

We are IntechOpen, the world's leading publisher of Open Access books Built by scientists, for scientists

4,800

Open access books available

122,000

International authors and editors

135M

Downloads

Our authors are among the

154

Countries delivered to

TOP 1%

most cited scientists

12.2%

Contributors from top 500 universities



WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index
in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?
Contact book.department@intechopen.com

Numbers displayed above are based on latest data collected.

For more information visit www.intechopen.com



Simulation of Piecewise Hybrid Dynamical Systems in Matlab

Fatima El Guezar and Hassane Bouzahir

Additional information is available at the end of the chapter

<http://dx.doi.org/10.5772/48570>

1. Introduction

A hybrid dynamical system is a system containing on the same time continuous state variables and event variables in interaction. We find hybrid systems in different fields. We cite robotic systems, chemical systems controlled by vans and pumps, biological systems (growth and division) and nonlinear electronics systems.

Because of interaction between continuous and discrete aspects, the behavior of hybrid systems can be seen as extremely complex. However, this behavior becomes relatively simple for piece-wise affine hybrid dynamical systems that can, in contrast, generate bifurcation and chaos. There are many examples such as power electronics DC-DC converters.

The common power electronics DC-DC converters are the buck converter and the boost converter. They are switching systems with time variant structure [9].

DC-DC converters are widely used in industrial, commercial, residential and aerospace environments. These circuits are typically controlled by PWM (Piece Wise Modulation) or other similar techniques to regulate the tension and the current given to the charges. The controller decides to pass from one configuration to another by considering that transitions occur cyclically or in discrete time. In order to make the analysis possible, most of mathematical treatments use some techniques that are based on averaging or discretization. Averaging can mean to wrong conclusions on operation of a system. Discrete models do not give any information on the state of the system between the sampled instants. In addition, they are difficult to obtain. In fact, in most cases, a pure analytic study is not possible. Another possible approach to analyze these converters can be done via some models of hybrid dynamical systems. DC-DC converters are particularly good candidates for this type of analysis because of their natural hybrid structure. The nature of commutations of these systems makes them strongly nonlinear. They present specific complex phenomena such as fractals structures of bifurcation and chaos.

The study of nonlinear dynamics of DC-DC converters started in 1984 by works of Brockett and Wood [4]. Since then, chaos and nonlinear dynamics in power electronics circuits have attracted different research groups around the world. Different nonlinear phenomena have been observed such as routes to chaos following the period doubling cascade [16], [5], [19], [20] and [23] or quasi-periodic phenomena [6], [7] and [8], besides border collision bifurcations [23] and [2].

Switched circuits behavior is mostly simulated by pure numerical methods where precision step is increased when the system is near a switching condition. Those numerical tools are widely used mainly because of their ease-of-use and their ability to simulate a wide range of circuits including nonlinear, time-variant, and non-autonomous systems.

Even if those simulators can reach the desired relative precision for a continuous trajectory, they can miss a switching condition and then diverge drastically from the trajectory as in figure 1. This could be annoying when one is interested by border collision bifurcations, or when local behavior is needed with a good accuracy. In those applications, an alternative is to write down analytical, or semi-analytical, trajectories and switching conditions to obtain a recurrence which is very accurate and fast to run. Building and adapting such *ad hoc* simulators represent a lot of efforts and a risk of mistakes.

Generic and accurate simulators can be proposed if we are restricted to a certain class of systems. A simulation tool with no loss of events is proposed in [14] and [15] for PWAHSs defined on polytopes (finite regions that are bounded by hyperplanes). This class of PWA differential systems has been widely studied as a standard technique to approximate a range of nonlinear systems.

But closed polytopic partition of the state space does not allow simulation of most switching circuits where switching frontiers are mostly single affine constraints or time-dependent periodical events.

This chapter follows our previous study in [13], [12] and [11].

We focus on planar PWAHSs with such simple switching conditions which can model a family of switched planar circuits: bang-bang regulators, the Boost converter, the Charge-Pump Phase Locked Loop (CP-PLL), ...

This class of systems has analytical trajectories that help to build fast algorithms with no loss of events. We propose a semi-analytical solver for hybrid systems which provides:

- A pure numerical method when the system is nonlinear or non-planar;
- A pure analytic method when all continuous parts of the system and switching conditions can be solved symbolically. This can be the case for the boost converter [3], [21], the second order charge-pump phase locked loop [17], [22].
- A mixed method using analytical trajectories and numerical computation of the switching instant when those solutions are transcendent. This has been used for the third order CP-PLL [17]. It can also be the case for the Buck converter [10], [21], ...

This chapter is organized as follows. Section 2 contains our main results. We describe the problem to be deal with, we introduce a general algorithm to solve planar HSs, we

present the algorithm that detects events' occurrence and devote a subsection to our approach efficiency. Section 3 Illustrates the current-mode controlled Boost converter example. Finally, a conclusion is stated in Section 4.

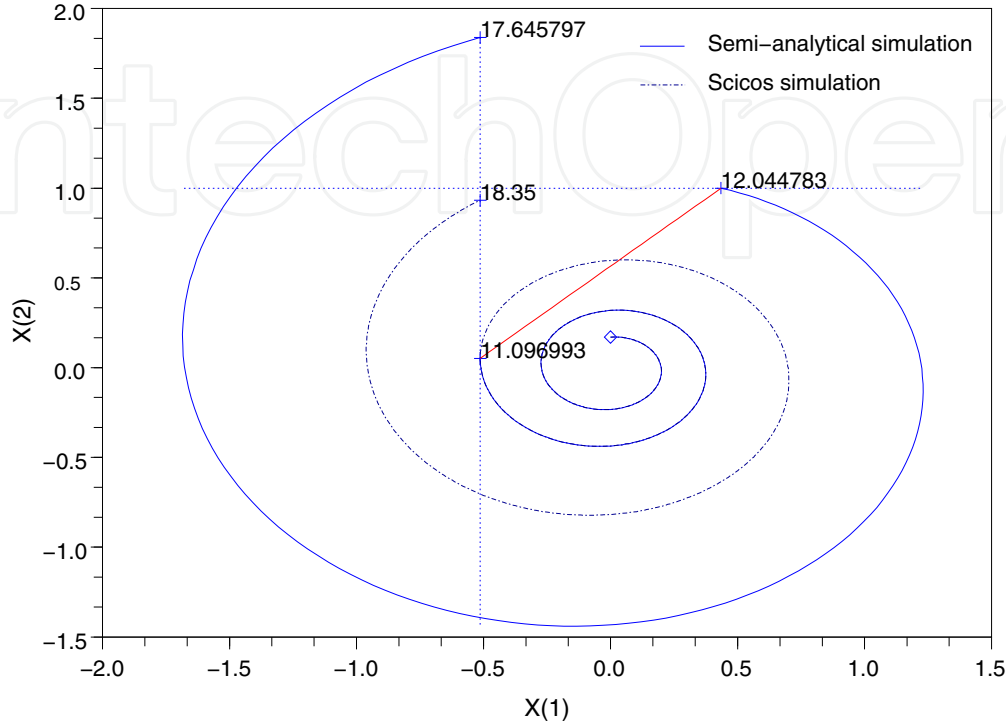


Figure 1. Numerical simulation versus semi-analytical simulation.

2. Main results

2.1. A HS (X, E, t) : general definition

A general definition of HSs is presented here. This type of dynamical systems is characterized by the coexistence of two kinds of state vectors: continuous state vector $X(t)$ of real values, and discrete state vector $E(t)$ belonging to a countable discrete set \mathcal{M} .

Definition 1. A continuous-time, autonomous HS is a system of the form:

$$\begin{aligned}\dot{X}(t) &= F(X(t), E(t)), F: \mathcal{H} \rightarrow \mathbb{R}^n \\ E^+(t) &= \phi(X(t), E(t)), \phi: \mathcal{H} \rightarrow \mathcal{M}\end{aligned}\quad (1)$$

$\mathcal{H} = \mathbb{R}^n \times \mathcal{M}$ is called the hybrid state space. $X(t) \in \mathbb{R}^n$ is the continuous state vector of the HS at time instant t and $E(t) \in \mathcal{M} := \{1, \dots, N\}$ is its discrete state. $E^+(t)$ denotes the updated discrete state right after time instant t . $\phi: \mathcal{H} \rightarrow \mathcal{M}$ describes the discrete dynamic, it is usually modeled by Petri Nets. A transition from $E(t) = i$ to $E^+(t) = j$ is valid when the state X reaches a switching set called \mathcal{S}_{E_i, E_j} . Such transitions are called state dependent events. A HS is called piece-wise affine if for each $E \in \mathcal{M}$, $F(X, E)$ can be defined by $F(X, E) = A_E X + B_E, \forall X$.

Remark — For non-autonomous HSs, the function ϕ can also depend on time $\phi(X, E, t) : \mathbb{R}^n \times \mathcal{M} \times \mathbb{R} \rightarrow \mathcal{M}$. Then time dependent events can occur and validate a transition, such as periodic events.

