Chapter 14
Time-Stepping via Complementarity

Vincent Acary

14.1 Introduction

In this chapter, the numerical time integration methods for switched electronic circuits are delineated. The chapter starts with a brief introduction to the simulation of switched electrical circuits. In this introduction, various approaches, among them the hybrid approach, the event-driven approach, the regularisation approach and the time-stepping method via complementarity are recalled. For each of these methods, their advantages and weaknesses are outlined, and the main conclusion is that the time-stepping methods based on complementarity are the most robust and efficient approach to simulate large switched electrical circuits together with a large number of events. A general formulation of dynamical complementarity systems (DCS) over cones is given as

$$\begin{cases} \dot{x}(t) = f(t, x(t), \lambda(t)), \\ y(t) = h(t, x(t), \lambda(t)), \\ K^* \ni y(t) \perp \lambda(t) \in K, \end{cases} \tag{14.1}$$

where K is a non-empty closed convex cone,[1] and K^* its dual cone.[2] This formulation serves as a basis for the design of the time-stepping methods developed in the sequel. The mathematical nature of the expected solution will also be discussed knowing that the nature of the solution has a strong influence on the choice and the consistency of the numerical method. The continuously differentiable (C^1) solution, the absolutely continuous solution and the solution of bounded variation (BV) will

[1] A set K is called a *cone* if for any $x \in K$ and any scalar $a \geq 0$, $ax \in K$.

[2] The *dual cone* K^* of the cone K is the set $K^* = \{y \mid y \cdot x \geq 0 \; \forall x \in K\}$.

V. Acary (✉)
INRIA Rhône-Alpes, Centre de recherche Grenoble, 655 avenue de l'Europe, Inovallée de Montbonnot, 38334 St Ismier Cedex, France
e-mail: Vincent.Acary@inrialpes.fr

F. Vasca, L. Iannelli (eds.), *Dynamics and Control of Switched Electronic Systems*, Advances in Industrial Control,
DOI 10.1007/978-1-4471-2885-4_14, © Springer-Verlag London Limited 2012

be described together with some basic mathematical results concerning existence and uniqueness. Then the formulation of the time-stepping methods via complementarity will be described. For each class of solutions, a suitable method is provided. The mathematical properties (convergence, stability, order) and the practical properties (numerical damping, computational cost) are discussed. Theses properties will be illustrated on simple electrical circuits. For instance, various configurations of the four-diode bridge circuit will be used to exhibit the interest of using such a time-stepping method. We also touch on wrong choices in the design of a time-stepping method which yield numerical instabilities and non-convergence issues. Some extensions of the basic time-stepping method via complementarity are surveyed. Mainly, the inclusion into the normal cone to a convex set will be introduced leading to the direct extensions of the previous time-stepping for relay feedback systems.

Some implementation details will be then explained. Especially, the complementarity solvers that are used at each time-step will be described recalling the main families of available solvers. Some insights on the software implementation will also be given. Finally, numerical applications and examples on more realistic circuits will be considered. We will mainly focus on the architecture of a direct current–direct current (DC–DC) power converter, the buck converter.

14.2 Strengths and Weaknesses of Different Simulation Techniques

The major difficulties that are inherent to the simulation of switched electronic systems are the stiffness and non-smoothness of the dynamical system that models the time evolution of the circuit. To handle such difficulties, four main approaches are listed in the sequel. Their strengths and their weaknesses are briefly described.

The first basic approach considers some regularised models of ideal components to end up with a standard system of smooth differential algebraic equations (DAE). This is the *standard analog modelling and simulation* approach commonly used in the family of analog simulators, e.g. SPICE® (Simulation Program with Integrated Circuit Emphasis)-like simulators. It enables the simulation of switched electronic circuits in the experienced framework of the analog simulation. The regularised model can be the result of a pure mathematical regularisation. Take, as an example, the regularisation of an ideal relay multi-function by using a sigmoid function. A smooth model can also be justified by more accurate physical models. Let us consider for instance the diode component depicted in Fig. 14.1a. A smooth model of the diode based on physical considerations can be invoked by using Shockley's law, an exponential model of the diode illustrated in Fig. 14.1b. Obviously, a better physical model is an advantage from the point of view of the modelling, but there is a price to pay in the numerical practise. The main weaknesses of such a smooth modelling of a switching electrical component are the resulting stiffness of the obtained DAE system and the introduction of small time-scales in the dynamics. The use

