Grazing bifurcations occur when a small parameter variation induces a change in the event sequence of a hybrid system, i.e., a system where continuous dynamics and discrete events strongly interact. At such a bifurcation, the system trajectory makes tangential contact with (grazes) an event triggering hypersurface. This bounding case separates regions of (generally) quite different dynamic behaviour. The paper formulates the conditions governing grazing bifurcation points, and extends those conditions to limit cycles. A shooting method is used to solve for bifurcating limit cycles. The approach is applicable for general nonlinear hybrid systems.

Grazing bifurcations [1, 2, 3] refer to situations where a small parameter variation in-duction in event sequence of hybrid systems can be captured 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,

\[ \dot{x} = f(x, y) \]
\[ x^+ = h_j(x^-, y^-) \quad \text{when } y_{r,j} = 0 \]
\[ 0 = g(x, y) \equiv g^{(0)}(x, y) + \sum_{i=1}^{s} g^{(i)}(x, y) \]

where

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

and

- \( x \in \mathbb{R}^n \) are dynamic states, and \( y \in \mathbb{R}^m \) are algebraic states;
- \( x^+ \) 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.
- \( y_{r,i} \) and \( y_{s,i} \) are selected elements of \( y \) that trigger state reset (impulsive) and algebraic switching events respectively; \( y_r \) and \( y_s \) may share common elements.

Research supported by the National Science Foundation through grant ECS-0332777.
A compact development of the equations describing grazing bifurcations results from incorporating system parameters $\lambda \in \mathbb{R}^4$ into the dynamic states $x$. This is achieved by introducing trivial differential equations

$$\dot{\lambda} = 0$$

into (1), and results in the natural partitioning

$$x = \begin{bmatrix} \bar{x} \\ \lambda \end{bmatrix}, \quad f = \begin{bmatrix} f \\ 0 \end{bmatrix}, \quad h_j = \begin{bmatrix} b_j \\ \lambda \end{bmatrix}$$

where $\bar{x}$ are the true dynamic states and $\lambda$ are parameters.

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

$$\begin{align*}
\dot{x} &= f(x, y) \\
0 &= g(x, y)
\end{align*}$$

where $g$ is composed of $g^{(0)}$ together with appropriate choices of $g^{(1\pm)}$ or $g^{(2\pm)}$, depending on the signs of the corresponding elements of $y$. At switching events (4), some component equations of $g$ change. To satisfy the new $g = 0$, constraints, algebraic variables $y$ may undergo a step change. Impulse events (2) force a discrete change in elements of $x$. Algebraic variables may again step to ensure $g = 0$ is always satisfied.

The flows of $x$ and $y$ are defined as

$$\begin{align*}
x(t) &= \phi_1(x_0, t) \\
y(t) &= \phi_2(x_0, t)
\end{align*}$$

where $x(t)$ and $y(t)$ satisfy (1)-(4), along with initial conditions,

$$\phi_1(x_0, t_0) = x_0 \quad g(x_0, \phi_2(x_0, t_0)) = 0.$$ (11)

2.2. Grazing bifurcations

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

$$b(x, y) = 0$$

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

$$\begin{bmatrix} b_x \\ b_y \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} = 0.$$ (13)

The vector $[x^T \ y^T]^T$ is directed tangentially along the flow, so it must satisfy (13) at a border collision bifurcation. Furthermore, differentiating (8) and substituting (7) gives,

$$\begin{align*}
0 &= \frac{\partial g}{\partial x} \dot{x} + \frac{\partial g}{\partial y} \dot{y} \\
\Rightarrow 0 &= g_x f(x, y) + g_y v
\end{align*}$$

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

A single degree of freedom is available for varying parameters to find a bifurcation value. It follows from (6) that system parameters $\lambda$ are incorporated into the initial conditions $x_0$. Therefore the single degree of freedom can be achieved by parameterization $x_0(\theta)$, where $\theta$ is a scalar.

Grazing bifurcation points are therefore described by combining together the flow definition (9) (appropriately parameterized by $\theta$), algebraic equations (8), target hypersurface (12), and tangency conditions (13),(15), to give

$$\begin{align*}
F_1(x_0, \theta, t_0) &= \phi_1(x_0, t_0) - x_0 = 0 \\
F_2(x_0, y_0) &= g(x_0, y_0) = 0 \\
F_3(x_0, y_0) &= b(x_0, y_0) = 0 \\
F_4(x_0, y_0, v) &= \begin{bmatrix} b_x & b_y \\ g_x & g_y \end{bmatrix} \begin{bmatrix} f(x_0, y_0) \\ v \end{bmatrix} = 0.
\end{align*}$$ (19)

This set of equations may be written compactly as

$$F_g(x_0, y_0, \theta, t_0, v) = 0$$

where $F_g : \mathbb{R}^{n+m+2} \rightarrow \mathbb{R}^{n+m+2}$. Solution of (20) can be achieved using Newton’s method [5]. The solution process involves numerical simulation to obtain the flow (16), and is therefore classed as a shooting method.