2.2. HSs class of interest

We consider a two dimensional PWAHS ($X(t) \in \mathbb{R}^2$). F has then the affine piece-wise form, $F(.,.)$ is defined for each $E \in \mathcal{M}$ and $X \in \mathbb{R}^2$ by $F(X(t), E(t)) = A_{E(t)}X + B_{E(t)}$, where $A_{E(t)} \in \mathbb{R}^{2 \times 2}$ and $B_{E(t)} \in \mathbb{R}^2$ are two matrices that depend on the discrete state $E(t)$. Hence, a two dimensional PWAHS is a HS that take the form:

$$\begin{aligned} \dot{X}(t) &= A_{E(t)}X + B_{E(t)}, \\ E(t) &\in \mathcal{M} = \{1, 2, \dots, N\} \end{aligned} \tag{2}$$

We consider two kinds of events: state dependent events and periodic events.

The state dependent event transition $\mathcal{S}_{E_i E_j}$ is defined by an affine state border of the form $N'_{ij} \cdot X < l_{ij}$. In this case an event can occur when the continuous state reaches the border of the set $\mathcal{S}_{E_i E_j} = \{X(t) \in \mathbb{R}^2 : N'_{ij} \cdot X \leq l_{ij}\}$.

Note that the set $\mathcal{S}_{E_i E_j}$ is not polytopic in the sense that the domain is not the interior of a closed bounded polytope.

Remark — We consider, without loss of generality, the case where a transition occurs at time $d\mathcal{S}_{E_i E_j}$ if and only if the state $X(d\mathcal{S}_{E_i E_j})$ reaches a border of the set $\mathcal{S}_{E_i E_j}$ from outside. Figure 2 defines a transition with the complimentary set $\bar{\mathcal{S}}$, which allows to detect the event in both directions. Both transitions can be met with the set $B = \mathcal{S} \cup \bar{\mathcal{S}}$. Periodic

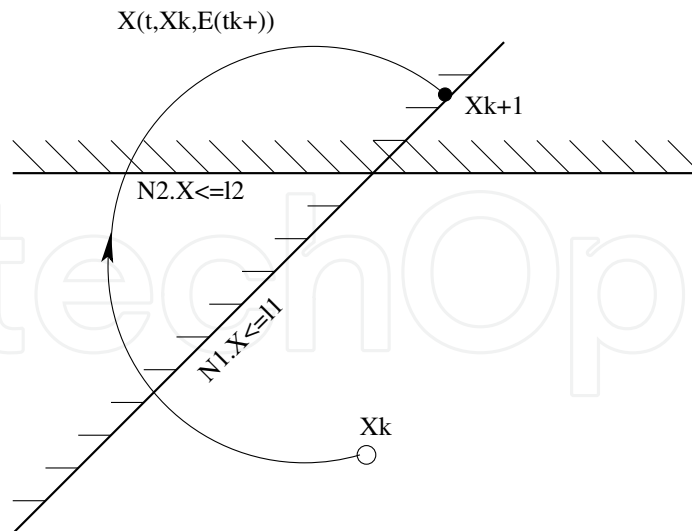


Figure 2. Oriented polytopic state dependent transitions.

events are simply defined by time instants $t = d\mathcal{P}_{E_i E_j}$, where $d\mathcal{P}_{E_i E_j}$ belongs to the set $\mathcal{P}_{E_i E_j} = \{t : t = kT + \varphi, k \in \mathbb{N}\}$. T is the period and φ is the phase of such periodic events.

2.3. Event-driven simulation of PWAHSs

The simulation will compute the hybrid state from event to event. Knowing the states $X(t_k)$ and $E(t_k^+)$, one can compute the trajectory $X(t > t_k) = \int_{t_k}^t f(X(t_k), E(t_k^+)) dt + X(t_k)$, assuming that the discrete state is constant $E(t > t_k^+) = E(t_k^+)$. Then the following algorithm runs the simulation determining the date at the next event as the smallest:

```

Data:  $t_k, X_k, E_k$ .
while  $t < t_{fin}$  do
  Compute all events' dates  $dS_{E_i E_j}$  and  $dP_{E_i E_j}$ ;
   $t_{k+1} = \min(dS_{E_i E_j}, dP_{E_i E_j})$ ;
   $X_{k+1} = f(X_k, E_k, t_{k+1})$ ;
   $E_{k+1} = \phi(X_k, E_k, t_{k+1})$ ;
end

```

Algorithm 1: Algorithm computing the hybrid state at t_{k+1} .

2.4. Event detection occurrence: description and algorithm

We consider the affine Cauchy problem in \mathbb{R}^2 :

$$\begin{cases} \dot{X}(t) = AX(t) + B, t > t_0 \\ X(t_0) = X_0 \end{cases} \quad (3)$$

where X_0 is the initial value. We compute the smallest strictly positive time t_i^* so that the trajectory of $X(t)$ intersects the fixed border B_i arriving from the part of the plan where $N_i' \cdot X < l_i$. The function $f_i(t) = N_i' \cdot X(t) - l_i$ defines the guard condition for a border B_i . Thus, the problem can be formulated as follows:

$$\text{Find the smallest } t_i^* \text{ such that } \begin{cases} f_i(t_i^*) = 0 \\ \exists \delta > 0, \forall t \in]t_i^* - \delta, t_i^*[, f_i(t) < 0 \end{cases} \quad (4)$$

If f_i does not have any strictly positive root or the last condition is not satisfied, t_i^* is given the infinite value.

2.4.1. Analytical trajectories

Definition 2. For any square matrix A of order n and t in \mathbb{R} , the exponential matrix e^{tA} is defined by

$$e^{tA} = \sum_{k=0}^{\infty} \frac{t^k A^k}{k!} = \mathbb{I} + tA + \frac{t^2 A^2}{2!} + \frac{t^3 A^3}{3!} + \dots \quad (5)$$

where \mathbb{I} is the identity matrix.

It is well known that the analytical trajectory $X(t)$ of the initial value matrix differential equation (3) is given in terms of the exponential matrix and the variation of constants formula

by the general integral form:

$$X(t) = e^{(t-t_0)A} X_0 + \int_{t_0}^t e^{(t-s)A} B ds. \tag{6}$$

When A is invertible, the above expression becomes linear:

$$X(t) + A^{-1}B = e^{(t-t_0)A} (X_0 + A^{-1}B) \tag{7}$$

The analytical expression of the exponential matrix e^{At} takes two forms depending on whether the eigenvalues p_1 and p_2 of the matrix A are equal or not:

If $p_1 \neq p_2$, then

$$e^{tA} = \frac{(p_1 \mathbb{I} - A^\circ)}{p_1 - p_2} e^{p_1 t} - \frac{(p_2 \mathbb{I} - A^\circ)}{p_1 - p_2} e^{p_2 t} \tag{8}$$

If $p_1 = p_2 = p$, then

$$e^{tA} = (\mathbb{I} + (p \mathbb{I} - A^\circ) t) e^{p t} \tag{9}$$

where $A = \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix}$, $A^\circ = \begin{pmatrix} a_{22} & -a_{12} \\ -a_{21} & a_{11} \end{pmatrix}$ and $\mathbb{I} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$.

Using these expressions, we can determine the function $f(t)$ of the problem (4) as follows:

$$f(t) = a_1 + a_2 t + a_3 t^2 + (a_4 + a_5 t) e^{p_1 t} + a_6 e^{p_2 t}$$

where a_i are real scalars.

Depending on the eigenvalues p_1 and p_2 , there are five cases that determine the values of the coefficients a_i as shown in Table 1. **Remark** — Coefficients a_i are real scalars that depend on

$f(t) = a_1 + \dots$	$p_1 \in \mathbb{R}^*$	$p_1 = 0$
$p_2 \in \mathbb{R}^*$	$a_4 e^{p_1 t} + a_6 e^{p_2 t}$	$a_2 t + a_6 e^{p_2 t}$
$p_2 = 0$	$a_2 t + a_4 e^{p_1 t}$	$a_2 t + a_3 t^2$
$p_1 = \overline{p_2} \in \mathbb{C}^*$	$a_4 e^{p_1 t} + a_5 e^{p_2 t}$, with $a_5 = \overline{a_4} \in \mathbb{C}^*$	
$p_1 = p_2 \in \mathbb{R}^*$	$(a_4 + a_5 t) e^{p_1 t}$	

Table 1. Expressions of $f(t)$ depending on the eigenvalues p_1 and p_2 .

the eigenvalues p_1 and p_2 , the initial point X_k and the border parameters are N_i and l_i .

In some cases, ($p_1 = p_2 = 0$, gray cell in Table 1) roots of $f(t)$ can be found analytically and the problem is solved with machine precision.

