implying that the parameter space vector $(\partial F_{gl} / \partial p)^T \lambda \equiv N(p)$ is normal to \mathcal{P} at (\tilde{z}, p) .

It may be concluded that (77) describes points on the manifold (73) where the normal vector $N(p)$ aligns with $A(p - p_0)$.

7.2. Algorithm

An iterative procedure for finding the closest (in parameter space) grazing point can be established by utilizing the fact that vectors $A(p - p_0)$ and $N(p)$ align at that closest point. The following algorithm is adapted from [Dobson, 2003]. It converges locally if \mathcal{P} is convex (at least locally) or if it is only slightly concave.

Parameter values along the ray emanating from p_0 and in the direction of unit vector $\rho \in \mathbb{R}^\ell$ are given by $p = p_0 + \theta\rho$, where θ is a scalar. As parameters vary along that ray, the grazing hypersurface \mathcal{P} will be encountered at a point described by $F_{gl}(\tilde{z}, p_0 + \theta\rho) = 0$. This is a slightly modified version of (59), and can be solved using a shooting method similar to (60). The following iterative procedure seeks to minimize $\|p - p_0\|_A$.

1. Choose an initial direction ρ^1 . For example, this initialization could be achieved by freeing a single parameter and holding other parameters fixed.
2. Given ρ^i , compute the first grazing point in the direction ρ^i ; that is, compute θ^i such that $p^i = p_0 + \theta^i \rho^i \in \mathcal{P}$.
3. Compute the vector $N(p^i) = (\partial F_{gl} / \partial p)^T \lambda$ normal to \mathcal{P} at p^i . The Lagrangian multiplier vector λ is given by the first equation in (77), and can be obtained directly from the factorization $\partial F_{gl} / \partial \tilde{z} = QR$. The last column of Q is the desired λ . Computation of $N(p^i)$ is then straightforward.

4. Set $\rho^{i+1} = A^{-1}N(p^i)$ and normalize. Iterate until convergence of ρ^i to ρ^* . The (locally) closest grazing point occurs at $p^* = p_0 + \theta^* \rho^* \in \mathcal{P}$.

The parameter space direction ρ^* of a closest grazing point is aligned with the normal vector $N(p^*)$, and is the fixed point of the iterations. Note that in the special case of \mathcal{P} being a hyperplane, iterations converge in a single step.

7.3. Example 7 (Closest grazing point)

The SVC example again provides a useful illustration. The algorithm in Sec. 7.2 was used to find values for parameters $p = [L_r \ C]^T$ that give rise to grazing, and that are closest, in the A -norm sense, to the nominal values $p_0 = [1.2 \ 1.5]^T$. Notice from Fig. 17 that variations in L_r are about ten times greater than for C . Therefore, for the sake of the illustration, this scaling difference was compensated via the A -norm, by choosing $A = \begin{bmatrix} 1 & 0 \\ 0 & 10^2 \end{bmatrix}$.

The initial direction was chosen as $\rho^1 = [1 \ 0]^T$. Successive points generated by the algorithm are plotted in Fig. 19. (The figure is scaled to match the scaling introduced by A .) It can be seen that convergence was fast, even though a poor initial direction was specified. The vector $p^* - p_0$ is shown as a dashed line in the figure. In the scaled coordinate system, this vector is normal to the continuation curve \mathcal{P} at the point p^* .