2.3. Limit cycles

Periodic behaviour of limit cycles implies that the system state returns to its initial value every cycle.\(^1\) This can be expressed in terms of the flow as $x^* = \phi_1(x^*, T)$ where $T$ is the limit cycle period. 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 $\tau_r$, can be found using Poincaré map concepts [8]. It is a function of the limit cycle starting point, so autonomous limit cycles can be expressed as

$$x^* = \phi_1(x^*, \tau_r(x^*)).$$ (21)

Computationally this case is more interesting, so is the focus of the paper.

For a limit cycle to undergo a grazing bifurcation, (16)-(19) must be satisfied along with (21). These extra conditions can generally only be met by increasing the degrees of freedom beyond the single parameter $\theta$ of (16). In fact it becomes necessary to allow all initial conditions $x_0$ to vary. Accordingly, (16) becomes

$$F'_1(x_0, x_0, t_0) := \phi_1(x_0, t_0) - x_0 = 0$$

and (21) can be expressed as

$$F_5(x_0) := \phi_1(x_0, \tau_r(x_0)) - x_0 = 0.$$ (23)

Combining together (17)-(19), (22) and (23) gives

$$F(z) = F(x_0, y_0, x_0, t_0, v) = \begin{bmatrix} F'_1(x_0, x_0, t_0) \\ F_2(x_0, y_0) \\ F_3(x_0, y_0) \\ F_4(x_0, y_0, v) \\ F_5(x_0) \end{bmatrix} = 0$$

where $z = [x_0^T \ y_0^T \ x_0^T \ t_0 \ v]^T$ and $F : \mathbb{R}^{n+m+2} \rightarrow \mathbb{R}^{n+m+2}$.

Notice though that $F$ defines an over-determined set of equations, with one more equation than variable. This can be resolved by allowing one free parameter $\lambda$. To see this, recall that if system

\(^1\)Limit cycles also require that the periodic trajectory is isolated in state-space.
parameters \( \lambda \in \mathbb{R}^\ell \) are incorporated into \( x \) as in (6), then (5) ensures that for the \( \lambda \) component of \( x \), \( \phi_1(x_0, t) = x_0 \) always. The associated equations in (23) are redundant and can be eliminated with no loss of information. It follows that in order to achieve an exactly determined problem, one equation should be eliminated, implying that \( \lambda \) should be a scalar. If \( \dim(\lambda) = \ell > 1 \), the problem becomes under-determined, resulting in a solution manifold of dimension \( \ell - 1 \), rather than a point solution. In the remainder, it shall be assumed that \( \ell = 1 \), and that scalar \( \lambda \) is the \( n \)-th element of \( x \). Accordingly, the last equation in \( F_5 \) is deleted from \( F \) in (24).

3. SHOOTING METHOD

3.1. Algorithm

Numerical solution of (24) using Newton’s method amounts to iterating on the standard update formula

\[
z^{k+1} = z^k - (DF(z^k))^{-1} F(z^k)
\]

(25)

where \( DF \) is the Jacobian matrix

\[
DF = \begin{bmatrix}
-I_n & 0 & \Phi_1 \\
 g_x & g_y & b_y \\
 f'^2 b_{xx} + b_x f_x + v' b_{yx} & f' b_{xy} + b_y f_y + v' b_{yy} & 0 \\
 f'^2 g_{xx} + g_x f_x + v' g_{yx} & f' g_{xy} + g_y f_y + v' g_{yy} & 0 \\
 0 & 0 & 0
\end{bmatrix}
\]

(26)

where \( I_n \) the \((n \times n)\) identity matrix, and

\[
\hat{f} = \begin{bmatrix}
 f \\
 f \\
 \vdots \\
 f \\
 v \\
 v \\
 \vdots \\
 v
\end{bmatrix} \in \mathbb{R}^{m_n \times m}
\]

\[
\hat{v} = \begin{bmatrix}
 v \\
 v \\
 \vdots \\
 v
\end{bmatrix} \in \mathbb{R}^{m_v \times m}
\]

\[
g_{xx} = \begin{bmatrix}
\frac{\partial^2 f_1}{\partial x^2} \\
\frac{\partial^2 f_2}{\partial x^2} \\
\vdots \\
\frac{\partial^2 f_m}{\partial x^2}
\end{bmatrix} \in \mathbb{R}^{m_n \times n}, \quad
\hat{g}_{xx} = \begin{bmatrix}
\frac{\partial^2 g_1}{\partial y^2} \\
\frac{\partial^2 g_2}{\partial y^2} \\
\vdots \\
\frac{\partial^2 g_m}{\partial y^2}
\end{bmatrix} \in \mathbb{R}^{m_n \times n}
\]

\[
g_{yy} = \begin{bmatrix}
\frac{\partial^2 f_1}{\partial y^2} \\
\frac{\partial^2 f_2}{\partial y^2} \\
\vdots \\
\frac{\partial^2 f_m}{\partial y^2}
\end{bmatrix} \in \mathbb{R}^{m_m \times m}, \quad
\hat{g}_{yy} = \begin{bmatrix}
\frac{\partial^2 g_1}{\partial y^2} \\
\frac{\partial^2 g_2}{\partial y^2} \\
\vdots \\
\frac{\partial^2 g_m}{\partial y^2}
\end{bmatrix} \in \mathbb{R}^{m_m \times m}
\]

