Demonstration of a New Software for Stability and Bifurcation Analysis of Switching Dynamical Systems

Kuntal Mandal

Department of Electrical and Electronics Engineering
National Institute of Technology Sikkim
Ravangla – 737139, India

E-mail: dr.kuntal.mandal@gmail.com

Outline

- Introduction
- Different Class of Switching Dynamical Systems
- Saltation Matrix for different cases
- Automated Algorithm and Implementation
- Examples
- Demonstration
- Conclusions

Switching Dynamical Systems

Mathematically, an n-dimensional switching dynamical system can be described by piecewise linear (or nonlinear) set of differential equations of the form

$$\dot{x} = f(x, t, \rho) = \begin{cases} f_1(x, t, \rho), & \text{for } x \in M_1 \\ f_2(x, t, \rho), & \text{for } x \in M_2 \\ \ldots\ldots \\ \ldots\ldots \\ f_m(x, t, \rho), & \text{for } x \in M_m \end{cases}$$

where M_1, $M_2, \cdots M_m$ are different regions of state-space separated by $(n-1)$ dimensional switching surfaces given by algebraic equation of the form $h(x, t) = 0$ and ρ is a system parameter.

Different Class of Switching Dynamical Systems

(a) Systems with discontinuous vector fields
[Ex: Relay systems, Power Electronic Systems, Dry friction]

(b) Systems with discontinuous state jump
[Ex: Impact Oscillators, Converter with reset switch, overhead camshaft automotive valve train]

(c) Systems with Piecewise Nonlinear Subsystems
[Ex: Alpazur Oscillators, DC Series Motor]

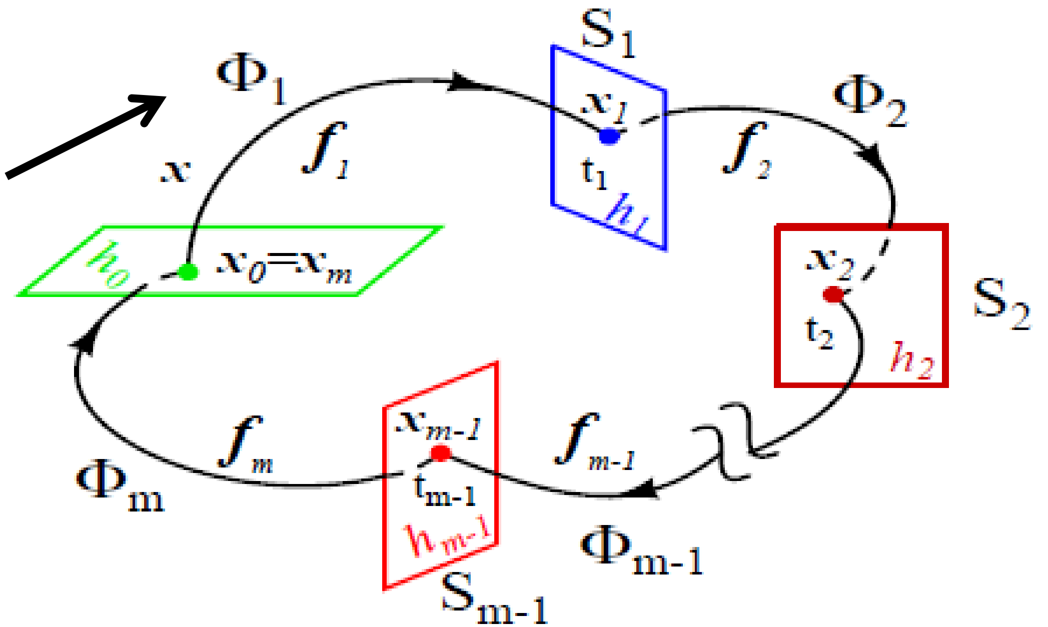
(d) Systems with Differential algebraic equations
[Ex: PV connected Converters, Robotic Systems, Power Systems]

Filippov Method: Monodromy Matrix

$$M_i : \frac{d\boldsymbol{x}}{dt} = \boldsymbol{f}_i(\boldsymbol{x}, t) = \boldsymbol{A}_i \boldsymbol{x} + \boldsymbol{B}_i \boldsymbol{u}$$

$$h_i(\boldsymbol{x}, t) = 0$$

$$\Phi_i = \frac{\partial \boldsymbol{x}_i}{\partial \boldsymbol{x}_{i-1}} = e^{\boldsymbol{A}_i(t_i - t_{i-1})}$$



$$\boldsymbol{M}(T_s, t_0, \boldsymbol{x}_0) = \frac{\partial \boldsymbol{x}_m}{\partial \boldsymbol{x}_{m-1}} \cdot \boldsymbol{S}_{m-1} \cdot \frac{\partial \boldsymbol{x}_{m-1}}{\partial \boldsymbol{x}_{m-2}} \cdots \boldsymbol{S}_2 \cdot \frac{\partial \boldsymbol{x}_2}{\partial \boldsymbol{x}_1} \cdot \boldsymbol{S}_1 \cdot \frac{\partial \boldsymbol{x}_1}{\partial \boldsymbol{x}_0}$$

$$= \Phi_m \cdot \boldsymbol{S}_{m-1} \cdot \Phi_{m-1} \cdots \boldsymbol{S}_2 \cdot \Phi_2 \cdot \boldsymbol{S}_1 \cdot \Phi_1$$

STM for Linear Subsystems

Integrate the differential equations along with the checking of switching condition.

A proper adaptive (variable) step-size integration gave us satisfying performance along with precious control of the numerical error to find out t_1.

$$\Phi_1 = exp(\boldsymbol{A_1}(t_1 - t_0))$$

Matlab's *expm function* which is ten-term Taylor's series approximation

Ref: Nick Higham, How and How Not to Compute the Exponential of a Matrix, The University of Manchester.

STM for Nonlinear Subsystems

We have to solve the system equations and variational equations simultaneously.

Differential equations for Subsystem M_1:

$$\dot{x} = f_1(x, t, \rho)$$
$$x(t = 0) = x_0$$

Variational equations for Subsystem M_1:

$$\dot{\Phi}_1 = \frac{\partial f_1(x, t, \rho)}{\partial x} \Phi_1$$
$$\Phi_1(t = 0) = I_n$$

For n-dimensional system, $(n + n^{2})$ differential equations have to be solved.