In other cases, the solution can not be found with classical functions and then a numeric algorithm should be used. Using classical methods like Newton does not guaranty existence or convergence of the smallest positive root. To meet these conditions, let us use analytical

roots of the derivative function $f'(t)$ expressed in Table 2. We can then compute analytically

$f'(t) = \dots$	$p_1 \in \mathbb{R}^*$	$p_1 = 0$
$p_2 \in \mathbb{R}^*$	$a_4 p_1 e^{p_1 t} + a_6 p_2 e^{p_2 t}$	$a_2 + a_6 p_2 e^{p_2 t}$
$p_2 = 0$	$a_2 + a_4 p_1 e^{p_1 t}$	$a_2 + 2 a_3 t$
$p_1 = \overline{p_2} \in \mathbb{C}^*$	$a_4 p_1 e^{p_1 t} + \overline{a_4 p_1} e^{\overline{p_1} t}$, with $a_4 \in \mathbb{C}^*$	
$p_1 = p_2 \in \mathbb{R}^*$	$(a_4 p_1 + a_5 + a_5 p_1 t) e^{p_1 t}$	

Table 2. Expressions of $f'(t)$ depending on the eigenvalues p_1 and p_2

the set L of ordered roots of $f'(t)$, those roots determines monotone intervals of $f(t)$. The following algorithm is used to return the solution t^* when it exists or the value ∞ if not.

Remark — When $(p_1, p_2) \in \mathbb{C}^* \times \mathbb{C}^*$ the set L is infinite: when the real part of p_i is positive, the algorithm

```

Data:  $N_i, l_i, A, B, X_k$ 
Result: construct the set  $L$ , compute  $t^*$ 
 $T \leftarrow \{0, L, \infty\}$ ;
 $t^* \leftarrow \infty$ ;
for  $i \leftarrow 1$  to  $(\text{card}(T) - 1)$  do
    if  $f(T(i)) < 0$  &  $f(T(i+1)) > 0$  then
         $t^* \leftarrow \text{solve}[T(i), T(i+1)]$ ;
        Break;
    end
end
end

```

Algorithm 2: Algorithm computing t^* when a solution is transcendent.

will end by finding a root. In the other case, the set L should be reduced to its three first elements, to find a crossing point when it exists.

3. Matlab modelling

Our semi-analytical solver is composed of different main programs that define the studied affine system. First, we create the affine system given with a specifically chosen name. Then, we define the matrices A_i and B_i . After that, we give the switching borders with the sign of transitions and all necessary elements or we give the period if it is about a periodic event. Finally, we execute the simulation by specifying the initial state and the time of simulation.

3.1. Application: Current-mode controlled Boost converter

A current-mode controlled Boost converter in open loop consists of two parts: a converter and a switching controller. The basic circuit is given in figure 3.

This converter is a second-order circuit comprising an inductor, a capacitor, a diode, a switch and a load resistance connected in parallel with the capacitor.

The general circuit operation is driven by the switching controller. It compares the inductor current i_L with the reference current I_{ref} and generates the on/off driving signal for the switch S . When S is on, the current builds up in the inductor.

When the inductor current i_L reaches a reference value, the switch opens and the inductor current flows through the load and the diode. The switch is again closed at the arrival of the next clock pulse from a free running clock of period T .

The Boost converter controlled in current mode is modeled by an affine piece-wise hybrid system defined by the same sub-systems given in equation as follows:

$$S_1 : \dot{X}(t) = \begin{bmatrix} \frac{-1}{RC} & 0 \\ 0 & 0 \end{bmatrix} X(t) + \begin{bmatrix} 0 \\ \frac{V_{in}}{L} \end{bmatrix}$$

$$S_2 : \dot{X}(t) = \begin{bmatrix} \frac{-1}{RC} & \frac{1}{C} \\ \frac{-1}{L} & 0 \end{bmatrix} X(t) + \begin{bmatrix} 0 \\ \frac{V_{in}}{L} \end{bmatrix},$$
(10)

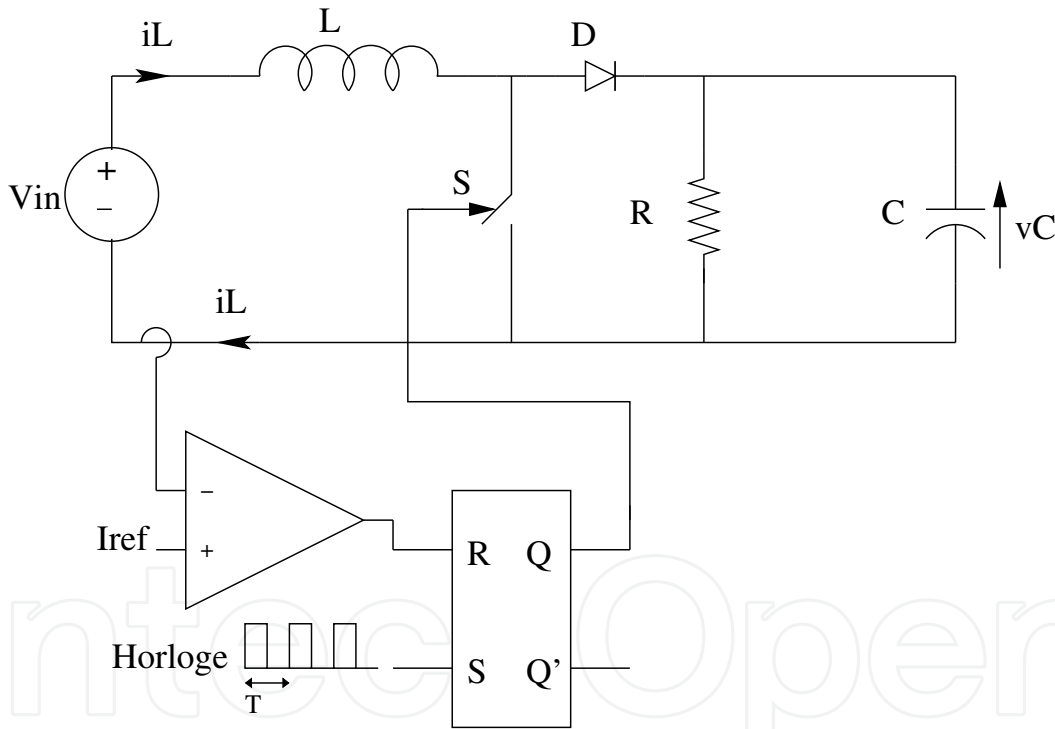


Figure 3. Boost converter controlled in current mode.

In the case of the Boost converter controlled in current mode, there are two types of events:

A state event defined by a fixed border of the set $\mathcal{S}_{E_1E_2}$:

$$\mathcal{S}_{E_1E_2} = \left\{ X \in \mathbb{R}^2 : [0 \ 1] X < I_{ref} \right\}$$
(11)

and another periodic event defined by the dates $t = d\mathcal{P}_{E_iE_j}$, where $d\mathcal{P}_{E_iE_j}$ belongs to the set:

$$\mathcal{P}_{E_2E_1} = \{ t \in \mathbb{R} : t = nT, n \in \mathbb{N} \}$$
(12)

where T is the period of this periodic event. The different simulations are obtained using our planar PWA solver.

Before performing any study of the observed bifurcations in this circuit, a numerical simulation in the parametric plane is needed.

The following program `calcule_balayage_mod.m` is used to obtain the parametric plane:

```

%%-----calcule_balayage_mod.m-----
%
%
clear all;
close all;
%
%% File calculating points to display a figure of parametric plane
% Example of boost
% Save data in...
%
monfich=('data_balais');
% You should specify the path of hybrid_solver_matlab
%
addpath('.\hybrid_solver_matlab');
%
eps=1E-6;
ordre_max=15;
x_eps = 1e-5;
Xmax=100;
nb_trans=500;
nb_inits=1;
ta= 0.5:1.1/200:1.6;
tb= 5:15/200:20;
a=ta(1);
b=tb(1);
%% Definition of BOOST
%Iref changes and noted a
%Vin changes and noted b
%
L=1.5e-3;
T=100e-6;
R=40;
Vin=b;
C=5e-6;
Iref=a;
AE1=[
    -1/R/C  0 ;
    0      0
];

