2014 International Symposium on Nonlinear Theory and its Applications
NOLTA2014, Luzern, Switzerland, September 14-18, 2014

A new software for stability and bifurcation analysis of switched dynamical systems

K. Mandal†‡, A. Abusorrah†, M. M. Al-Hindawi†, Y. Al-Turki†, and S. Banerjee‡†

†Department of Electrical and Computer Engineering, King Abdulaziz University, Jeddah, Saudi Arabia
‡Indian Institute of Science Education and Research–Kolkata, Mohanpur Campus, Nadia – 741252, India
Email: kmandal1980@gmail.com

Abstract—In this paper, we introduce a new MATLAB-based program for numerical analysis of switched dynamical systems. The program provides a graphical user interface (GUI) that allows the users to enter the system description in terms of the symbolic ordinary differential equations for each subsystem, along with the information about the switching surfaces. Using this program, one can locate the periodic orbits (stable as well as unstable) and can calculate their eigenvalues. The algorithm calculates the fundamental solution matrix using exponential matrices for evolution within the subsystems, and the saltation matrices for transition across subsystems through switching conditions. The tool is verified using several hybrid dynamical systems.

1. Introduction

Brute-force simulation of the complex Filippov-type hybrid dynamical systems can be done by many available softwares (for example SIMULINK). But a computational tool for stability and bifurcation analysis of such systems provides distinct advantage over such brute-force simulation. Stability and bifurcation analysis offers explanation of the change in behavior which is not possible by simple system simulation.

Traditionally, collocation-based algorithms like AUTO are used for this purpose. However, in the softwares like AUTO (and specialized drivers, such as SLIDECONT [1], HOMCONT [2], TC-HAT ($\widehat{TC}$) [3]), and MATCONT [4] (in MATLAB environment), the entire event order has to be predescribed. The AUTO-based collocation method also has the limitation on problem size: For example in TC-HAT ($\widehat{TC}$), if a hybrid periodic trajectory corresponding to an N dimensional system has S segments, then NS has to be less than 100. An important limitation of the collocation method is that it requires all the segments be equally meshed irrespective of their length, i.e., even very small segments (almost zero length) have to have the same number of mesh points as the largest segment in the periodic trajectory. This limitation places constraints on the format of the initial solution used for continuation.

Kowalczyk and Piiroinen also developed an algorithm [5] based on fundamental solution matrix to analyze sliding bifurcations of Filippov systems. Though the algorithm is capable of handling high dimensional systems with a large number of subsystems, the system analyzed in [5] has low dimension and less number of subsystems. Since the program was not openly available, it was not possible for us to check its applicability in the complex systems being handled in this paper.

Ma et al. developed a method [6], which locates the periodic orbit by the Newton-Raphson method, and in the process of convergence, obtains the Jacobian for calculating the Floquet multipliers. However, this method also has the limitation that the orbit should contain a small number of switchings, because in its Newton-Raphson search algorithm the size of the Jacobian matrix depends on that number.

In the present work, we present a software for detection of periodic orbits (stable and unstable). It uses the shooting method to detect the periodic orbits, and the Jacobian matrix needed for the algorithm is calculated from the fundamental solution matrix for one period of the trajectory starting from the initial guess. In each step the point is updated as well as the corresponding Jacobian matrix. Thus, when the algorithm converges, it detects the fixed point as well as its eigenvalues.

2. The Shooting Method

Suppose a hybrid dynamical system is comprised of m subsystems M_1, M_2, $M_3 \cdots M_m$, and the solution flow is described as

$$\begin{aligned}
\boldsymbol{x}_1 &= \varphi_1(\tau_1, \tau_0, \boldsymbol{x}_0), \text{ for } \boldsymbol{x} \in M_1 : \dot{\boldsymbol{x}} = \boldsymbol{f}_1 = \boldsymbol{A}_1 \boldsymbol{x} + \boldsymbol{B}_1 \\
&\vdots \\
\boldsymbol{x}_m &= \varphi_m(pT, \tau_{m-1}, \boldsymbol{x}_{m-1}), \\
&\quad \text{for } \boldsymbol{x} \in M_m : \dot{\boldsymbol{x}} = \boldsymbol{f}_m = \boldsymbol{A}_m \boldsymbol{x} + \boldsymbol{B}_m
\end{aligned}$$