Ref: The Variational Equation, Liz Bradley, Department of Computer Science, University of Colorado.

Van der Pole System

2nd Order ODE $\ddot{x} - \mu(1 - x^2)\dot{x} + x = 0,$

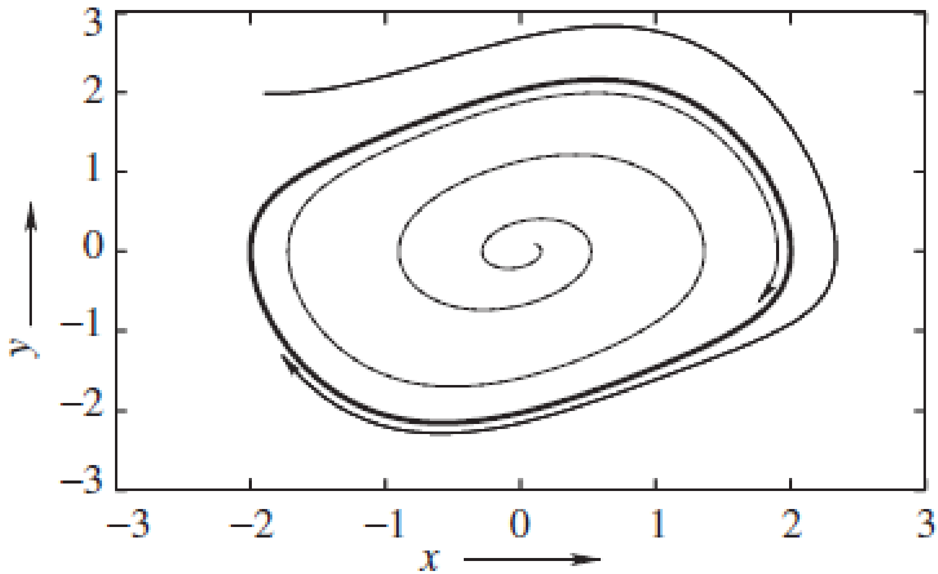
1st Order ODEs $\dot{x} = y,$

$\dot{y} = \mu(1 - x^2)y - x.$

f

$$\begin{bmatrix} \dot{x} \\ \dot{y} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -2\mu x y - 1 & \mu - \mu x^2 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

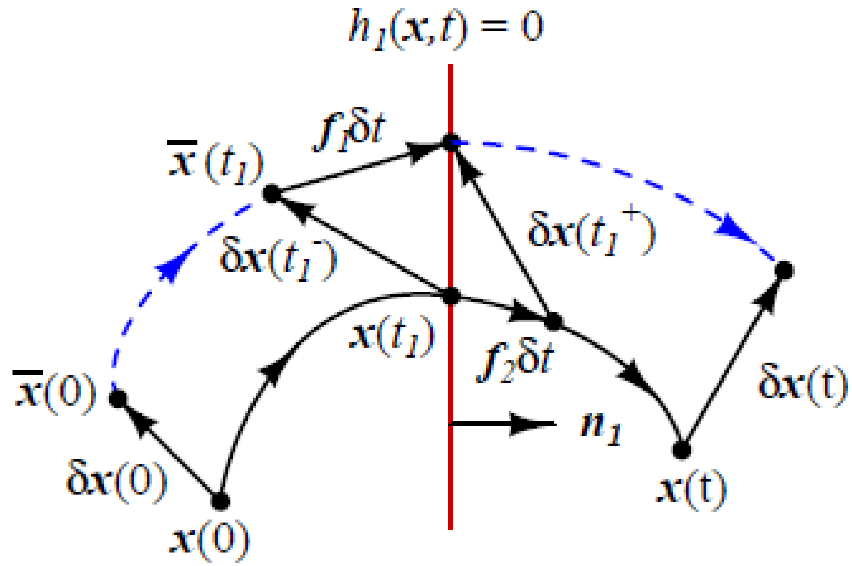
J



Saltation Matrix (Transversal Intersection)

The saltation matrix relates how the perturbation **before** the switching maps to the perturbation **after** the switching.

$$\delta x(t_1^+) = S_1 \delta x(t_1^-)$$

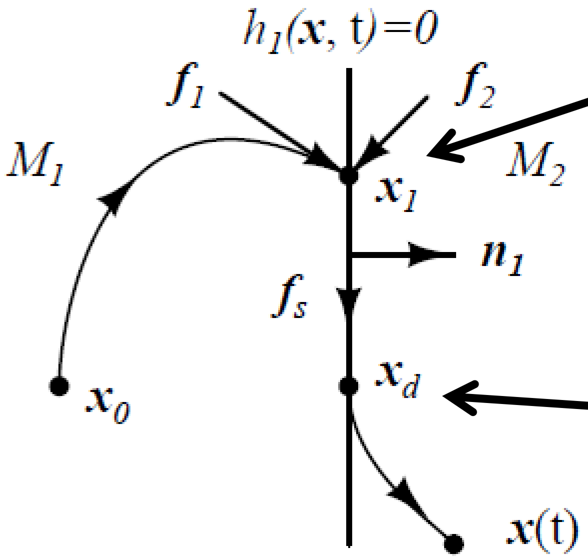


$$S_1 = \Phi(t_1^+, t_1^-) = I_n + \frac{[f_2(x_1, t_1) - f_1(x_1, t_1)]\, n_1^\top}{n_1^\top f_1(x_1, t_1) + \left.\frac{\partial h_1}{\partial t}\right|_{t=t_1}}$$

Ref: Books by R.I. Leine and H. Nijmeijer (2004) , A. F. Filippov (1988) , di Bernardo et al. 2008

Saltation Matrix (Crossing Sliding)

To find out the sliding vector field $\boldsymbol{f}_s$ we use Filippov's convex method (or Utkin's equivalent control).