```

```

AE2=[
    -1/R/C  1/C ;
    -1/L    0
];
B1=[0;
    b/L];
B2=B1;
N1 = [0 1];
S1 = '<';
[p1,p2]=racines(AE1);
H=create_hybrid_system('affine');
H=add_state(H,1,'On',AE1,B1);
H=add_state(H,2,'Off',AE2,B2);
H=add_event(H,1,'Iref',N1,a,S1);
H=add_periodic_event(H,1,'Clock',T,0);
H=add_transition(H,1,2,1);
H=add_periodic_transition(H,2,1,1);
%
%% initial state
Xi.t=T/1000;
Xi.E=1;
Xi.Xc=[16.5;0.47];
mape=colormap(ma_color);
na=length(ta)
tic
for ia = 1 : na
    a=ta(ia);
    for ib= 1 : length(tb)
        b=tb(ib);
        %% update of the equation with a new a
        % here only Iref that changes and modifies a border
        B1=[0;
            b/L];
        H=add_state(H,1,'On',AE1,B1);
        H=add_state(H,2,'Off',AE2,B2);
        H=add_event(H,1,'Iref',N1,a,S1);
        H=update_transition(H,1,2,1);
        ordres(ib,ia)=-2;
        for init = 1:nb_inits
            X=Xi;
            or = ordre_max;
            for i = 1 : nb_trans
                [X]=recu(H,X); %1->2;
                [X]=recu(H,X); %2->1;
                if (X.t==Inf)
                    or=-1;

```

```

        break;
    end
    if (max(abs(X.Xc))>Xmax)
        or=0;
        break;
    end
end
end
if (or==ordre_max)
%% check if we have periodic event state E=1
if (X.E ~= 1)
[X]=recu(H,X);
end
it=1;
X0 = X;
tt0=X.t;
while (it<ordre_max) & (or == ordre_max)
[X]=recu(H,X); %1->2;
tt=X.t-tt0;
ii=1;
while (ii*T<tt)
it=it+1;
ii=ii+1;
end
[X]=recu(H,X);
tt0=X.t;
if (max(abs(X.Xc-X0.Xc))<x_eps)
or=it;
break;
else
it = it + 1;
end
end
end
or;
ordres(ib,ia)=max(or,ordres(ib,ia));
end
end
fprintf('About %2.1f %% done, still about %5.0f secondes to be...
%3.0f minutes\n',ia/na*100,toc/ia*(na-ia),toc/ia*(na-ia)/60)
end
temps=toc
save(monfich)
affiche_balayage

```

After calculating the necessary points of the parametric plane saved in the file named `dat_balais`, the next program `affiche_balayage.m` plots the figure given in Fig.4

```

%%%------affiche_balayage_mod.m-----
% Used in general after calcule_balayage

% Charge the saved 2-D bifurcation scan
%if the file was not executed
if (exist('ordres')==1)
    disp('use the points matrix of the workspace');
elseif (exist('data_balais.mat')==2)
    disp('charge the points that are in data_balais.mat');
    load data_balais.mat
else
    disp('There are no points or files of points: try ordres.mat...
        insha ALLAH! It may be long...')
    load ordres.mat
end
%
%% Display the bifurcation scan diagram
da=(ta(2)-ta(1))/2;
db=(tb(2)-tb(1))/2;
colormap(mape)
for ia = 1 : length(ta)
    a=ta(ia);
    for ib= 1 : length(tb)
        b=tb(ib);
        if (ordres(ib,ia)<0)
            %plot(a,b,'.w');
            fill([a-da a-da a+da a+da],[b-db b+db b+db b-db],...
                'w','EdgeColor','none')
        elseif (ordres(ib,ia)==0)
            %plot(a,b,'+w');
            fill([a-da a-da a+da a+da],[b-db b+db b+db b-db],...
                'w','EdgeColor','none')
        else
            %plot(a,b,'s','color',mape(ordres(ib,ia),:),...
                'MarkerFaceColor',mape(ordres(ib,ia),:),'MarkerSize');
            fill([a-da a-da a+da a+da],[b-db b+db b+db b-db],...
                mape(ordres(ib,ia),:),'EdgeColor','none')
        end
    end
    hold on
end
end
colormap(mape)
colorbar %(mape)
colorbar('YTickLabel',...
        {'01','02','03','04','05','06','07',...
        '08','09','10','11','12','13','14','0+'})

```

The figure 4 of the parametric plane allows to emphasize the parameters values for which there exists at least one attractor (fixed point, cycle of order k , strange attractor).

Figure 5 shows a bifurcation diagram (Feigenbaum type) in the plane (I_{ref}, i_L) . However, figure 6 shows the bifurcation diagram in the space plane (I, i_L, v_C) .

To draw these two figures we use programs: `calculer_figurier.m` and `affiche_figurier.m`

```

%%-----calculer_figurier.m-----
%
clear all;
close all;
%
%% Code that calculates then displays the points of a bifurcation
tree
% Boost converter example
% Save the data in ...
monfich=('data_points');
%
addpath('../hybrid_solver_matlab');
%
eps=1E-6; % precision of the solver
nb_trans=400;%400 %Number of iterations to pass the transient phase
ordre_max=100; %100% nombre de points affichés après le transitoire
ta= 0.5:0.0025:1.6; % values of the parameter a to be calculated
%ta= 1.22:0.001:1.4;%0.5:0.001:1.6;
points=zeros(2,length(ta),ordre_max);
% points (x or y, index a, the ordre_max of the last trajectory
points)
a=ta(1);
%
%% Definition of the Boost converter
%Iref is a variable denoted a
L=1.5e-3;
T=100e-6;
R=40;
Vin=10;
C=5e-6;
Iref=a;
AE1=[
    -1/R/C  0 ;
      0     0
];
AE2=[
    -1/R/C  1/C ;
    -1/L    0
];

```

```

B1=[0;
    Vin/L];
B2=B1;
N1 = [0 1];
S1 = '<';
[p1,p2]=racines(AE1)
H=create_hybrid_system('affine');
H=add_state(H,1,'On',AE1,B1);
H=add_state(H,2,'Off',AE2,B2);
H=add_event(H,1,'Iref',N1,a,S1);
H=add_periodic_event(H,2,'Clock',T,0);
H=add_transition(H,1,2,1);
H=add_periodic_transition(H,2,1,2);
%
%% Initial condition
X0.t=T/1000;
X0.E=1;
X0.Xc=[16.4549;0.4648];
%
%% Vary a and memorize the points for the bifurcation tree
na=length(ta);
tic;
for ia = 1 : na
    a=ta(ia);
    if a==1.3
        end
    %% Update of the equation with a new a
    vi=X.Xc;
    cc=ia;
    % Here only Iref varies and the corresponding border is then
    modified
    H=add_event(H,1,'Iref',N1,a,S1);
    H=update_transition(H,1,2,1);
    X=X0;
    %
    %% transient zone
    for i = 1: nb_trans
        [X]=recu(H,X);
        [X]=recu(H,X);
    end
    %% Assure that we are on a periodic event, state E=1
    %
    if (X.E ~= 1)
        [X]=recu(H,X);
    end
    %

```

```

Xin=X.Xc;
tt0=X.t;
it=1;
%% memorize ordremax points issued from the periodic transition
2-> 1
%
while (it<(ordre_max+1))
[X]=recu(H,X);%1->2
tt=X.t-tt0;
ii=1;
while (ii*T<tt)
    points(:,ia,it)=traj_ni(AE1,B1,p1,p2,Xin,ii*T);
    Xin=traj_ni(AE1,B1,p1,p2,Xin,ii*T);
    it=it+1;
    ii=ii+1;
end
%
[X]=recu(H,X);%2->1

points(:,ia,it)=X.Xc;
Xin=X.Xc;
tt0=X.t;
it=it+1;
end
%
fprintf('Approximately %2.1f %% are done, yet approximately %5.0f
seconds...
        that is %3.0f minutes\n',ia/na*100,toc/ia*(na-ia),toc/ia*
        (na-ia)/60)
end
%scan the values of 'a'
temps_ecoule=toc
save(monfich)
%%
cc
vi
affiche_figurier

%%%------affiche_figurier.m-----
%% Charges the file containing the saved points
% if the file "figurier" is not executed
if (exist('points')==1)
    disp('use the points matrix of the workspace');
elseif (exist('data_points.mat')==2)
    disp('charge the points that are in data_points.mat');
    load data_points

```



```

else
    disp('There are no points or files of points: try points.mat insh
        ALLAH!')
    load points
end
%% bifurcation tree depending on the dimensions x then y
% % for dim=1:2
% %
% %     plot(points(dim,1,1));
% %     hold on;
% %     for ia=1:length(ta)
% %         for io=2:ordre_max
% %             plot(ta(ia),points(dim,ia,io));
% %         end
% %     end
% %     xlabel('a');
% %     figure
% % end
% figure
%
%     plot(points(1,1,1));
%     hold on;
%     for ia=1:length(ta)
%         for io=2:ordre_max
%             plot(ta(ia),points(1,ia,io));
%         end
%     end
%     xlabel('Vin(V)');
%     ylabel('vC(V)');
plot(points(2,1,1));
hold on;
for ia=1:length(ta)
    for io=2:ordre_max
        plot(ta(ia),points(2,ia,io));
    end
end
xlabel('Iref(A)');
ylabel('iL(A)');
%% bifurcation tree in 3D
% z = variable parameter denoted a
% x the dimension x
% y the dimension y of the point

plot3(points(1,1,1),points(2,1,1),ta(1));
plot3(ta(1),points(2,1,1),points(1,1,1));
hold on;

```

```

for ia=1:length(ta)
    for io=1:ordre_max
        plot3(ta(ia),points(2,ia,io),points(1,ia,io));
    end
end
xlabel('Iref(A)');
ylabel('iL(A)');
zlabel('vC(V)');

```

In these two figures, the voltage V_{in} is fixed to 10V and the current I_{ref} varies in the interval $[0.5, 1.6]$. We observe a period cascade doubling leading to a chaotic regime, interrupted by a border collision bifurcation at $I_{ref} = 1.23A$ (see figure 7). In this figure, a distinction is given between the attractors of attractive cycle type of the order 1 to 14. Each cycle of order k is associated with one color.