Starting from ($\boldsymbol{x}_0$, τ_0), the solution flow crosses the switching surfaces at time instants $\tau_1, \tau_2 \cdots, \tau_{m-1}$ before it reaches the state $\boldsymbol{x}_m$ at time pT. Then the Newton-Raphson search rule for locating the period-p fixed point can be formulated in the following way. Suppose an initial guess for the fixed point is $\boldsymbol{x}_0$. Then the next step in the Newton-Raphson procedure is

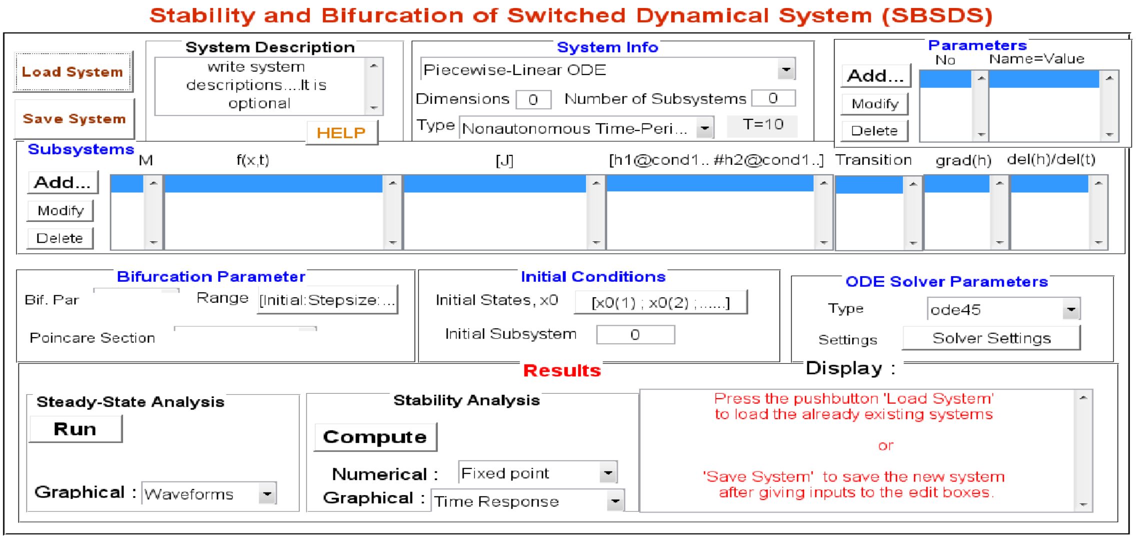

Figure 1: Screenshots of the Matlab GUI for implementation of the algorithm.

given by

$$\boldsymbol{x}_0' = \boldsymbol{x}_0 - \left(\frac{\partial \boldsymbol{x}_m}{\partial \boldsymbol{x}_0} - \boldsymbol{I}_N\right)^{-1} (\boldsymbol{x}_m - \boldsymbol{x}_0)$$

where $\boldsymbol{I}_N$ is an N-dimensional identity matrix. This requires the Jacobian for the fundamental solution matrix which is calculated as

$$\frac{\partial \boldsymbol{x}_m}{\partial \boldsymbol{x}_0} = \boldsymbol{S}_m \frac{\partial \boldsymbol{x}_m}{\partial \boldsymbol{x}_{m-1}} \boldsymbol{S}_{m-1} \frac{\partial \boldsymbol{x}_{m-1}}{\partial \boldsymbol{x}_{m-2}} \boldsymbol{S}_{m-2} \cdots \frac{\partial \boldsymbol{x}_2}{\partial \boldsymbol{x}_1} \boldsymbol{S}_1 \frac{\partial \boldsymbol{x}_1}{\partial \boldsymbol{x}_0}$$