The entry \( \Phi_1 \) in (26) gives the sensitivity of the flow (9) to perturbations in initial conditions \( x_0 \),

\[
\Phi_1(x_0, t) \equiv \frac{\partial \phi_1}{\partial x_0}(x_0, t).
\]

The variational equations describing the evolution of trajectory sensitivities \( \Phi_1, \Phi_2 \) are given in [7]. Note that these quantities are defined for non-smooth trajectories generated by hybrid systems. Furthermore, if an implicit numerical integration technique is used, these sensitivities incur minimal additional computational cost.

The submatrix \( DF_{5_{-}} \) refers to the Jacobian of \( F_5 \) in (23), with the last row removed. It is shown in [8] that \( DF_5 \) is given by

\[
DF_5(x) = \left( I - \frac{f|_{\tau_r(x)}g^T}{\sigma^2 f|_{\tau_r(x)}} \right) \Phi_1(x, \tau_r(x)) - I
\]

(27)

where \( \sigma \) is a unit vector normal to the hyperplane \( \Sigma \) defining the Poincaré map.

Care must be taken in evaluating the terms of (24) and (26) that relate to trajectory solution. The flow term \( \phi_1(x_0^k, t_0^k) \) of (22) evaluates, via numerical integration, to the value of \( x \) at time \( t_0^k \) along the trajectory that has initial value \( x_0^k \). The associated Jacobian entries \( \Phi_1 \) and \( f \) in the first row of \( DF \) should also be evaluated at time \( t_0^k \) along that trajectory. Similarly, \( F_5 \) in (23) involves the flow term \( \phi_1(x_0^k, \tau_r(x_0^k)) \). This evaluates to \( x \)

at the point where the trajectory, starting from \( x_0^k \in \Sigma \), re-encounters \( \Sigma \). The associated Jacobian entry \( DF_{5_{-}}(x_0^k) \) is also evaluated at the return point, see (27).

The matrices \( g_{xx}, g_{yx}, g_{xy}, g_{yy} \) 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 they do affect convergence. Efficient computation of these matrices is discussed in [5].

4. EXAMPLE

A model of the compass gait biped robot is discussed in detail in [9], with a summary given in [10]. The biped robot can be treated as a double pendulum. Figure 2 provides a schematic representation and identifies important parameters, including the incline angle \( \gamma \). The robot configuration is described by the support angle \( \theta_s \) and the non-support angle \( \theta_{ns} \).

Dynamic equations describe the evolution of the state vector \( x = [\theta_{ns}, \theta_s, \dot{\theta}_{ns}, \dot{\theta}_s, \dot{\gamma}]^T \in \mathbb{R}^5 \) during the swing phase. An event occurs when the non-support (swinging) leg collides with the ground. This establishes the triggering condition \( \theta_{ns} + \theta_s + 2\gamma = 0 \). The biped robot is therefore an example of a hybrid system, with walking motion corresponding to a periodic orbit.

To illustrate the shooting algorithm, a target hypersurface was established by \( b(x, y) = x_3 - 2.5 = 0 \) where \( x_3 \equiv \theta_{ns} \). This could be interpreted as a maximum allowable non-support leg velocity. (Perhaps the robot falls apart at higher velocities!) Walking motion that just satisfied this constraint was achieved by varying the incline angle \( \gamma \). Solution progress is given in Table 4, and illustrated in Fig. 3. The figure shows the initial trajectory, and final grazing limit cycle.

\(^{2}\)This occurs at return time \( \tau_r(x_0^k) \), though that time is not explicitly determined.
The Poincaré hyperplane $\Sigma = \{ x : x_3 = 0.1 \}$, with normal vector $\sigma = [0\ 0\ 1\ 0]^T$. A projection of this hyperplane is shown in Fig. 3. The initial trajectory started 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 these eigenvalues lies outside the unit circle, this limit cycle is non-stable. Even so, shooting method convergence was fast and reliable. On the other hand, locating this limit cycle by repeated simulations would be extremely difficult.

It can be seen from Figure 3 that this example exhibits non-smooth (hybrid) behaviour. In this case, only one event (impact) occurs per cycle. However the shooting method is more widely applicable, to systems that undergo any (finite) number of events per cycle.

5. CONCLUSIONS

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

Grazing bifurcation points of limit cycles 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.

6. REFERENCES