For example, the blue area O1 represents the parameters' values for which there exists an attractive fixed point (fundamental periodic regime). The red area O2 represents the existence of an attractive cycle of order 2. The yellow area O4 represents the existence of an attractive cycle of order 4 and so on until getting the cycles O14 of order $k = 14$. The black area O+ corresponds to parameters values (I_{ref}, V_{in}) for which there exist cycles of order $k \geq 15$ or other types of attractors. In this last area, a chaotic phenomenon could be observed. This bi-dimensional diagram shows some bifurcation curves. In fact, for the rectangle defined by the interval of parameter $V_{in} \in [7, 15]$ and the parameter $I_{ref} \in [0.5, 1.6]$, we observe an area of blue color (existence of attractive fixed point) followed by an area of red color (existence of cycle of order 2), an area of yellow color (existence of cycle of order 4) and another area of black color (existence of cycle of order $k \geq 15$ or another attractor type); this succession of zones corresponds to the existence of period doubling cascade.

This representation of the parametric plane is not enough to establish a bifurcation structure of the hybrid model of the Boost converter, but it is useful for the initialization of programs to draw bifurcation curves.

The simulation results (temporal domain and voltage-current plane (v_C, i_L)) are obtained using the planar PWH solver in the case of the Boost converter controlled in current mode for periods: $1T$ (figure 8) for $I_{ref} = 0.7A$, $2T$ (figure 9) for $I_{ref} = 1A$, $4T$ (figure 10) for $I_{ref} = 1.3A$ and the chaotic regime (figure 11) for $I_{ref} = 1.5A$. For these plots we used the code below by choosing the bifurcation parameter I_{ref} corresponding to each period case.

```

%%
% Define an affine system ii at random and simulate it
% by use of our Matlab toolbox solver
%
% detection of errors manually
%
% Warning
% In the affine case '=' is not supported yet !
addpath('.\hybrid_solver');
%

```

```

L=1.5E-3;
T=100E-6;
R=40;
E=10;
C=5E-6;
%Bifurcation parameter
Iref=0.7; %1;%1.3;%1.5;
% X = [vc ; iL]
%
%System 1 On
A1=[ -1/R/C, 0 ;
      0      0 ];
B1=[0 ; E/L];
N1 = [0 1];
Lim1= Iref;

% System 2 Off
A2=[ -1/R/C, 1/C ;
      -1/L   0 ];
B2=[0 ; E/L];
%
%% Initial condition
X0.t=0;
X0.E=1;
X0.Xc=[16.4549;0.4648];%[13.6097 ;0.3435];
%
%% Boost converter
clear H;
H=create_hybrid_system('Boost Converter');
H=add_state(H,1,'On',A1,B1);
H=add_state(H,2,'Off',A2,B2);
%
H=add_event(H,1,'Iref',N1,Lim1,'<');
H=add_transition(H,1,2,1);
H=add_periodic_event(H,2,'Clock',T,0);
H=add_periodic_transition(H,2,1,2);
Han=H;
%
Xan = hsim(Han,X0,4*T);
%
[XcAn,EAn,tAn]=split_state(Xan);
%
trajplane(Xan,Han)
    figure;
    subplot(211);
    trajplot(Xan,Han,1);
    subplot(212);
    trajplot(Xan,Han,2);

```

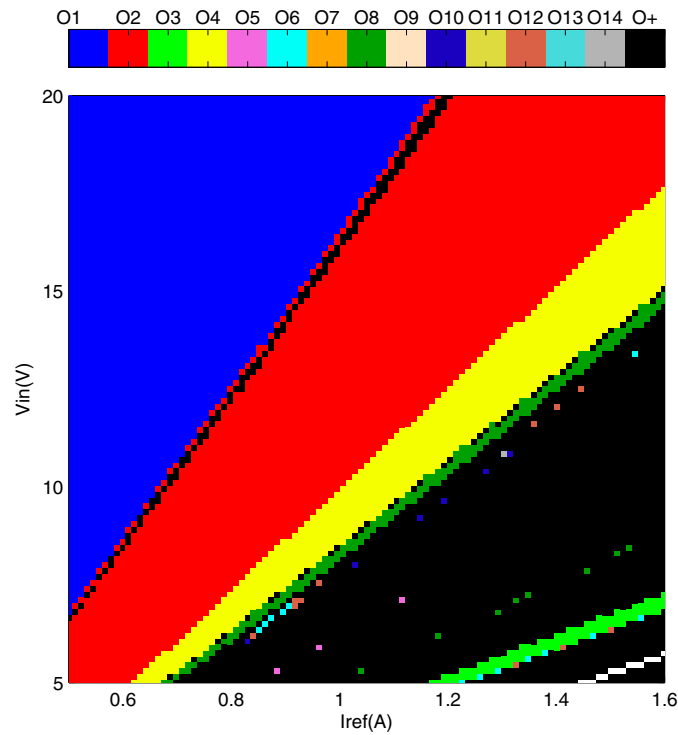


Figure 4. Parametric diagram of the Boost converter in the plane (I_{ref}, V_{in}) for $I_{ref} \in [0.5, 1.6]$ A and $V_{in} \in [5, 20]$ V.

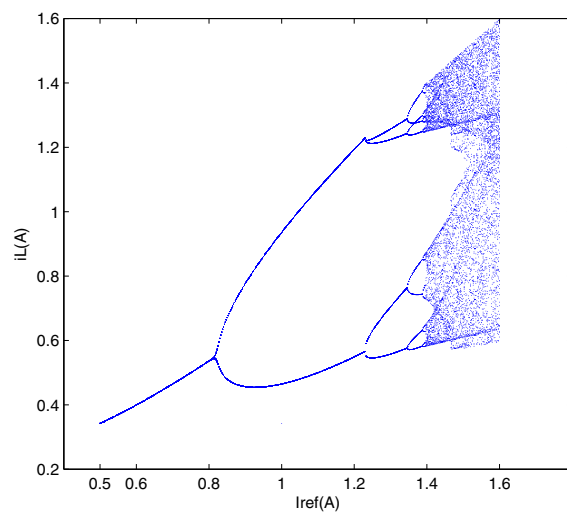


Figure 5. Bifurcation diagram of the Boost in the plane (I_{ref}, i_L) for $I_{ref} \in [0.5, 1.6]$ A and $V_{in} = 10$ V fixed.

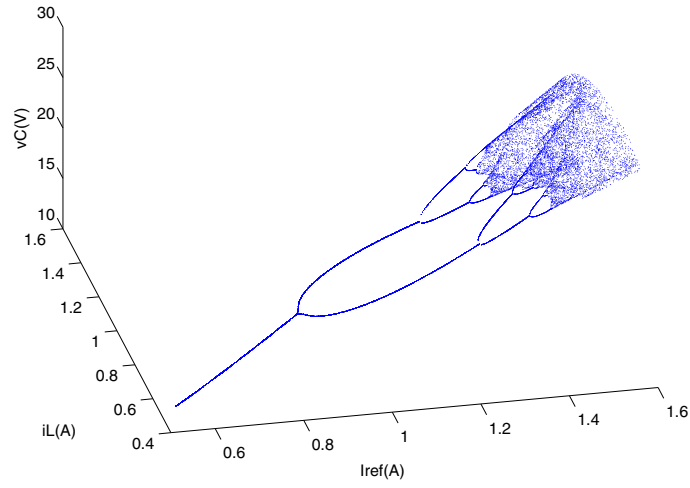


Figure 6. Bifurcation diagram of the Boost in the space (I_{ref}, i_L, v_c) for $I_{ref} \in [0.5, 1.6]A$ and $V_{in} = 10V$ fixed.