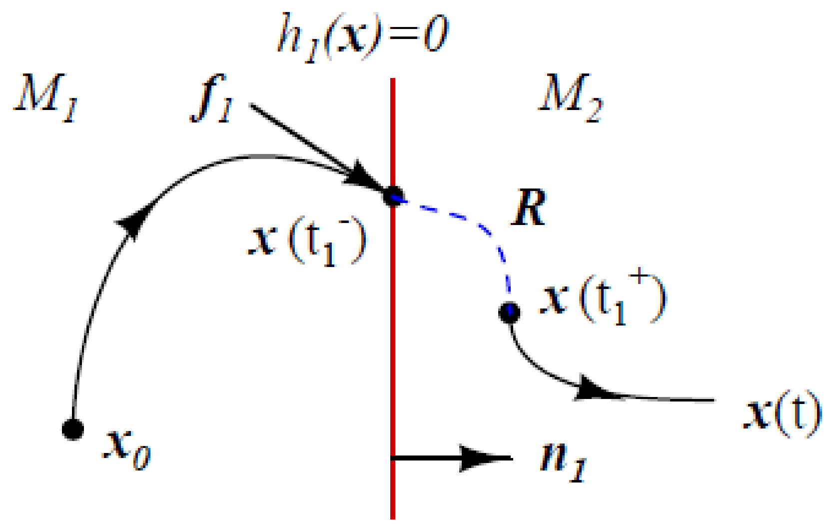


$$\boldsymbol{S}_1 = \Phi(t_1^+, t_1^-) = \boldsymbol{I}_n + \frac{[\boldsymbol{f}_s(\boldsymbol{x}_1, t_1) - \boldsymbol{f}_1(\boldsymbol{x}_1, t_1)]\, \boldsymbol{n}_1^\top}{\boldsymbol{n}_1^\top \boldsymbol{f}_1(\boldsymbol{x}_1, t_1)}$$

$$\boldsymbol{S}_d = \Phi(t_d^+, t_d^-) = \boldsymbol{I}_n$$

Ref: Books by R.I. Leine and H. Nijmeijer (2004) , A. F. Filippov (1988) , di Bernardo et al. 2008

Saltation Matrix (Impacting System)



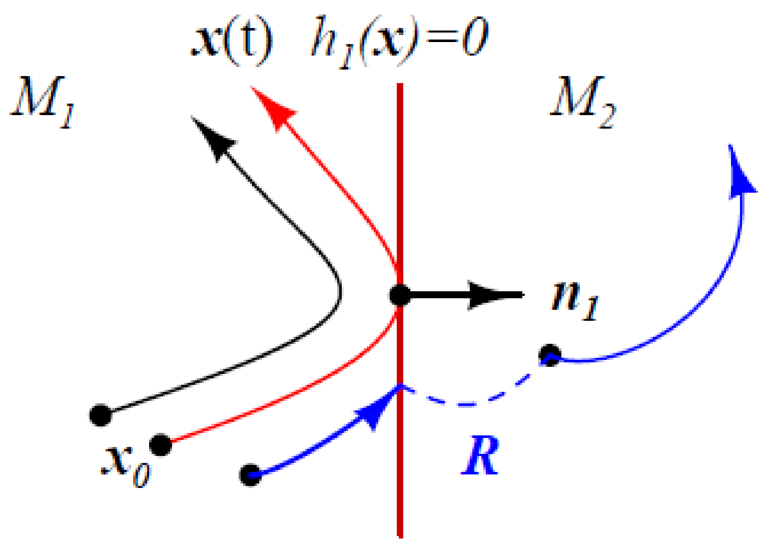
Impact law: $\boldsymbol{x}(t_1^+) = \boldsymbol{R}\boldsymbol{x}(t_1^-)$

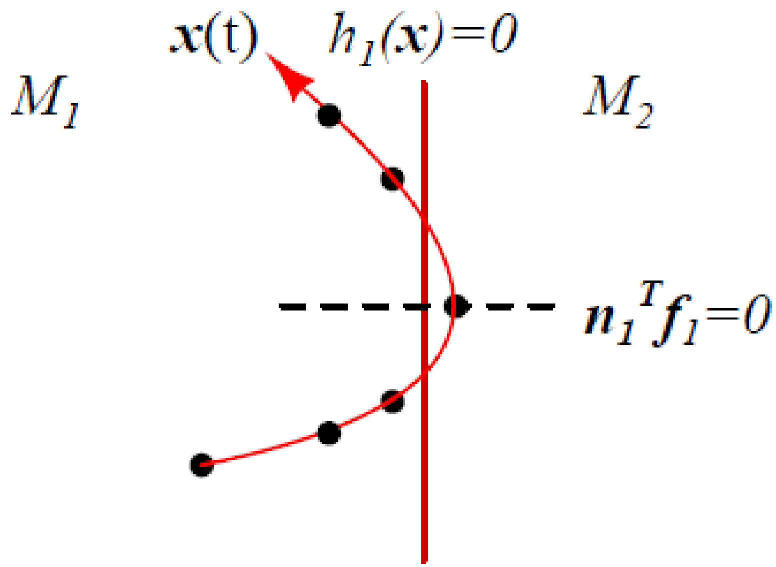
$$\boldsymbol{S}_1 = \Phi(t_1^+, t_1^-) = \boldsymbol{R} + \frac{[\boldsymbol{f}_2(\boldsymbol{x}_1, t_1) - \boldsymbol{R}\boldsymbol{f}_1(\boldsymbol{x}_1, t_1)]\,\boldsymbol{n}_1^\top}{\boldsymbol{n}_1^\top \boldsymbol{f}_1(\boldsymbol{x}_1, t_1)}$$

For piecewise-smooth systems, $\boldsymbol{R} = \boldsymbol{I}_n$.

Ref: P. T. Piiroinen et al., Chaos and Period-Adding in an Impacting System, Journal of Nonlinear Science, 2004.

Detection of Grazing





Ref: P. T. Piiroinen et al., Chaos and Period-Adding in an Impacting System, Journal of Nonlinear Science, 2004.