If the system is piecewise linear, the state transition matrices for the flow across each subsystem are nothing but the exponential matrices

$$\frac{\partial \boldsymbol{x}_1}{\partial \boldsymbol{x}_0} = e^{\boldsymbol{A}_1(\tau_1 - \tau_0)}, \quad \frac{\partial \boldsymbol{x}_2}{\partial \boldsymbol{x}_1} = e^{\boldsymbol{A}_2(\tau_2 - \tau_1)} \ldots$$

It can be evaluated using MATLAB's *expm* function which is ten-term Taylor's series approximation. For piecewise nonlinear subsystem (e.g., Alpazur oscillator) the method given in [7] has been used.
The first saltation matrix (or the state transition matrix for the passage across the first switching condition) is given by

$$\boldsymbol{S}_1 = \boldsymbol{I}_N + \frac{(\boldsymbol{f}_2 - \boldsymbol{f}_1)\boldsymbol{n}_1^\top}{\boldsymbol{n}_1^\top \boldsymbol{f}_1 + \frac{\partial h_1}{\partial t}|_{t=\tau_1}}.$$

Here $\boldsymbol{n}_1$ is the normal vector to the switching surface $h_1(t, \boldsymbol{x}) = 0$, and $\boldsymbol{n}_1^\top$ is its transpose. The saltation matrices for the transitions between the other subsystems are obtained in a similar way. Thus once the switching time instants and the flow equations in each subsystem are known, the fundamental solution matrix can be obtained.

3. Program Architecture and Implementation

- Graphical user interface (GUI): MATLAB provides a mechanism to generate GUI by using GUIDE (the standard tool within MATLAB). It allows the end users to use the program with minimal knowledge and input. Using this functionality, we have created a GUI for our program (shown in Figure 1), in which some windows are for user interaction—typically input acquisition, parameter tuning and option selection tasks. Others are windows to display graphical output.
- Main Function: It provides the basic simulation routine steady-state and stability analysis as shown in Figure 2. The parameters, the initial subsystem, the initial values of the states and time (by default zero), the final time are given as input. For nonautonomous time-periodic system the final time is a fixed value but in case of autonomous system the time period is detected after steady state analysis. Using this information, it calculates the trajectory for a time period and on that basis it computes the state transition matrices across the subsystems and the saltation matrices. On that basis it calculates the Jacobian, and takes a Newton-Raphson step. This is repeated until it converges. Thus this routine calculates the fixed point and eigenvalues. If the parameters are changed, this algorithm can follow the periodic orbit by a continuation method.
- Subfunction for each subsystem: It receives input (state and time) from the main function and

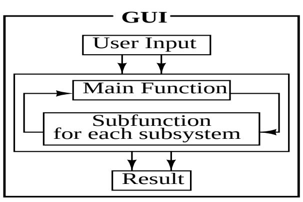

Figure 2: Structure of the program inside the GUI.

computes a set of differential equations. In each subsystem there are some switching conditions indicating transitions to other subsystems. When a switching condition is satisfied the integration stops, and the time instant and the state vector at that instant are stored and used as the starting value for the next subsystem. The computational work is performed by the integrators, which are the standard MATLAB ODE [8] solvers with built-in event detection routines. The solution is sent back to main function for graphical representation.

- The necessary results are shown graphically and stored in an appropriate format for the further use (i.e., if in a parameter range more than one solution exists, the final bifurcation diagram will be drawn by superimposition of all the solutions).

4. User Actions

For the user, the first job is to introduce a new system by giving input to the edit fields and click 'Save System' or load an existing one by clicking 'Load System' as shown in Figure 1. Filling the fields is straightforward since the MATLAB syntax is used. There are six mandatory fields of interest, namely, System Info, Parameters, Subsystems, Bifurcation Parameter, Initial Conditions, and ODE Solver Parameters. Once the data are given in the appropriate fields, the user clicks the 'Run' or 'Compute' pushbutton in Results depending on the steady-state or stability analysis. In stability analysis 'Fixed point', 'Eigenvalues', 'Subsystem sequence', and 'Bifurcation type' will be displayed by clicking the radio buttons. The program stops if the event detection routine does not work properly. One can change the 'ODE Solver Parameters' to influence the event detection routine. Graphical representation of the results is also possible.