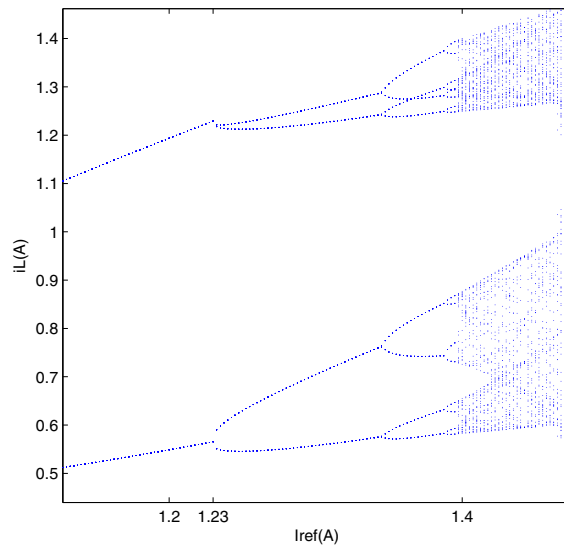
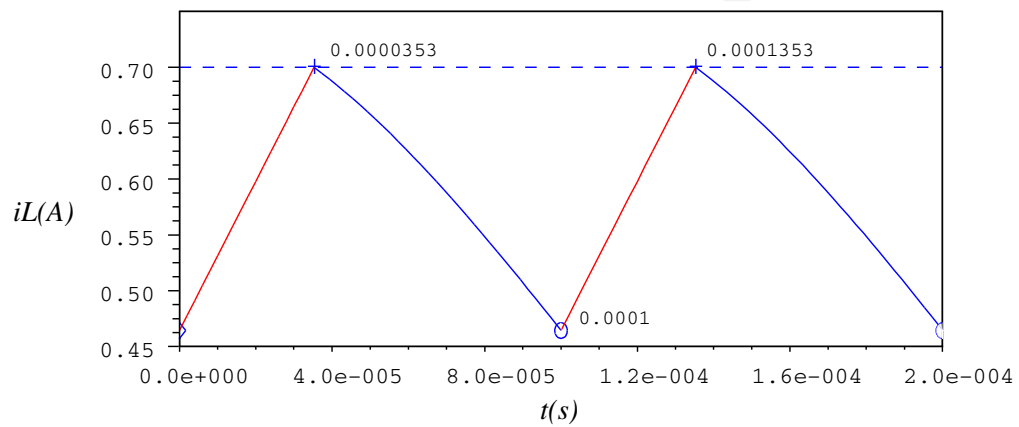
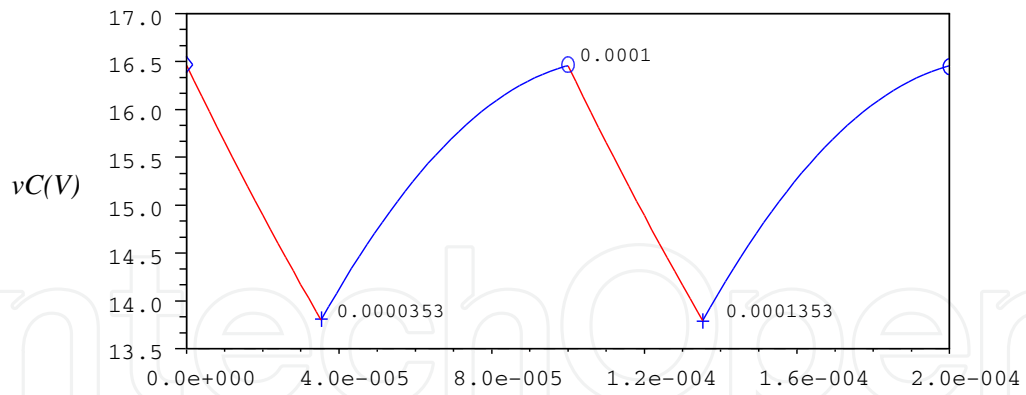
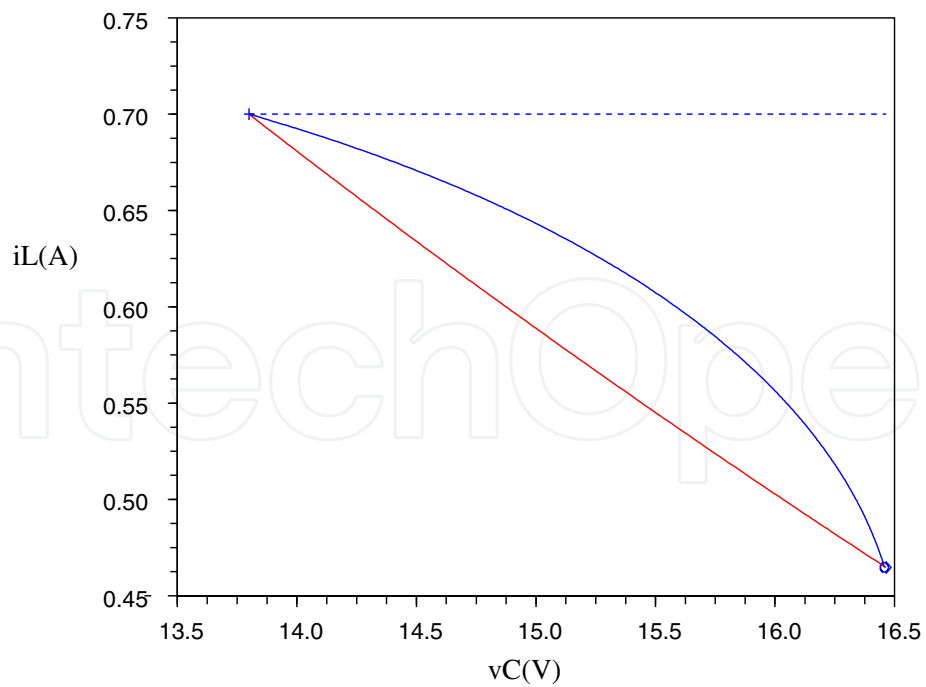


Figure 7. Zoom of figure 5: Border collision for $I_{ref} = 1.23A$.



(a)



(b)

Figure 8. Fundamental periodic regime for $I_{ref} = 0.7A$: (a) (up) temporal waveform of the voltage v_C , (down) temporal waveform of the current i_L ; (b) phase plane (v_C, i_L).