Locating the Periodic Orbit and Finding Its Stability

For a periodic orbit, $\boldsymbol{H}(\boldsymbol{x}_0) = \boldsymbol{x}_m - \boldsymbol{x}_0 = 0.$

Newton-Raphson method:

$$\begin{aligned}
\boldsymbol{x}_0^{k+1} &= \boldsymbol{x}_0^k - \left[\frac{\partial \boldsymbol{H}(\boldsymbol{x}_0^k)}{\partial \boldsymbol{x}_0^k}\right]^{-1} \boldsymbol{H}(\boldsymbol{x}_0^k) \\
&= \boldsymbol{x}_0^k - \left[\frac{\partial \boldsymbol{x}_m^k}{\partial \boldsymbol{x}_0^k} - \boldsymbol{I}_n\right]^{-1} (\boldsymbol{x}_m^k - \boldsymbol{x}_0^k).
\end{aligned}$$

The Jacobian matrix:

$$\boldsymbol{J} = \frac{\partial \boldsymbol{x}_m}{\partial \boldsymbol{x}_0} = \boldsymbol{M}(T_s, t_0, \boldsymbol{x}_0)$$

Our Algorithm

- Start from a suitable initial condition.
- Simulate for one period. Determine the subsystem sequence in that period, and the times spent in each subsystem.
- Determine the state transition matrices for each subsystem, and the saltation matrices for each switching. Obtain the monodromy matrix.

The Algorithm (continued)

- Take one Newton-Raphson step using this Jacobian matrix. Obtain a new initial condition. Repeat the steps.
- When the algorithm converges on the fixed point, the Jacobian converges on that of the periodic orbit.
- Obtain the Eigenvalues.
- Once a periodic orbit is located, make a small change in the parameter and use the old periodic orbit as the initial guess (**natural continuation method**).

The Algorithm (continued)

- It is not restricted by the problem size (dimension, number of subsystems, switching conditions, etc.).
- Convergence may be a problem unless the initial condition is close to the periodic orbit.

We overcome the problem by starting from a stable periodic orbit, which can be located using **brute force simulation.**

Advantages of the Software

- No limitation in problem size i.e., dimensions and number of subsystems. Higher number of subsystems in a period only increases the number of matrices to be multiplied, and not the dimension of the matrices.
- From the **algorithmic complexity** point of view, the size of the Jacobian, condition number of the Jacobian and the number of Newton iterations are significantly less than that in multiple shooting and collocation methods.
- Applicable to linear as well as nonlinear subsystems.
- Applicable to different operation modes.
- It can handle the crossing-sliding and grazing-sliding segments.

K. Mandal et al., "An Automated Algorithm for Stability Analysis of Hybrid Dynamical Systems," *European Physical Journal (EPJ)*, 2013 .

Comparison with the Available Programs

Algorithmic complexity:

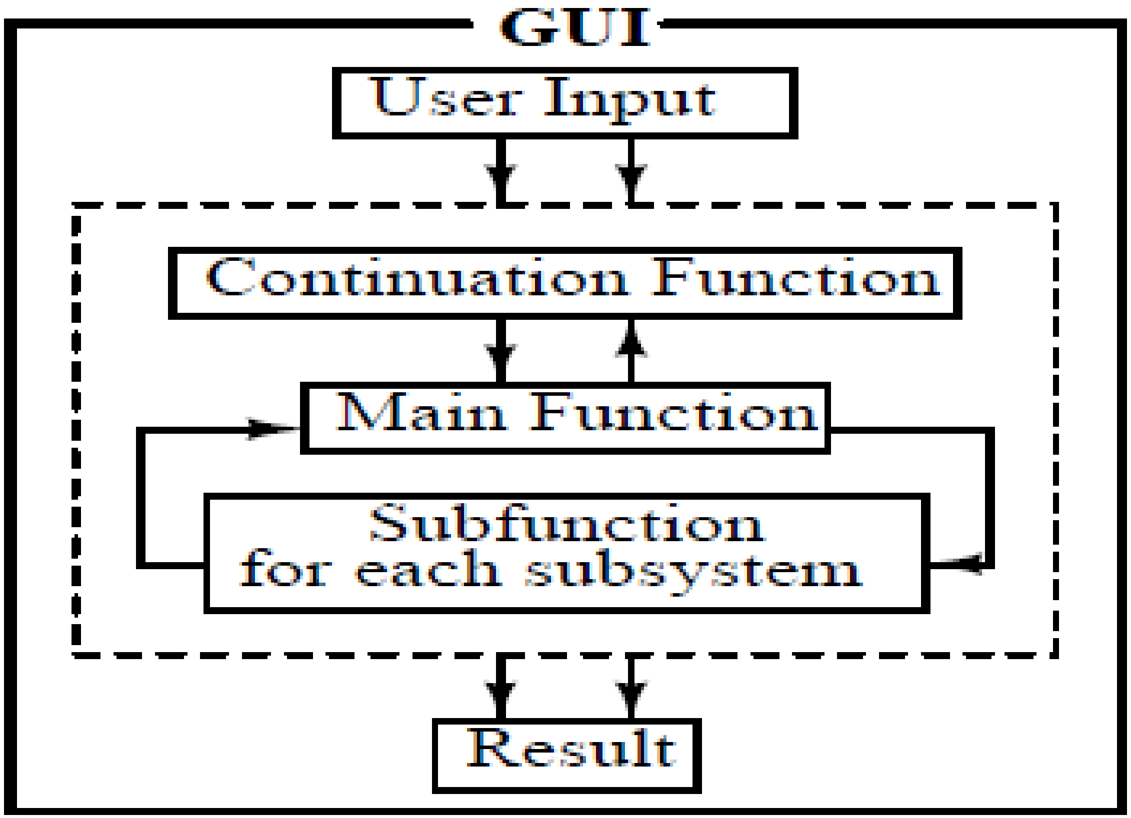
- the size of the Jacobian,
- condition number of the Jacobian,
- the number of Newton iterations,
- the domains of convergence.

In collocation method, the Jacobian is of dimension $(mnN + b + q + 1)$, where n is the number of subintervals, m the number of collocation points per subinterval, N the dimension of the system, b the number of boundary conditions and q the number of integral conditions.

In the multiple-shooting methods, the Jacobian is of dimension $(N - 1)s + 1$, where s is the number of cross-sections that are used.

For systems with high complexity, these become unmanageable.

Program Structure inside GUI



Event Detection Routine in Matlab

The function is of the form:

function [value,isterminal,direction] = events(t,y)

- value(i) is the value of the i^{th} event function.
- isterminal(i) = 1 if the integration is to terminate at a zero of this event function,
  0 otherwise
- direction(i) = 0 if all zeros are to be located (the default),
  +1 only zeros for increasing event function and
  -1 only zeros for decreasing event function

Ref: F. Zhang, M. Yeddanapudi, P. J. Mosterman, Zero-Crossing Location and Detection Algorithms For Hybrid System Simulation, The MathWorks, USA, 2008.