5. Some Illustrative Examples

Many switched dynamical systems were taken from [5, 7, 9] and were tested by this newly developed tool. Here, two systems are chosen to show the applicability of the proposed tool.

5.1. Buck converter with current-mode control

A current controlled dc-dc buck converter is taken as an example [10] as shown in Figure 3. In this converter, the switch (S) turns off when the inductor current i_L reaches a pre-specified reference value I_{ref}. It turns on at the arrival of the next pulse from a free-running clock. If a clock pulse arrives while the switch is on, it is ignored.
From "hybrid system" point of view the system can be modeled as

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

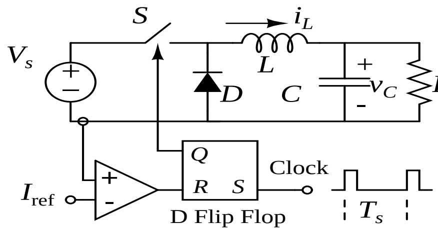

Figure 3: Circuit diagram of a buck converter with current mode control. The parameter values are: $L = 0.62$ mH, $C = 1$ mF, $R = 10\ \Omega$, $f_s = 1/T_s = 30$ kHz, $I_{\text{ref}} = 1$ A.

where, $\boldsymbol{x} = [\, i_L \ v_C \,]^\top = [\, x_1 \ x_2 \,]^\top$, and the coefficient matrices are

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

The first and second subsystems are denoted by M_1 and M_2 respectively. The switching surfaces are given by $h_1 : x_1 - I_{\text{ref}} = 0,$ and $h_2 : t \ mod \ T_s = 0.$

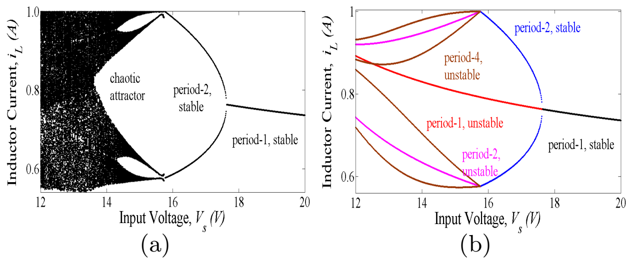

Figure 4: Bifurcation diagrams with V_s as varying parameter (a) brute-force (only stable attractors), (b) path-following (stable and unstable periodic orbits).

After giving the inputs to the tool, the usual approach to analyze the dynamics of the system is often to make a parameter sweep and to create a bifurcation diagram by direct numerical simulation (DNS). By choosing the input voltage V_s as the bifurcation parameter, with the variation (decreasing value) of it a bifurcation diagram as shown in Figure 4(a) has been obtained which shows the stable attractors. A path following bifurcation diagram (using the combination of shooting method with Newton-Raphson and continuation algorithm) is also drawn which shows the stable and the unstable periodic orbits (Figure 4(b)) to explain the underlying behavior more accurately. Both the bifurcation diagrams have been drawn by the developed software.

All the bifurcation points and their types are identified by monitoring the eigenvalues of the Jacobian matrix of the fixed-point as shown in Table 1.

5.2. Load resonant converter with fixed frequency control

A phase-shift modulated non-isolated series-parallel load resonant dc-dc converter [11] is considered to il-

Table 1: Eigenvalues of the fixed point with variation of input voltage V_s.

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± 0.0842j (≃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

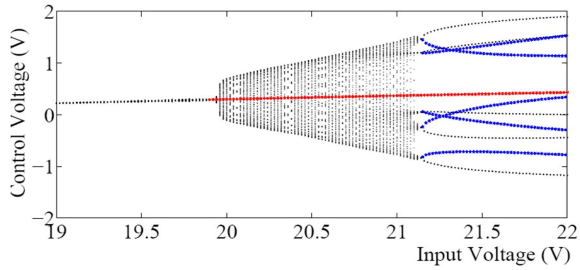

Figure 5: Bifurcation diagram with input voltage as varying parameter. The orbits in black color are drawn by brute-force simulation. The red and blue colors indicate unstable period-1 and unstable period-5 orbit respectively.

lustrate the ability of this algorithm to work with systems with a high degree of complexity. In this system, a five-dimensional state space is divided into nine subsystems by four switching surfaces. The bifurcation diagram (Figure 5) shows that a Neimark-Sacker and saddle-node bifurcations have occurred at $V_s = 19.83$ V and $V_s = 21.14$ V respectively. A mode-locked period-5 orbit after second bifurcation consists of 32 subsystems, which also includes sliding segments.