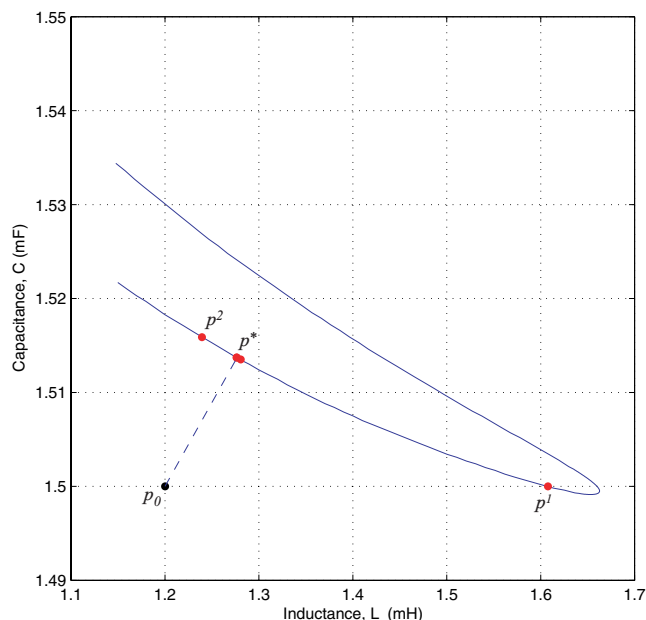


Fig. 19. Parameter space view of closest grazing iterations.

8. Conclusions

Hybrid (piecewise smooth) dynamical systems are characterized by intrinsic interactions between continuous dynamics and discrete events. Such systems are susceptible to grazing phenomena. Grazing refers to situations where the system trajectory tangentially (rather than transversally) encounters an event triggering hypersurface. Behavior beyond the grazing point is often not well defined.

Grazing points can be formulated as a boundary value problem, consisting of a set of nonlinear, algebraic equations that incorporates the system flow. Iterative solution via Newton's method requires numerical integration of the system trajectory, and therefore has the form of a shooting method. The Jacobian required by Newton's method incorporates trajectory sensitivities, which can be efficiently computed along with the trajectory. The shooting method is therefore practical for arbitrarily large hybrid systems.

As the number of parameters increases, the grazing point formulation gives rise to a continuum of solutions. A predictor-corrector continuation process has been developed to follow 1-manifolds, or curves, of grazing solutions. Furthermore, the grazing point formulation underlies the development of an algorithm for finding the smallest (in an A -norm sense) parameter change necessary to induce grazing. Such information is valuable in determining the robustness of a design (set of parameters) to grazing phenomena.

The paper has developed and illustrated shooting, continuation and optimization methods for both transient and periodic grazing. Examples are drawn from power electronics, power systems, and robotics.