Errors and Control of Errors in Matlab

Accuracy vs Speed

RelTol: the relative accuracy tolerance, controls the number of correct digits in the answer.

AbsTol: the absolute error tolerance, controls the difference between the answer and the solution.

At each step, the error e in component i of the solution satisfies

$|e(i)| \leq \max(RelTol*abs(y(i)),AbsTol(i))$

Ref: T. Park and P. I. Barton, State event location in differential-algebraic models, *ACM Transactions on Modeling and Computer Simulation,* 996.

Bifurcation and Stability Analysis

Bifurcation Diagram

and Phase-Space: Direct time integration

Unstable Periodic Orbit: Single Shooting method

Coexisting Periodic Orbits: Natural Continuation Method

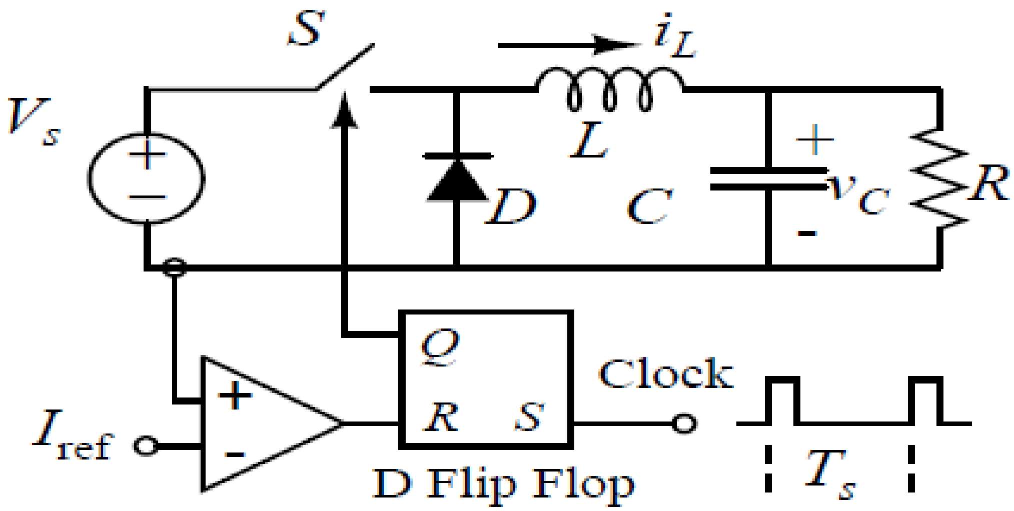
Bifurcation Point: Monitoring the Maximum Eigenvalue of the Jacobian.

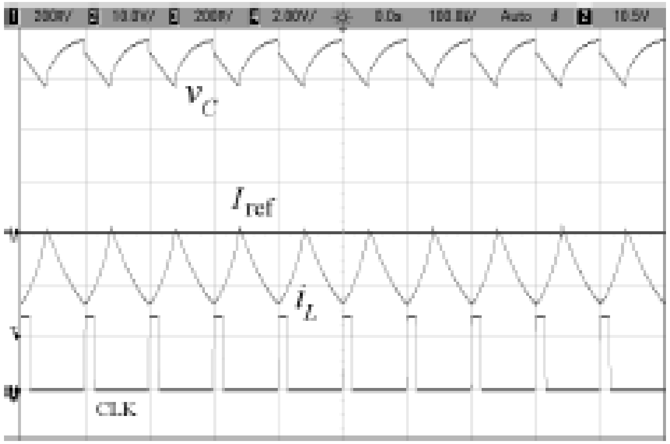
GUI in Matlab

Write ‘guide’ in Matlab Command window

Ref: Book by P. Marchand and O. T. Holland, *Graphics and GUIs with MATLAB,* 2003.

Example 1: Buck Converter with Current-Mode Control





Waveforms

Parameters
$V_s = 20$ V, $R_L = 10\ \Omega$, $f_s = 30$ kHz,
$L = 0.62$ mH, $C = 1$ mF, $I_{\text{ref}} = 1$ A.

Buck Converter: Modeling

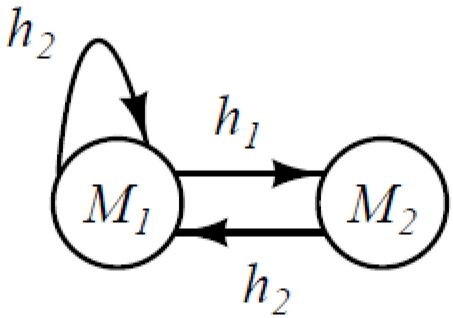
From "hybrid system" point of view the system can be modeled as

$$\frac{dx}{dt} = \begin{cases} M_1 : \boldsymbol{A}_1 x + \boldsymbol{B}_1 u & \text{S is ON} \\ M_2 : \boldsymbol{A}_2 x + \boldsymbol{B}_2 u & \text{S is OFF} \end{cases}$$

where, $x = [\, i_L \ v_C \,]^T = [\, x_1 \ x_2 \,]^T$, $u = V_s$ and the coefficient matrices are