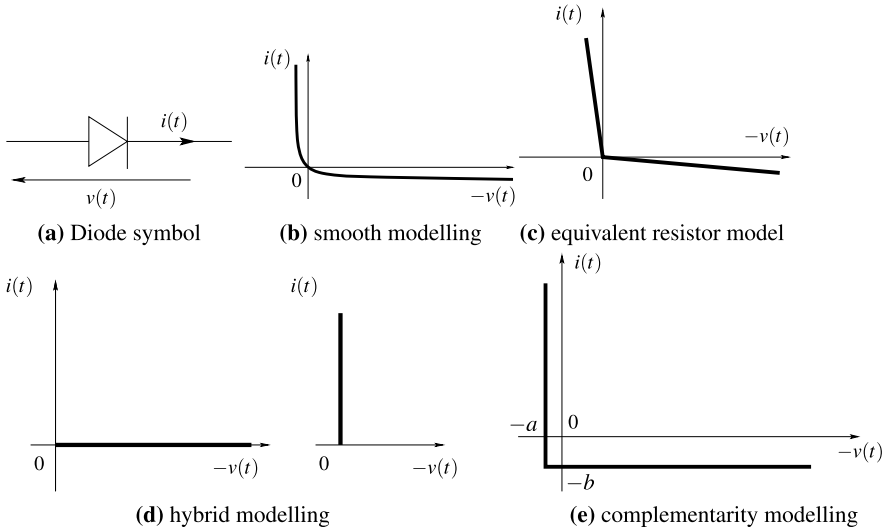


Fig. 14.1 Various modelling of a diode

of implicit time integration methods with a strict control of the time-step length is mandatory to ensure the consistency of the scheme and the convergence of Newton's method. An alternative model is the use an equivalent resistor model as depicted in Fig. 14.1c, which can be naively introduced in the SPICE framework. However, it has been shown in [39] that a standard Newton solver may fail on such a piecewise linear model. Furthermore, the intrinsic non-smoothness of the model destroys the order of accuracy of the integration scheme and disturbs the adaptive time-step strategy. More generally, the simulation of the switches with analog simulators becomes very time consuming and provides the user with very poor results which may contain chattering or even fail as reported in the literature [13, 20, 41, 53].

A substantial part of the stiffness of the regularised or smooth physical models can be removed by using ideal or idealised models. Such models often possess a discrete or digital character in the sense that they are stated with the help of conditional statements. They may be defined by single-valued piecewise-smooth functions or multi-valued mappings that may be naturally described by a set of distinct modes. The introduction of a discrete model with a set of continuous modes calls for the *hybrid modelling and simulation* approach. In Fig. 14.1d, a hybrid model of the diode is described, in which two modes are outlined. For instance, the Modelica description [23] of a diode given by

$$\begin{aligned} &\mathsf{off} = \mathsf{s} < 0 \\ &v = \text{if} \quad \mathsf{off} \quad \text{then} \quad \mathsf{s} \quad \text{else} \quad 0 \\ &i = \text{if} \quad \mathsf{off} \quad \text{then} \quad 0 \quad \text{else} \quad \mathsf{s} \end{aligned} \tag{14.2}$$

generates two distinct modes associated with the values of the boolean variable off. The hybrid simulation is based on a standard time integration of the smooth sys-

tem in the current mode up to when an event is triggered. Then the following mode is selected, and the time integration is continued. The advantage of the hybrid approach is the possibility to take into account quite complicated models; even pure logical behaviours can be treated since digital modelling is implicitly contained in this approach. The main weaknesses are that we need: (a) to accurately trigger the events, (b) to know in advance all the possible modes and (c) to efficiently select the next mode. The knowledge of all the possible modes is not as simple as it seems. The presence of sliding modes for instance can be difficult to detect in a hybrid approach. The selection procedure of the next mode can also be very time-consuming. Let us consider a circuit with n diodes. We can have potentially 2^n modes to test, and this can be prohibitive. Fortunately, some heuristics can be practically implemented based for instance on mixed-integer programming techniques.

The last approaches are based on the *non-smooth modelling and simulation* framework. The two last approaches only differ in their time-integration techniques. For the modelling part, the goal is to design some ideal electrical models by means of formulations extensively used in the mathematical programming theory. To list a few of them, the models that are based on variational inequalities (VI) [24], complementarity problems (CP) [22], inclusions into a normal cone to a convex set [46] and non-smooth and convex optimisation problems [33] are very good candidates. The main difference with the hybrid approach is the intrinsic functional description (possibly multi-valued) of the component rather than a discrete/modal description. In Fig. 14.1e, a complementarity modelling of the diode with possible residual current b and voltage a is depicted. It can be defined by the following complementarity condition, or inclusion into a normal cone, as

$$0 \leq i + b \perp a - v \geq 0 \quad \Longleftrightarrow \quad -(i+b) \in \mathbb{N}_{[-a,+\infty)}(-v). \tag{14.3}$$