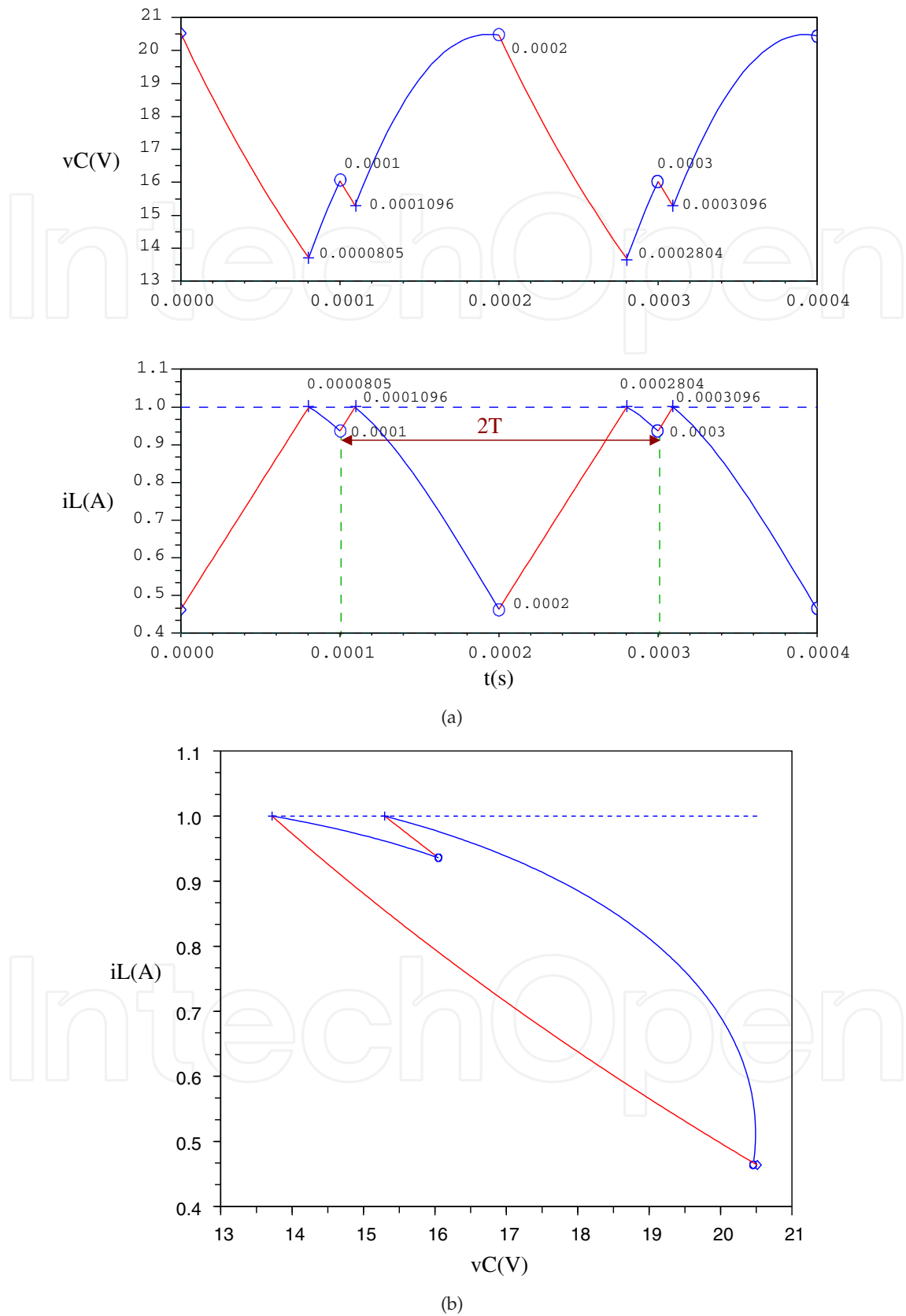
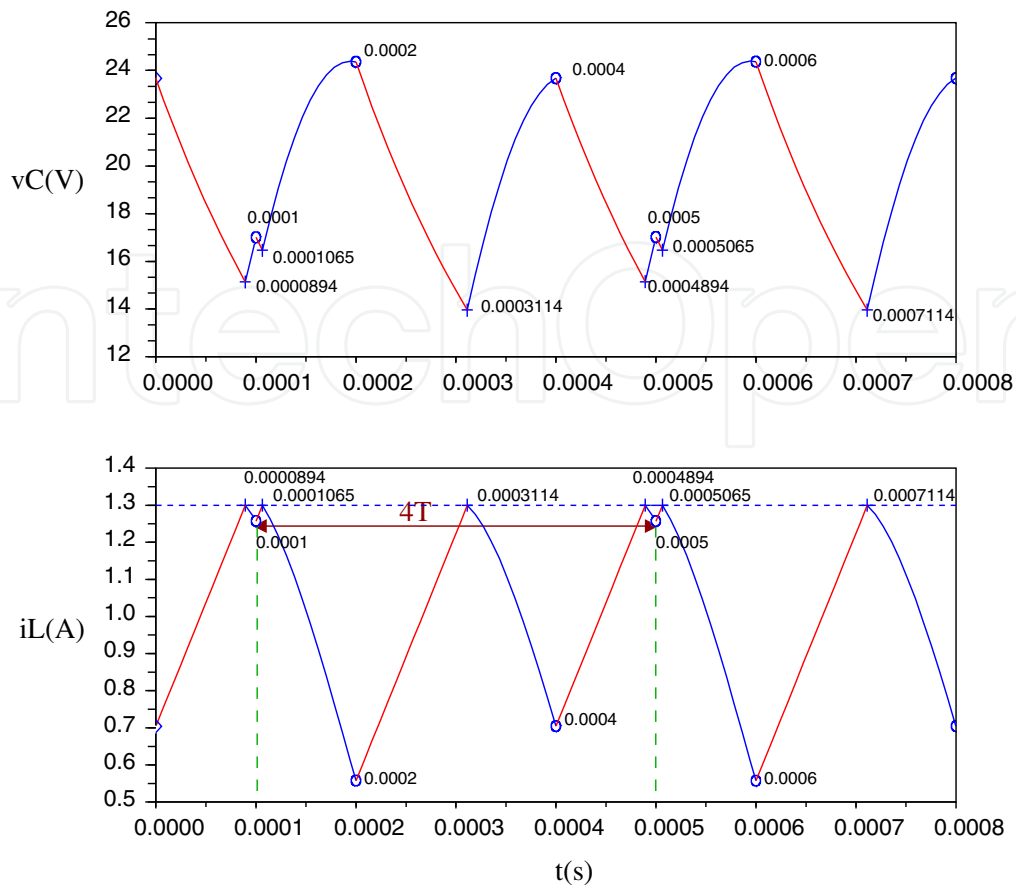
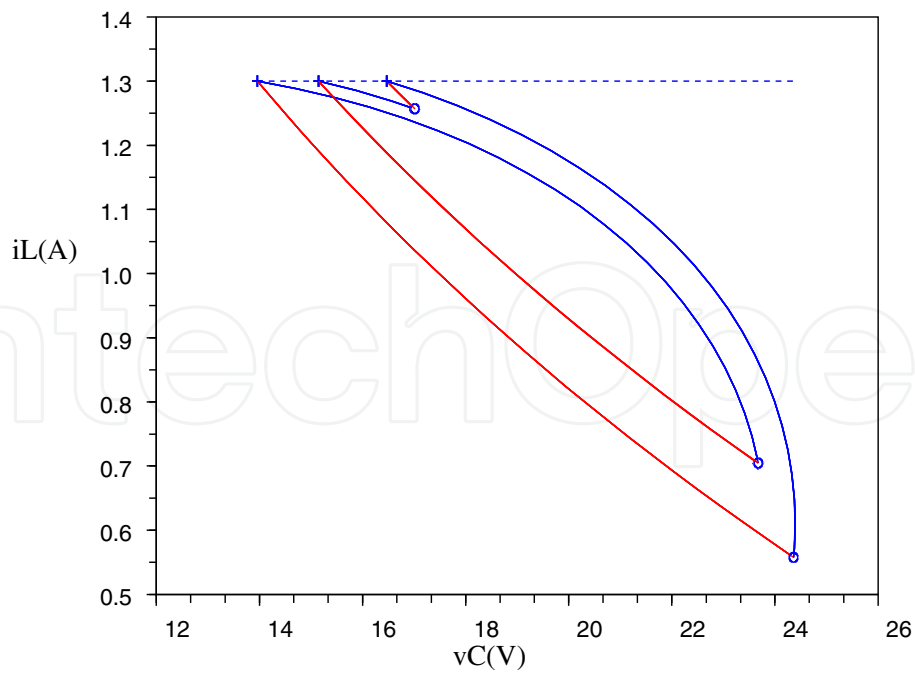


Figure 9. Cycle of order 2 for $I_{ref} = 1A$:(a) (up) temporal waveform of the voltage v_C , (down) temporal waveform of the current i_L ; (b) phase plane (v_C, i_L).

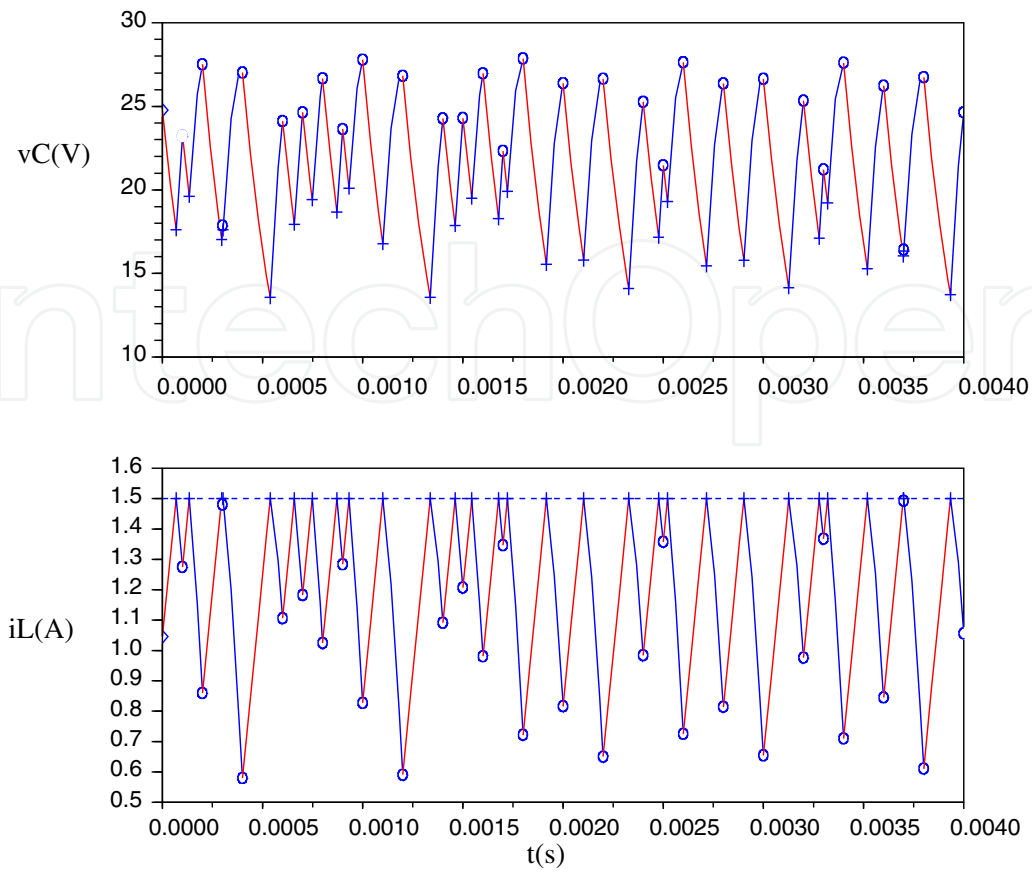


(a)