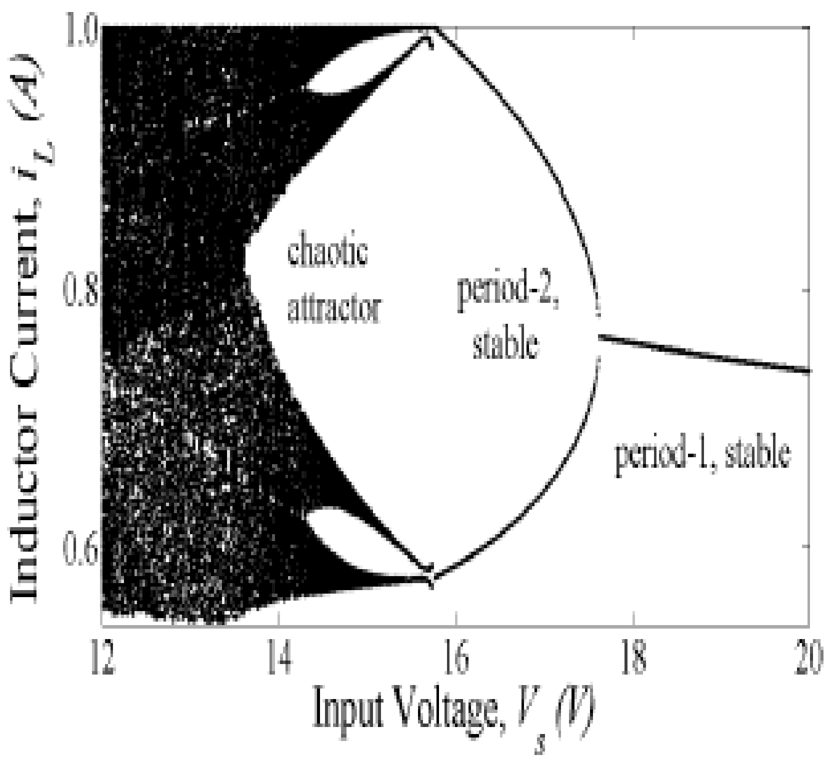
$$\boldsymbol{A}_1 = \boldsymbol{A}_2 = \begin{bmatrix} 0 & -\frac{1}{L} \\ \frac{1}{C} & -\frac{1}{RC} \end{bmatrix}, \quad \boldsymbol{B}_1 = \begin{bmatrix} \frac{1}{L} \\ 0 \end{bmatrix}, \quad \boldsymbol{B}_2 = \begin{bmatrix} 0 \\ 0 \end{bmatrix}.$$

The switching surfaces are given by $h_1 : x_1 - I_{\text{ref}} = 0$, and $h_2 : t \ mod \ T_s = 0$.

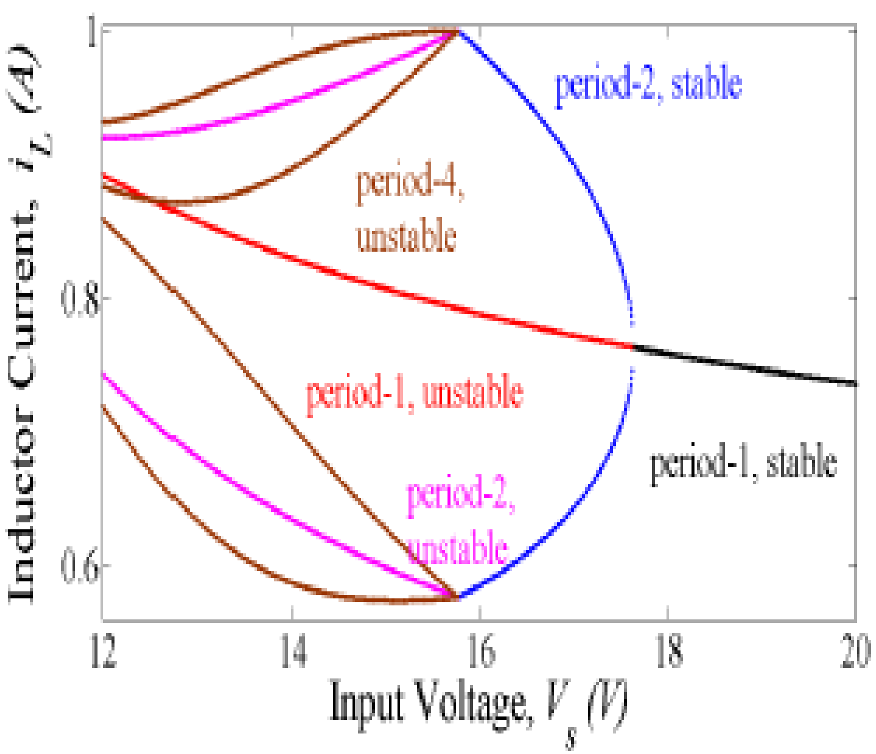


Buck Converter: Bifurcation Diagrams

Brute-Force



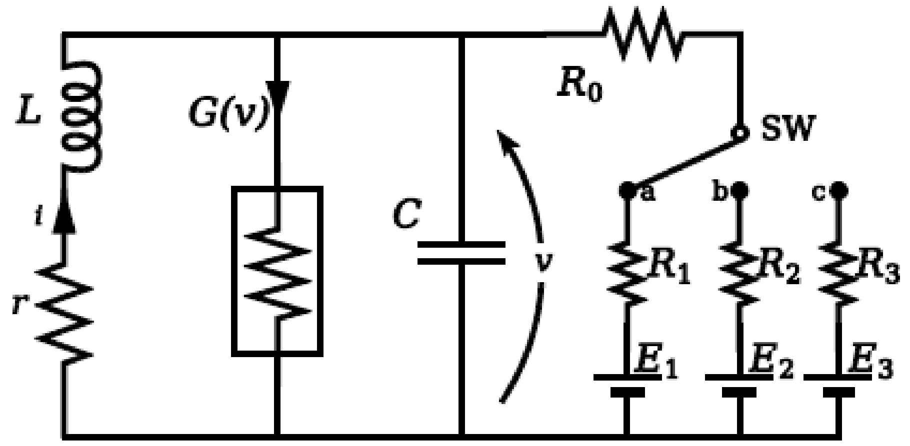
Path-Following



Buck Converter: Fixed point and Eigenvalues

V_s	Periodic orbit	Subsystem sequence	Fixed point	Eigenvalues	Type
20.00	stable period-1	[1-2]	$[0.7359\ 8.6792]$	$-0.7668, 0.9966$	
17.631	stable period-1	[1-2]	$[0.763015\ 8.81507]$	$-0.9999, 0.9967$	
17.630	unstable period-1	[1-2]	$[0.763028\ 8.81514]$	$-1.0000, 0.9967$	smooth
17.630	stable period-2	[1-2 – 1-2]	$[0.7604\ 8.8150]$	$0.9999, 0.9935$	
15.763	stable period-2	[1-2 – 1-2]	$[0.5763\ 7.8815]$	$0.9931 \pm 0.0842j\ (\simeq 0.9967)$	
15.762	unstable period-2	[1 – 1-2]	$[0.5762\ 7.8811]$	$-1.0000, 0.9934$	nonsmooth
15.762	unstable period-4	[1-2 – 1-2 – 1 – 1-2]	$[0.5762\ 7.8810]$	$-1.0007, 0.9868$	nonsmooth