The notation $x \perp y$ means that $x^\top y = 0$. Inequalities involving vectors are understood to hold component-wise. The normal cone to K at $x \in K$ is $\mathbb{N}_K(x) = \{z \in \mathbb{R}^n | \langle z, \zeta - x\rangle \leq 0 \text{ for all } \zeta \in K\}$. From the definition of the normal cone it follows that an equivalent definition of the model of the diode as a VI is

$$(i+b)(v+u) \geq 0 \quad \text{for all } u \in [-a, +\infty). \tag{14.4}$$

The first time integration technique is half way between the hybrid approach and the non-smooth approach; we will call it *event-tracking time-stepping schemes* or shortly *event-driven schemes*. As in the hybrid approach, the time integration is performed with standard integration methods for DAE [29] for smooth periods up to an event is triggered. To select the following mode, contrary to the hybrid approach, one of the problems of mathematical programming theory that has been cited above (VI, CP, etc.) is solved. In most favourable cases, these problems can be solved very efficiently in polynomial time. For more details on event-driven schemes, we refer to [1, Chaps. 7 and 8]. When the number of events is too large, a second time-integration technique is more suitable, the so-called *event-capturing time-stepping schemes* or simply *time-stepping schemes*. In this family of methods, the events are

not accurately located, and the time integration is performed without explicitly taking into account the changes of modes. The price to pay is their intrinsic low order, but this is balanced by very good properties of robustness and consistency. In particular, when the number of events is huge, even worse infinite (finite accumulation of events or Zeno-behaviour), or when a sliding-mode motion is encountered, the time-stepping schemes remain really efficient. In the remaining part of the chapter, we will detail the design and the properties of the time-stepping schemes mainly based on complementarity formulations of switching electrical components.

14.3 Solution Concepts and Well-Posedness for Complementarity Systems

The aim of this section is to give a flavour of what types of solutions are expected when some non-smooth components are in the loop. The goal is not to enter into the deepest details of the mathematical nature of solutions, but to give a sufficient insight when we want to deal with the numerical time integration of switched electrical circuits with non-smooth components.

Complementarity modelling of electrical components was introduced in the pioneering works of the Eindhoven school led by W. M. G. van Bokhoven [37, 48–52]. Their aim was to compute the steady-state of piecewise linear circuit by means of the complementarity models. See the book [36] for a good account. The notion of solution was extremely clear since it reduces to the question of the existence and possible uniqueness of solutions as a vector, say x, in a finite-dimensional space, say $\mathbb{R}^n$. When we deal with dynamics, the question of the solution as a function of time $x(t) \in \mathbb{R}^n$ is more difficult since we deal with some functional spaces of functions of time.

The more general systems that we addressed in this chapter may be defined by

$$\begin{cases} \dot{x}(t) = f(t, x(t), \lambda(t)), \\ y(t) = h(t, x(t), \lambda(t)), \\ -y(t) \in \mathbb{N}_X(\lambda(t)), \end{cases} \tag{14.5}$$

where X is a non-empty closed set of $\mathbb{R}^n$. The mappings f and h are assumed to be sufficiently smooth. If the set X is a closed cone denoted by K, the following dynamical complementarity system over cones is obtained:

$$\begin{cases} \dot{x}(t) = f(t, x(t), \lambda(t)), \\ y(t) = h(t, x(t), \lambda(t)), \\ K^* \ni y(t) \perp \lambda(t) \in K. \end{cases} \tag{14.6}$$

If K is the non-negative orthant of $\mathbb{R}^m$, a cone which has a central role in the complementarity theory, a standard dynamical complementarity system is derived:

$$\begin{cases} \dot{x}(t) = f(t, x(t), \lambda(t)), \\ y(t) = h(t, x(t), \lambda(t)), \\ 0 \leq y(t) \perp \lambda(t) \geq 0. \end{cases} \tag{14.7}$$

If the mappings f and h are affine, we get a Linear Complementarity System (LCS).

Well-posedness properties of LCS have been extensively studied in [14, 15, 17, 32]. Some instances of system (14.5) where X is not cone are also very interesting in practise. Indeed, note that

$$-y(t) \in \mathbb{N}_{[-1,1]}\big(\lambda(t)\big) \quad \Longleftrightarrow \quad -\lambda(t) \in \operatorname{sgn}\big(y(t)\big), \tag{14.8}$$

where the function $\operatorname{sgn}(\cdot)$ is the multi-valued signum function defined by

$$\operatorname{sgn}\big(y(t)\big) = \begin{cases} 1 & \text{if } y > 0, \\ -1 & \text{if } y < 0, \\ [-1, 1] & \text{if } y = 0. \end{cases} \tag{14.9}$$

For a vector $y \in \mathbb{R}^m$, $\operatorname{sgn}(y)$ holds component-wise. Let us consider for instance that $X = [-1, 1]^m$ in (14.5). We end up with a dynamical relay system

$$\begin{cases} \dot{x}(t) = f(t, x(t), \lambda(t)), \\ y(t) = h(t, x(t), \lambda(t)), \\ -\lambda(t) \in \operatorname{sgn}(y(t)). \end{cases} \tag{14.10}$$

In the affine case, the well-posedness of linear relay systems has been studied in [2, 45].