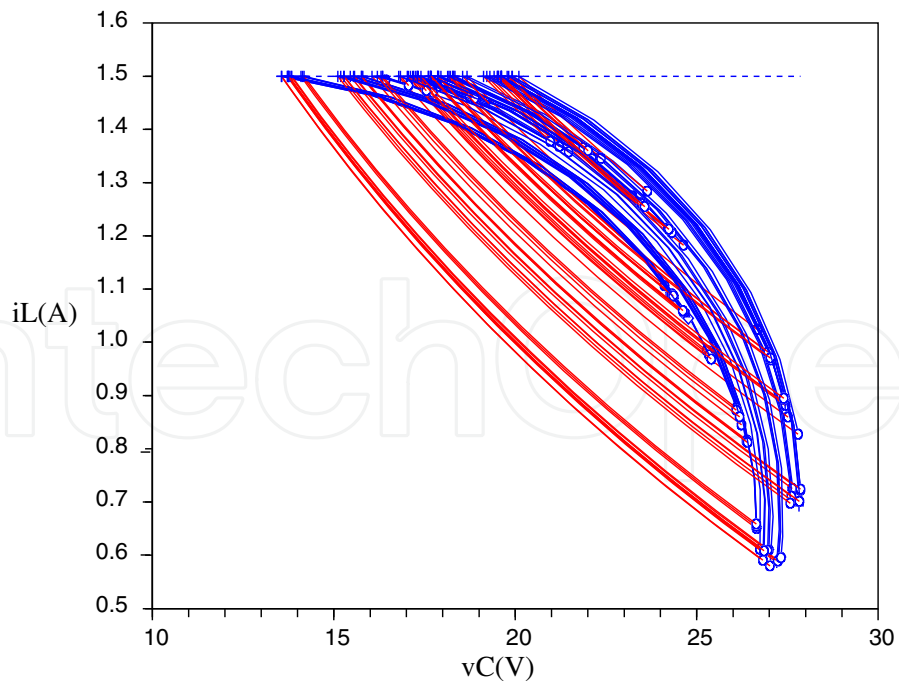


(b)

Figure 10. Cycle of order 4 for $I_{ref} = 1.3$ A: (a) (up) temporal waveform of the voltage v_C , (down) temporal waveform of the current i_L ; (b) phase plane (v_C, i_L).



(a)



(b)

Figure 11. Chaotic regime for $I_{ref} = 1.5A$: (a) (up) temporal waveform of the voltage v_C , (down) temporal waveform of the current i_L ; (b) phase plane (v_C, i_L).

4. Conclusion

In this chapter, we have showed an accurate and fast method to determine events' occurrence for planar piece-wise affine hybrid systems. As a result, we have implemented our algorithm in Matlab toolbox version (free downloadable on <http://felguezar.000space.com/>).

This toolbox has also been completed by analysis tools such as displaying the bifurcation and parametric diagrams. The algorithm takes the advantage of the analytical form that appears in the planar case. Our approach can not be extended to a higher dimension. DC-DC converters like Boost converter are known to be simple switched circuits but very rich in nonlinear dynamics. As application, we have chosen the example of Boost converter controlled in current mode.

Acknowledgments

The authors would like to thank Pascal Acco and Danièle Fournier-Prunaret for crucial discussions on the original version of our work on this subject.

Author details

Fatima El Guezar

ESSI & ERMAGIM, Ibn Zohr University, EST, PO Box 32/S, Agadir, Morocco

Hassane Bouzahir

ESSI & ERMAGIM, Ibn Zohr University, EST, PO Box 32/S, Agadir, Morocco

Faculty of Engineering, AlHosn University, PO Box 38772, Abu Dhabi, United Arab Emirates

5. References

- [1] Acco, P. (December 2003). Etude de la boucle à verrouillage de phase par impulsions de charge: Prise en compte des aspects hybrides. *Ph D thesis*, Institut National des Sciences Appliquées de Toulouse, France.
- [2] Banerjee, S. (2000). Bifurcations in two-dimensional piecewise smooth maps - theory and applications in switching circuits, *IEEE Trans. on Circuits and Systems-I*, Vol.47, pp. 633-647.
- [3] Bouzahir H.; El Guezar, F. & Ueta, T. (2007). On Scicos simulation of a hybrid dynamical system. *Proceedings of the 15th IEEE International Workshop on Nonlinear Dynamics of Electronic Systems*, Tokushima, Japan, pp. 62-65.
- [4] Brockett R. W. & Wood J. R. (1984). Understanding power converter chaotic behavior mechanisms in protective and abnormal modes. *Proceedings of POWERCON 11*, pp. E-14.
- [5] Deane, J. H. B. & Hamill D. C. (1990). Instability, subharmonics, and chaos in power electronic systems. *IEEE Trans. Power Electronics*, Volume 5, pp. 260-268, 1990.
- [6] El Aroudi, A.; Benadero, L.; Toribio, E. & Machiche, S. (2000). Quasiperiodicity and chaos in the DC-DC buck-boost converter. *International Journal of Bifurcation and Chaos*, Vol. 10, pp. 359-371.
- [7] El Aroudi, A. (February 2000). Study of nonlinear phenomena and quasiperiodicity route to chaos in PWM DC/DC converters. *Ph D thesis*, Universitat Politècnica de Catalunya, Spain.

- [8] El Aroudi, A. & Leyva, R. (2001). Quasi-periodic route to chaos in a PWM voltage-controlled DC-DC boost converter. *IEEE Trans. on Circuits and Systems*, Vol. 48, No. 8, pp. 967-978.
- [9] El Aroudi, A.; Debbat, M.; Giral, R.; Olivar, G.; Benadero, L. & Toribio, E. (2005). Bifurcations in DC-DC switching converters: review of methods and applications. *International Journal of Bifurcation and Chaos*, Vol. 5, pp. 1549–1578.
- [10] El Guezar, F. & Bouzahir, H. (2008). Chaotic behavior in a switched dynamical system. *Modelling and Simulation in Engineering*, Vol. 2008, Article ID 798395, 6 pages.
- [11] El Guezar, F.; Acco, P.; Bouzahir, H. & Fournier-Prunaret, D. (2008). Accurate and Fast Event Detection Occurrence in Planar Piecewise Affine Hybrid Systems. *Proceedings of the International Symposium NOLTA (NON Linear Theory and its Applications)*, September 7-10, Budapest-Hungary, pp. 341–344.
- [12] El Guezar, F. (December 2009). Modélisation et simulation des systèmes dynamiques hybrides affines par morceaux. Exemples en électronique de puissance. *Ph D thesis*, Institut National des Sciences Appliquées de Toulouse, France.
- [13] El Guezar, F.; Bouzahir, H. & Fournier-Prunaret, D. (2011). Event Detection Occurrence For Planar Piecewise Affine Hybrid Systems. *Nonlinear Analysis: Hybrid Systems*, Vol. 5, pp. 626–638.
- [14] Girard, A. (2002). Detection of event occurrence in piece-wise linear hybrid systems. *Proceedings of the 4th International Conference on Recent Advances in Soft Computing*, Nottingham, United Kingdom, December, pp. 19–25.
- [15] Girard, A. (September 2004). Analyse algorithmique des systèmes hybrides. *Ph D thesis*, Institut National Polytechnique de Grenoble, France.
- [16] Hamill, D. C. & Jeffries, D. J. (1988). Subharmonics and chaos in a controlled switched-mode power converter. *IEEE Trans. Circuits Syst. I*, Vol. 35, No. 8, pp. 1059–1060.
- [17] Hedayat, C. D.; Hachem, A.; Leduc, Y. & Benbassat, G. (March–April 1997). High-level modeling applied to the second-order charge-pump PLL circuit. *Technical report, Texas Instrument Technical Journal*. Vol. 14, No. 2.
- [18] Mira, C. (1987). Chaotic dynamics. *World scientific Publishing*.
- [19] Tse, C.K. (1994). Chaos from a Buck switching regulator operating in discontinuous mode. *IEEE Transactions on International Journal of Circuit Theory and Application*. Vol. 22, No. 7, pp. 262–278.
- [20] Tse, C.K. (1994). Flip bifurcation and chaos in three-state boost switching regulators. *IEEE Transactions on Circuits and Systems I: Theory and Applications*, Vol. 41, No. 1, pp. 16–23.
- [21] Tse, C.K. (2003). *Complex behavior of switching Power converters*, CRC Press.
- [22] Van Paemel, M. (July 1994). Analysis of a charge pump PLL: a new model. *IEEE Transactions on Communications*, Vol. 42, No. 7, pp. 2490-2498.
- [23] Yuan, G. H.; Banerjee, S.; Ott, E. & Yorke, J. A. (1998). Border-collision bifurcations in the buck converter, *IEEE Trans. on Circuits and Systems-I*, Vol. 45, pp. 707–715.