6. Conclusion

This paper focuses on the numerical analysis of Filippov type complex hybrid dynamical systems. In this work, a new general purpose Matlab-based program for the stability analysis of any hybrid dynamical system is introduced. Armed with an appropriate GUI, this tool is aimed at providing end-users with a powerful computer tool to perform such analysis. It is expected to be useful in carrying out a number of computations (fixed points, their stability, bifurcation diagram, phase space) in an efficient and simple way. This tool has no bar on problem size because each switching is considered separately by saltation matrices.

Acknowledgments

This project was supported by the NSTIP strategic technologies program in the Kingdom of Saudi Arabia-Project No.(12-ENE3049-03). The authors also, acknowledge with thanks the Science and Technology Unit, King Abdulaziz University for technical support.

References

[1] F. Dercole and Y. A. Kuznetsov, "SlideCont: an AUTO97 driver for bifurcation analysis of Filippov systems," *ACM Transactions on Mathematical Software*, vol. 31, no. 1, pp. 95–119, March 2005.

[2] A. R. Champneys, Y. A. Kuznetsov, and B. Sandstede, "A numerical toolbox for homoclinic bifurcation analysis," *International Journal of Bifurcation and Chaos*, vol. 6, pp. 867–887, 1996.

[3] P. Thota and H. Dankowicz, "TC-HAT ($\widehat{TC}$) : A novel toolbox for the continuation of periodic trajectories in hybrid dynamical systems," *SIAM Journal of Applied Dynamical Systems*, vol. 7, no. 4, pp. 1283–1322, June 2008.

[4] A. Dhooge, W. Govaerts, and Y. A. Kuznetsov, "MATCONT: A MATLAB package for numerical bifurcation analysis of ODEs," *ACM Transactions on Mathematical Software*, vol. 29, no. 2, pp. 141–164, June 2003.

[5] P. Kowalczyk and P. Piiroinen, "Two-parameter sliding bifurcations of periodic solutions in a dry-friction oscillator," *Physica D*, vol. 237, pp. 1053–1073, 2008.

[6] Y. Ma, H. Kawakami, and C. K. Tse, "Bifurcation analysis of switched dynamical systems with periodically moving borders," *IEEE Transactions on Power Electronics*, vol. 51, no. 6, pp. 1184–1193, June 2004.

[7] Q. Brandon, T. Ueta, D. Fournier-Prunaret, and T. Kousaka, "Numerical bifurcation analysis framework for autonomous piecewise-smooth dynamical systems," *Chaos, Solitons and Fractals*, vol. 42, pp. 187–201, July 2009.

[8] L. F. Shampine, "Design of software for ODEs," *Journal of Computational and Applied Mathematics*, vol. 205, pp. 901–911, 2007.

[9] A. E. Aroudi, M. Debbat, R. Giral, G. Olivar, L. Benadero, and E. Toribio, "Bifurcations in dc-dc switching converters:review of methods and applications," *International Journal of Bifurcation and chaos*, vol. 15, no. 5, pp. 1549–1578, 2005.

[10] S. Banerjee, S. Parui, and A. Gupta, "Dynamical effects of missed switching in current-mode controlled dc-dc converters," *IEEE Transactions on Circuits and Systems. II*, vol. 51, no. 12, pp. 649–654, December 2004.

[11] K. Mandal, S. Banerjee, and C. Chakrabarty, "Bifurcations in load resonant dc-dc converters." Paris, France: IEEE International Symposium on Circuits and Systems, 30th May-2nd June 2010.