In order to say more on the mathematical properties of (14.5), we note that the inclusion in the third line of (14.5) is equivalent to the following VI:

$$y(t)\big(\tau - \lambda(t)\big) \geq 0 \quad \text{for all } \tau \in X, \tag{14.11}$$

that is

$$h\big(t, x(t), \lambda(t)\big)\big(\tau - \lambda(t)\big) \geq 0 \quad \text{for all } \tau \in X. \tag{14.12}$$

Thanks to (14.12), system (14.5) can be recast into the Differential Variational Inequalities (DVI) framework introduced by [44]. As it is noted by the authors, there can be substantial variations of the characteristics of DVIs regarding the existence and regularity of solutions depending mainly on the solution of the VI (14.12). Let

us denote by $\lambda(t) \in \mathrm{SOL}(X, h(t, x(t), \cdot))$ an element of $\mathbb{R}^m$ solution of (14.12). Depending on the mathematical nature of the mapping $(x, t) \mapsto \mathrm{SOL}(X, h(t, x, \cdot))$, various types of solutions to (14.5) are obtained. A brief description is given in the following sections.

14.3.1 Continuously Differentiable Solutions

Let us assume that the mapping $(x, t) \mapsto \mathrm{SOL}(X, h(t, x, \cdot))$ is a single-valued Lipschitz function. The substitution of $\lambda(t)$ into (14.5) yields an ordinary differential equation (ODE) with a Lipschitz right-hand side. The standard theory of ODEs [21] can be then applied, and the solution $x(t)$ is sought as a continuously differentiable function of time, that is a function of class C^1. In the general VI case (14.12), the conditions to obtain a Lipschitz mapping are quite technical and mainly rely on the Lipschitz property and the strong monotonicity of h. In the simplest case of the non-negative orthant $K = \mathbb{R}^m$ and an affine mapping $h(t, x, \lambda) = D\lambda + Cx + q$, the solution $\lambda(t)$ of the linear complementarity system

$$0 \leq \lambda \perp D\lambda + Cx + q \geq 0 \tag{14.13}$$

is unique for all $Cx + q$ if and only if D is a P-Matrix [22], and moreover λ is a Lipschitz function of x.

When the solutions are expected to be of class C^1, system (14.5) is said to be of index *one* [44] or of relative degree *zero* [4]. The notion of index in DVI is very similar to the notion of index in DAE theory. The naive definition of the (differential) index is based on the number of times that the algebraic equations need to be differentiated in order to obtain a solvable ODE in x [29]. The relative degree is a familiar notion in Control Theory [35]. When it is uniform, it is also based on the number of differentiation of the output $y(t)$ to obtain explicitly the input $\lambda(t)$ in function of $x(t)$. For simple linear time invariant systems, the index is given by the relative degree plus one.

14.3.2 Absolutely Continuous Functions

As in the DAE theory, higher index or relative degree systems are obtained if the mapping h is not a one-to-one mapping of λ. For instance, if the Jacobian matrix $\nabla_\lambda^\top h(t, x(t), \lambda(t))$ is singular or worse if the λ does not explicitly appear in the definition of h, indices higher than one are obtained. It is difficult to say more on the nature of solutions without giving more structure to the functions h and f. In [44], some results are given with a function f affine in λ and a function h separable in λ. Without entering into deeper details, the solution $x(t)$ is sought in the sense of Carathéodory: the trajectory $x(t)$ is an absolutely continuous function given by

$$x(t) = x(0) + \int_0^t \dot{x}(s)\,\mathrm{d}s, \tag{14.14}$$

where $\dot{x}(t)$ is the solution of (14.5) ds-almost everywhere. Usually, $\dot{x}(t)$ is sought as a function of Bounded Variation (BV).

If we consider the affine case (14.13) with $D = 0$, we get

$$\begin{cases} \dot{x}(t) = f(t, x(t)) + B\lambda(t), \qquad x(0) = x_0, \\ y(t) = Cx(t) + q, \\ 0 \leq y(t) \perp \lambda(t) \geq 0. \end{cases} \tag{14.15}$$

If CB is a positive definite matrix (relative degree *one*) and $Cx_0 + q \geq 0$ (consistent initial condition), the unique solution of (14.15) is an absolutely continuous function. The time-derivative of the state $\dot{x}(t)$ and $\lambda(t)$ are expected to be, in this case, discontinuous functions of time. Indeed, if the output $y(t)$ reaches the boundary of the feasible domain at time t_*, i.e. $y(t_*) = 0$, the time-derivative $\dot{y}(t)$ needs to jump if $\dot{y}(t_*^-) < 0$.