Example 2: 2-D, Three-Subsystems, 10 Parameters Alpazur Oscillator



Example 2: Modeling of Alpazur Oscillator

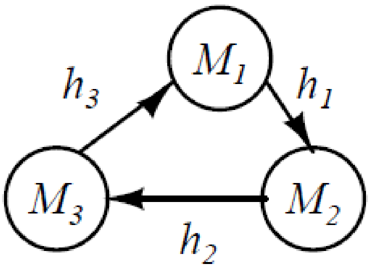
Modeling: Piecewise Nonlinear Subsystem (Normalized voltage and current)

For $i = 1,\ 2,\ 3$–

$$M_i: \quad \begin{bmatrix} \frac{dx_1}{dt} \\ \frac{dx_2}{dt} \end{bmatrix} = \begin{bmatrix} rx_1 - x_2 \\ x_1 + (1 - A_i)x_2 - \frac{1}{3}{x_2}^3 + B_i \end{bmatrix}$$

The switching surfaces are:

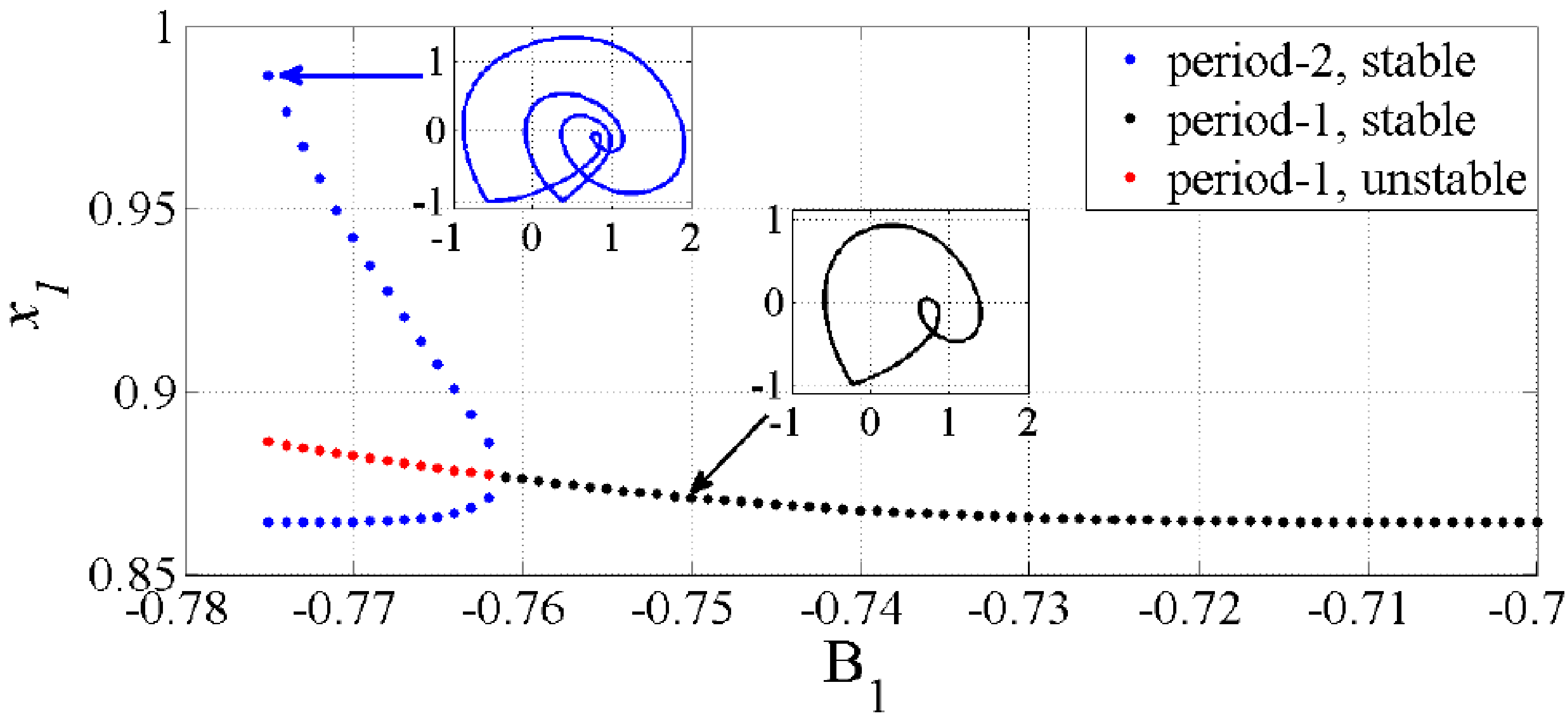
$$\begin{aligned} h_1 &: \quad x_2 - h = 0, \\ h_2 &: \quad x_2 - b = 0, \\ h_3 &: \quad x_2 - m = 0. \end{aligned}$$



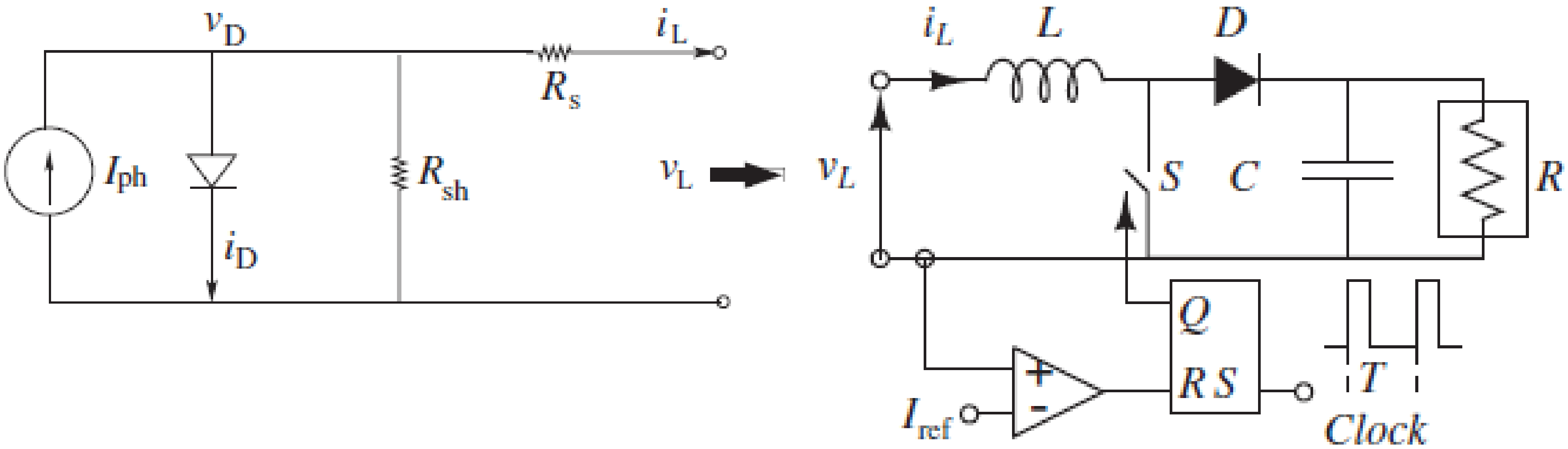
Parameters
$r = 0.1,\ A_1 = 0.2,\ A_2 = 2.0,\ A_3 = 0.8,\ B_1 = -0.70$ $B_2 = -0.8,\ B_3 = -0.1,\ h = -1,\ b = -0.3,\ m = -0.1.$