References

- Brogliato, B. [1999] *Nonsmooth Mechanics: Models, Dynamics and Control*, 2nd edition (Springer-Verlag, NY).
- Chin, W., Ott, E., Nusse, H. E. & Grebogi, C. [1995] "Universal behavior of impact oscillators near grazing incidence," *Phys. Lett. A* **201**, 197–204.
- Dankowicz, H., Piiroinen, P. & Nordmark, A. B. [2002] "Low-velocity impacts of quasiperiodic oscillations," *Chaos Solit. Fract.* **14**, 241–255.
- David, R. & Alla, H. [1992] *Petri Nets and Grafecet* (Prentice-Hall, Englewood Cliffs, NJ).
- Dercole, F. & Kuznetsov, Y. A. [2005] "Slide-Cont: An Auto97 driver for bifurcation analysis of Filippov systems," *ACM Trans. Math. Softw.* **31**, 95–119.
- di Bernardo, M., Feigen, M. I., Hogan, S. J. & Homer, M. E. [1999] "Local analysis of C-bifurcations in n -dimensional piecewise-smooth dynamical systems," *Chaos Solit. Fract.* **10**, 1881–1908.
- di Bernardo, M., Budd, C. J. & Champneys, A. R. [2001] "Grazing and border-collision in piecewise-smooth systems: A unified analytical framework," *Phys. Rev. Lett.* **86**, 2553–2556.
- di Bernardo, M., Chung, H. S. & Tse, C. K. [2003] "Guest editorial: Special issue on switching and systems," *IEEE Trans. Circuits Syst.-I: Fund. Th. Appl.* **50**, 973–974.
- Dobson, I. [2003] "Distance to bifurcation in multidimensional parameter space: Margin sensitivity and closest bifurcations," in *Bifurcation Control, Theory and Application*, eds. Chen, G., Hill, D. J. & Yu X., Lecture Notes in Control and Information Sciences, Vol. 293 (Springer, NY).
- Doedel, E., Champneys, A., Fairgrieve, T., Kuznetsov, Y., Sandstede, B. & Wang, X.-J. [1998] "Auto97: Continuation and bifurcation software for ordinary differential equations," Technical report, Department of Computer Science, Concordia University, Montreal, Canada. Available from <http://cmvl.cs.concordia.ca/auto/>.
- Feehery, W. F., Tolsma, J. E. & Barton, P. I. [1997] "Efficient sensitivity analysis of large-scale differential-algebraic systems," *Appl. Numer. Math.* **25**, 41–54.
- Fleming, W. [1977] *Functions of Several Variables* (Springer-Verlag, NY).
- Frank, P. M. [1978] *Introduction to System Sensitivity Theory* (Academic Press, NY).
- Fredriksson, M. H. & Nordmark, A. B. [1997] "Bifurcations caused by grazing incidence in many degrees of freedom impact oscillators," *Proc. Roy. Soc. London A* **453**, 1261–1276.
- Galán, S. & Barton, P. I. [1998] "Dynamic optimization of hybrid systems," *Comput. Chem. Eng.* **22**, S183–S190.
- Garcia, C. B. & Zangwill, W. I. [1981] *Pathways to Solutions, Fixed Points and Equilibria* (Prentice Hall, Englewood Cliffs, NJ).
- Goodwin, G., Graebe, S. & Salgado, M. [2001] *Control System Design* (Prentice Hall, Upper Saddle River, NJ).
- Goswami, A., Thuilot, B. & Espiau, B. [1998] "A study of the passive gait of a compass-like biped robot: Symmetry and chaos," *Int. J. Robot. Res.* **17**, 1282–1301.
- Hiskens, I. A. & Pai, M. A. [2000] "Trajectory sensitivity analysis of hybrid systems," *IEEE Trans. Circuits Syst.-I: Fund. Th. Appl.* **47**, 204–220.
- Hiskens, I. A. & Sokolowski, P. J. [2001] "Systematic modeling and symbolically assisted simulation of

- power systems,” *IEEE Trans. Power Syst.* **16**, 229–234.
- Hiskens, I. A. [2004] “Power system modeling for inverse problems,” *IEEE Trans. Circuits Syst.-I* **51**, 539–551.
- Jalali, S., Dobson, I., Lasseter, R. H. & Venkataramanan, G. [1996] “Switching time bifurcations in a thyristor controlled reactor,” *IEEE Trans. Circuits Syst.-I* **43**, 209–218.
- Lennartson, B., Tittus, M., Egardt, B. & Pettersson, S. [1996] “Hybrid systems in process control,” *Contr. Syst. Mag.* **16**, 45–56.
- Li, S., Petzold, L. & Zhu, W. [2000] “Sensitivity analysis of differential-algebraic equations: A comparison of methods on a special problem,” *Appl. Numer. Math.* **32**, 161–174.
- Liberzon, D. [2003] *Switching in Systems and Control* (Birkhauser, Boston).
- Nusse, H. E., Ott, E. & Yorke, J. A. [1994] “Border-collision bifurcations: An explanation for observed bifurcation phenomena,” *Phys. Rev. E* **49**, 1073–1077.
- Parker, T. S. & Chua, L. O. [1989] *Practical Numerical Algorithms for Chaotic Systems* (Springer-Verlag, NY).
- Piccoli, B. [1998] “Hybrid systems and optimal control,” in *Proc. 37th IEEE Conf. Decision and Control*, Tampa, FL.
- Piiroinen, P. [2002] “Recurrent dynamics of nonsmooth systems with application to human gait,” Doctoral Thesis, Department of Mechanics, Royal Institute of Technology, Stockholm.
- Rajaraman, R., Dobson, I. & Jalali, S. [1996] “Nonlinear dynamics and switching time bifurcations of a thyristor controlled reactor circuit,” *IEEE Trans. Circuits Syst.-I* **43**, 1001–1006.
- Sauer, P. W. & Pai, M. A. [1998] *Power System Dynamics and Stability* (Prentice Hall, Upper Saddle River, NJ).
- Seydel, R. [1994] *Practical Bifurcation and Stability Analysis*, 2nd edition (Springer-Verlag, NY).
- Shampine, L. [1994] *Numerical Solution of Ordinary Differential Equations* (Chapman and Hall, NY).
- Singh, C. & Hiskens, I. A. [2001] “Direct assessment of protection operation and non-viable transients,” *IEEE Trans. Power Syst.* **16**, 427–434.
- Spong, M. & Vidyasagar, M. [1989] *Robot Dynamics and Control* (Wiley, NY).
- Stoer, J. & Bulirsch, R. [1993] *Introduction to Numerical Analysis*, 2nd edition (Springer, NY).
- U.S.-Canada Power System Outage Task Force [2004] *Final Report on the August 14, 2003 Blackout in the United States and Canada: Causes and Recommendations*.
- van der Schaft, A. & Schumacher, H. [2000] *An Introduction to Hybrid Dynamical Systems* (Springer-Verlag, London).
- Yuan, G., Banerjee, S., Ott, E. & Yorke, J. A. [1998] “Border-collision bifurcations in the buck converter,” *IEEE Trans. Circuits Syst.-I* **45**, 707–715.