More generally, if (a) the relative degree of the system is *one*, or the index *two*, and if (b) the initial conditions are consistent, one looks for an absolutely continuous function $x(t)$ and a BV function $\lambda(t)$. If one of the conditions (a) or (b) is not satisfied, a stronger non-smoothness in $x(t)$ and/or $\lambda(t)$ has to be expected. Such a situation is detailed in the next section.

14.3.3 Functions of Bounded Variation

When discontinuities (jumps) are encountered in the solution $x(t)$, we often consider the solutions as functions of Bounded Variation (BV) [43]. A first source of jumps in such systems is the inconsistency of the initial conditions. Let us consider the previous example (14.15) with $Cx_0 + q < 0$. At the initial time, the solution has to jump to a consistent value with respect to the inequality. More generally, if λ does not appear explicitly in the function h in (14.5), the initial condition must meet a consistency condition such that

$$-y(t_0) = -h(t_0, x_0) \in \mathbb{N}_X\big(\lambda(t_0)\big) \tag{14.16}$$

must have at least one solution for $\lambda(t_0)$. Obviously, a first necessary condition is

$$-h(t_0, x_0) \in \text{range}\big(\mathbb{N}_X(\cdot)\big), \tag{14.17}$$

where range$(\mathbb{N}_X(\cdot))$ is the range of $\mathbb{N}_X(\cdot)$. If it is not the case, the state needs to jump to satisfy the inclusion (14.16) at t_0^+. Note that the nature of the set X is also very important in the consistency of the initial conditions. For instance, if $X = [-1, 1]$, then range$(\mathbb{N}_{[-1,1]}(\cdot)) = \mathbb{R}$, and condition (14.16) is trivially solved for any value of x_0. If $X = \mathbb{R}_+$, range$(\mathbb{N}_{\mathbb{R}_+}(\cdot)) = \mathbb{R}_-$, and condition (14.17) is not trivial. A second source of state discontinuities can also be caused by non-smooth external perturbations. If the output is given by $y(t) = Cx(t) + q(t)$ with $q(\cdot)$ a BV function, we may expect to have jumps at any time when the output is moved outside of

range($\mathbb{N}_{[-1,1]}(\cdot)$). The reader will find a good account on state discontinuities, jump rules and conditions to obtain absolutely continuous solutions in the context of linear complementarity systems in [17, 18, 32], in the context of linear switched systems (LSS) in [25, 26, 34] and in [15] for differential inclusions.

If the trajectory $x(t)$ has jumps in time, the time-derivative $\dot{x}(t)$ cannot be defined as the standard derivative with respect to time. The state $x(t)$ is usually assumed to be a right-continuous function of bounded variation (RCBV). For relative degree *one* system, the variable λ has to be replaced by a measure that contains Dirac distributions. In the same vein, the time-derivative of the state $x(t)$ cannot be considered in the usual sense but as a differential measure $\mathrm{d}x$ associated with an RCBV function [15, 40, 42]. The dynamics in problem (14.15) is written in terms of a measure differential equation as

$$\mathrm{d}x = f(t, x(t))\,\mathrm{d}t + B\,\mathrm{d}i, \tag{14.18}$$

where $\mathrm{d}x$ is the differential measure associated with the RCBV function $\dot{x}(t)$, and $\mathrm{d}i$ is also a measure. The absolutely continuous function $\lambda(t)$ is the Radon–Nikodym derivative of $\mathrm{d}i$ with respect to the Lebesgue measure, i.e.

$$\frac{\mathrm{d}i}{\mathrm{d}t} = \lambda(t). \tag{14.19}$$

If the singular part of the differential measure is neglected, a decomposition of the measure can be written as

$$\mathrm{d}i = \lambda(t)\,\mathrm{d}t + \sum_i \sigma_i \delta_{t_i}, \tag{14.20}$$

where δ_{t_i} is the Dirac measure at times of discontinuities t_i, and σ_i is the magnitude. Thanks to (14.20), the differential measure equation (14.18) is decomposed in a smooth dynamics,

$$\dot{x}(t) = f\big(t, x(t)\big) + B\lambda(t) \quad \mathrm{d}t\text{-almost everywhere}, \tag{14.21}$$

and in a jump dynamics at t_i,

$$x\big(t_i^+\big) - x\big(t_i^-\big) = B\sigma_i. \tag{14.22}$$

14.3.4 Summary

The higher the relative degree (or index), the higher the level of non-smoothness. Let us consider the following dynamical system:

$$\begin{cases} x(0) = x_0, \\ x^{(3)}(t) = -1 + \lambda(t), \\ 0 \le x(t) \perp \lambda(t) \ge 0. \end{cases} \tag{14.23}$$