Q. Brandon et al., Numerical bifurcation analysis framework for autonomous piecewise-smooth dynamical systems, Chaos, Solitons and Fractals, 2009.

Alpazur Oscillator: Bifurcation Diagram



Example 3: PV Connected Boost Converter



Parameters
$I_{ph} = 1$ A, $I_o = 10^{-10}$ A, $R_s = 3\ \Omega$, $R_{sh} = 100\ \Omega$, $A = 3.8647$
$I_{ref} = 1$ A, $L = 3.125$ mH, $C = 20\ \mu$F, $R = 3\ \Omega$, $T_s = 100\ \mu$s

State-Space Modeling: PV Connected Boost Converter

The system can be modeled using differential-algebraic equations as

$$\frac{dx}{dt} = \begin{cases} M_1 : \boldsymbol{A}_1 x + \boldsymbol{B}_1 & \text{S is ON, D is OFF} \\ M_2 : \boldsymbol{A}_2 x + \boldsymbol{B}_2 & \text{S is OFF, D is ON} \end{cases}$$

$$I_{\text{ph}} - I_o(e^{A(v_L + i_L R_s)} - 1) - \frac{v_L + i_L Rs}{R_{\text{sh}}} = i_L$$

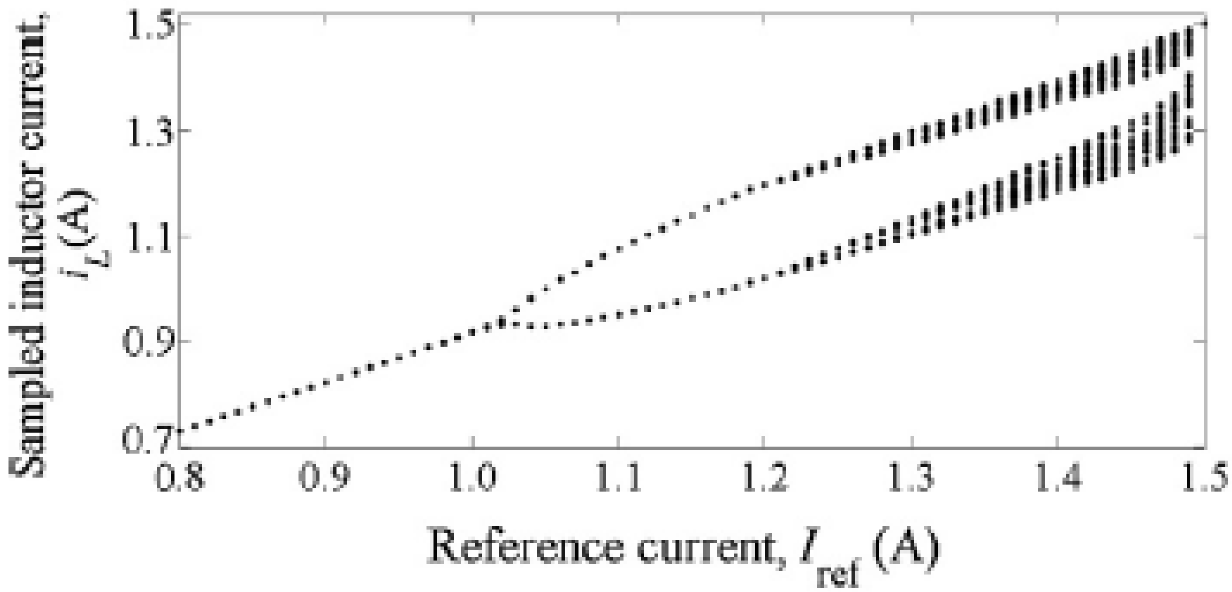
where, $x = [\, i_L \;\; v_C \,]^\top = [\, x_1 \;\; x_2 \,]^\top$, and

$$\boldsymbol{A}_1 = \begin{bmatrix} 0 & 0 \\ 0 & -\frac{1}{RC} \end{bmatrix}, \;\; \boldsymbol{A}_2 = \begin{bmatrix} 0 & -\frac{1}{L} \\ \frac{1}{C} & -\frac{1}{RC} \end{bmatrix}, \;\; \boldsymbol{B}_1 = \boldsymbol{B}_2 = \begin{bmatrix} \frac{v_L}{L} \\ 0 \end{bmatrix}.$$

The switching surfaces are -

$h_1 : x_1 - I_{\text{ref}} = 0$, and $h_2 : t \; mod \; T_s = 0$.

PV Connected Boost Converter



I_{ref} (A)	Orbit	Subsystem sequence	Eigenvalues
0.8	Stable Period-1	M_1–M_2	−0.7447, 0.5744
1.0	Stable Period-1	M_1–M_2	−0.9903, 0.5813
1.01	Unstable Period-1	M_1–M_2	−1.0023, 0.5816
1.01	Stable Period-2	M_1–M_2; M_1–M_2	0.9991, 0.3612

Ref: A. Abusorrah et al., Stability of the Boost Converter Fed from Photovoltaic Source, *Solar Energy*, 2013.

Demonstration

Conclusions

A general-purpose software is developed to handle high-dimensional switching dynamical systems with a large number of subsystems.

THANK YOU