Appendix A Variational Equations

Away from events, where system dynamics evolve smoothly, the sensitivities Φ and Ψ are obtained by differentiating (7)–(8) with respect to x_0 .¹³ This gives

$$\dot{\Phi} = f_x(t)\Phi + f_y(t)\Psi \quad (\text{A.1})$$

$$0 = g_x(t)\Phi + g_y(t)\Psi \quad (\text{A.2})$$

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 [Feehery *et al.*, 1997; Hiskens & Pai, 2000; Li *et al.*, 2000] 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 (7)–(8).

Initial conditions for Φ are obtained from (11) as

$$\Phi(x_0, t_0) = I$$

where I is the identity matrix. Initial conditions for Ψ follow directly from (A.2),

$$0 = g_x(t_0) + g_y(t_0)\Psi(x_0, t_0).$$

Equations (A.1)–(A.2) describe the evolution of the sensitivities Φ and Ψ between events. However at an event, the sensitivities are often discontinuous. It is necessary to calculate *jump conditions* describing the step change in Φ and Ψ . For clarity, consider a single switching/reset event, so the model (1)–(4) effectively reduces to the form

$$\dot{x} = f(x, y) \quad (\text{A.3})$$

$$0 = \begin{cases} g^-(x, y), & s(x, y) < 0 \\ g^+(x, y), & s(x, y) > 0 \end{cases} \quad (\text{A.4})$$

$$x^+ = h(x^-, y^-), \quad s(x, y) = 0. \quad (\text{A.5})$$

(The switching hypersurface s has been given an explicit form so as to elucidate its role in the jump conditions.)

¹³Keep in mind that parameters p are incorporated into x_0 .

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

$$\begin{aligned} x^- &\equiv x(\tau^-) = \phi(x_0, \tau^-) \\ y^- &\equiv y(\tau^-) = \psi(x_0, \tau^-) \end{aligned}$$

where

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

Similarly, x^+, y^+ are defined for time τ^+ , just after the event has occurred. Hiskens and Pai [2000] show that the jump conditions for the sensitivities Φ are given by

$$\Phi(x_0, \tau^+) = h_x^* \Phi(x_0, \tau^-) - (f^+ - h_x^* f^-) \tau_{x_0} \tag{A.6}$$

where

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

The sensitivities Ψ immediately after the event are given by

$$\Psi(x_0, \tau^+) = -(g_y^+(\tau^+))^{-1} g_x^+(\tau^+) \Phi(x_0, \tau^+).$$

Following the event i.e. for $t > \tau^+$, calculation of the sensitivities proceeds according to (A.1)–(A.2), until the next event is encountered. The jump conditions provide the initial conditions for post-event calculations.