The relative degree of (14.23) is equal to 3. If the initial conditions $x_0 \geq 0$ is consistent at the initial time, the solution is sought as an absolutely continuous function, but λ is no longer a BV function. When the state hits the boundary $x = 0$, λ must contain a derivative of the Dirac distribution $\delta^{(1)}$, and the meaning of the inequality $\lambda(t) \geq 0$ has to be rethought. Without going into deeper details, we refer to [4] for the analysis of higher relative degree systems.

In this chapter, only switched electrical systems with relative degree less than or equal to *one* will be considered. Three kinds of solutions can be therefore expected:

- $x(t)$ is a solution of class C^1, and $\lambda(t)$ is of class C^0 (see conditions in Sect. 14.3.1),
- $x(t)$ is an absolutely continuous function of time, and $\lambda(t)$ is a BV function (see conditions in Sect. 14.3.2),
- $x(t)$ is a BV function of time, and $\lambda(t)$ is the density of a differential measure $\mathrm{d}i$ with respect to the Lebesgue measure (see conditions in Sect. 14.3.3).

It is noteworthy that it can been shown that passive systems have a relative degree less than or equal to *one*. As for smooth passive electrical systems for which the DAE index is no more than *two*, we can conclude on the nature of solutions. Otherwise, each circuit configuration must be cautiously considered.

14.4 Time-Stepping Schemes via Complementarity

In this section, an event-capturing time-stepping scheme, which has been proved to be efficient and robust in many practical situations (see for instance [6, 8]), is proposed for each kind of expected solutions (C^1, absolutely continuous or BV). For convergence proofs, we refer to [2, 11, 16, 30, 31, 40, 42, 44]. In order to avoid some useless complexity about the treatment of standard nonlinearities, the affine instance of (14.5),

$$\begin{cases} \dot{x}(t) = Ax(t) + B\lambda(t) + u(t), \qquad x(0) = x_0, \\ y(t) = Cx(t) + D\lambda(t) + a(t), \\ K^* \ni y(t) \perp \lambda(t) \in K, \end{cases} \tag{14.24}$$

is first considered in this section. The design of the following time-stepping schemes is based on the following principle: a consistent evaluation of the unknown variables and their derivatives according to their smoothness. For instance, time-stepping schemes must not approximate high-order time-derivatives of functions which are

not sufficiently smooth or must not try to point-wisely evaluate distributions or measures.

The following notation is used throughout this part. We denote by $0 = t_0 < t_1 < \cdots < t_k < \cdots < t_N = T$ a finite partition (or a subdivision) of the time interval $[0, T]$ $(T > 0)$. The integer N stands for the number of time intervals in the subdivision. The length of a time step is denoted by $h_k = t_{k+1} - t_k$. For simplicity, in the sequel we consider only a constant time length $h = h_k$ $(0 \le k \le N-1)$. Then $N = T/h$. The approximation of $f(t_k)$, the value of a real function $f(\cdot)$ at the time t_k, is denoted by f_k. For $\theta \in [0, 1]$, the notation $f_{k+\theta}$ stands for $\theta f_{k+1} + (1-\theta) f_k$, and respectively for $\gamma \in [0, 1]$.

14.4.1 Time-Stepping for Continuously Differentiable Solutions

The following time-stepping scheme is used for (14.24) when a solution $x(t)$ of class C^1 with $\lambda(t)$ continuous is expected:

$$\begin{cases} x_{k+1} - x_k = h(Ax_{k+\theta} + u_{k+\theta} + B\lambda_{k+\gamma}), \\ y_{k+\gamma} = Cx_{k+\gamma} + D\lambda_{k+\gamma} + a_{k+\gamma}, \\ K^* \ni y_{k+\gamma} \perp \lambda_{k+\gamma} \in K, \end{cases} \tag{14.25}$$

where $\theta \in [0, 1]$ and $\gamma \in [0, 1]$. The initial value of $\lambda_0 = \lambda(t_0)$ is given by the solution of

$$K^* \ni Cx_0 + D\lambda_0 + a_0 \perp \lambda_0 \in K. \tag{14.26}$$

The discretised system (14.25) amounts to solving at each time-step the following one-step problem:

$$\begin{cases} y_{k+\gamma} = M\lambda_{k+\gamma} + q, \\ K^* \ni y_{k+\gamma} \perp \lambda_{k+\gamma} \in K, \end{cases} \tag{14.27}$$

with

$$\begin{aligned} M &= D + h\gamma C(I - h\theta A)^{-1}B, \\ q &= a_{k+\gamma} + \gamma C(I - h\theta A)^{-1}\big[\big(I + h(1-\theta)A\big)x_k + hu_{k+\theta}\big] + C(1-\gamma)x_k. \end{aligned} \tag{14.28}$$

Example 14.4.1 (four-diode bridge rectifier supplied by an LC oscillator) Let us illustrate the influence of the parameters θ and γ on the behaviour of the numerical scheme (14.25). Let us consider first the configuration of the four-diode bridge in Fig. 14.2. The following values are taken for the physical parameters: $R = 1\,\mathrm{k}\Omega$,

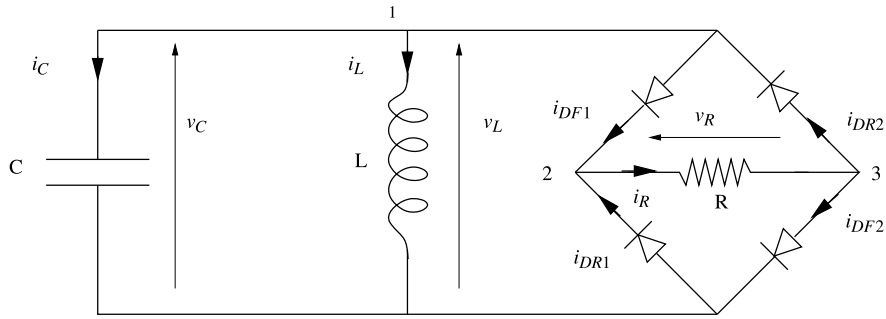


Fig. 14.2 The four-diode bridge rectifier. Inductor–capacitor (LC) oscillator with a load resistor

$L = 10^{-2}$ H, $C = 1\,\mu$F. The resistor inside the bridge is supplied by an LC oscillator. The dynamical equations (14.24) are stated choosing:

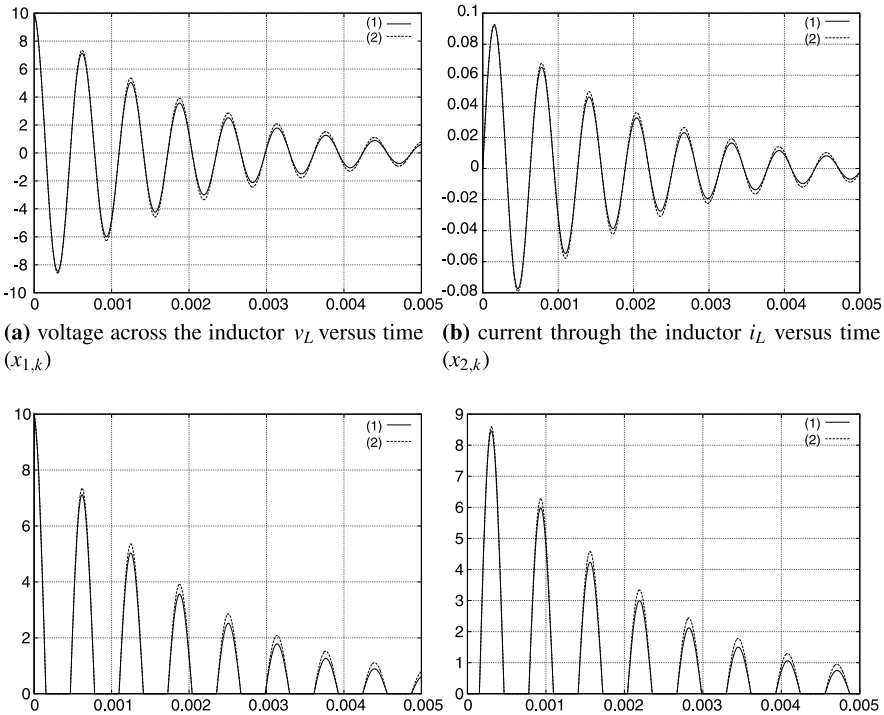
$$x = \begin{pmatrix} v_L \\ i_L \end{pmatrix}, \qquad y = \begin{pmatrix} i_{DR1} \\ i_{DF2} \\ v_2 - v_1 \\ v_1 - v_3 \end{pmatrix}, \qquad \lambda = \begin{pmatrix} v_2 \\ -v_3 \\ i_{DF1} \\ i_{DR2} \end{pmatrix}, \tag{14.29}$$

and with

$$A = \begin{pmatrix} 0 & -1/C \\ 1/L & 0 \end{pmatrix}, \qquad B = \begin{pmatrix} 0 & 0 & -1/C & 1/C \\ 0 & 0 & 0 & 0 \end{pmatrix}, \qquad u = 0,$$
$$C = \begin{pmatrix} 0 & 0 \\ 0 & 0 \\ -1 & 0 \\ 1 & 0 \end{pmatrix}, \qquad D = \begin{pmatrix} 1/R & 1/R & -1 & 0 \\ 1/R & 1/R & 0 & -1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix}, \tag{14.30}$$
$$a = 0, \qquad K = K^* = \mathbb{R}^4_+.$$

Note that the matrix D has full rank and is semi-definite positive. The solution $x(t)$ is of class C^1 since $x \mapsto \mathrm{BSOL}(\mathbb{R}^4_+, D\lambda + Cx + a)$ is a Lipschitz function of x. The results given by the scheme (14.25) are depicted in Fig. 14.3 with time-step $h = 10^{-6}$ and initial condition $x_0 = (10\,0)^\top$. As we can expect, the scheme correctly approximates the trajectory $x(t)$ and the variables $\lambda(t)$ and $y(t)$. We note that the use of a midpoint rule decreases the numerical damping and improves the quality of the solution for a fixed time-step.

As it is illustrated in Fig. 14.3, the scheme does not generate spurious oscillations although switches occur during the time steps. The fact that the matrix D is semi-definite positive yields a very easy solving of the LCP at each time step. The computational effort is of the same order as in solving a comparable linear system.



(a) voltage across the inductor v_L versus time $(x_{1,k})$

(b) current through the inductor i_L versus time $(x_{2,k})$

(c) potential at node 2 (v_2) versus time $(\lambda_{1,k})$

(d) potential at node 3 $(-v_3)$ versus time $(\lambda_{2,k})$

Fig. 14.3 Simulation of the circuit on Fig. 14.2 with the scheme (14.25) for two values of parameters: (1) $\theta = 1$, $\gamma = 1$; (2) $\theta = 1/2$, $\gamma = 1/2$

Finally, in Fig. 14.4, the discrete storage function

$$\mathscr{V}_{k+1} = \frac{1}{2}\left(C\, v_{L,k+1}^2 + L\, i_{L,k+1}^2\right) \tag{14.31}$$

and the discrete cumulative dissipation function

$$\mathscr{D}_{k+1} = h \sum_{j=1}^{k+1} R(i_{R,j})^2 \tag{14.32}$$

are plotted together with the cumulative function $\mathscr{V}_{k+1} + \mathscr{D}_{k+1}$. We remark that the scheme with $\theta = \gamma = 1/2$ is able to reproduce the exact energetic behaviour as in the continuous-time case. This result has been shown in detail in [28].

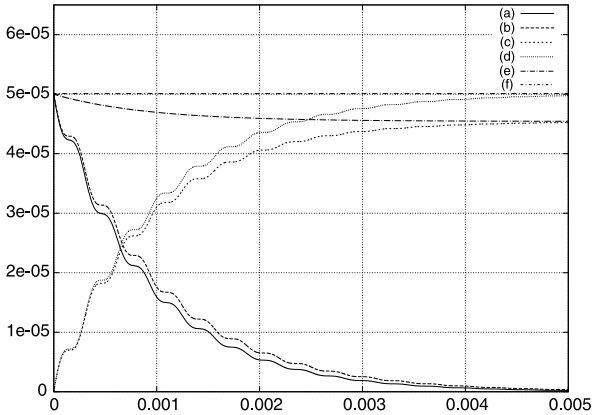


Fig. 14.4 Simulation of the circuit on Fig. 14.2 with the scheme (14.25): (**a**) storage function $\mathscr{V}_{k+1}$ for $\theta = 1$, $\gamma = 1$; (**b**) storage function $\mathscr{V}_{k+1}$ for $\theta = 1/2$, $\gamma = 1/2$; (**c**) dissipation function $\mathscr{D}_{k+1}$ for $\theta = 1$, $\gamma = 1$; (**d**) dissipation function $\mathscr{D}_{k+1}$ for $\theta = 1/2$, $\gamma = 1/2$; (**e**) cumulative function for $\theta = 1$, $\gamma = 1$; (**f**) cumulative function for $\theta = 1/2$, $\gamma = 1/2$

14.4.2 Time-Stepping for Absolutely Continuous Solutions

The following time-stepping scheme is used for (14.24) when an absolutely continuous solution $x(t)$ with $\lambda(t)$ a function of bounded variation is expected:

$$\begin{cases} x_{k+1} - x_k = h(Ax_{k+\theta} + u_{k+\theta} + B\lambda_{k+1}), \\ y_{k+1} = Cx_{k+1} + D\lambda_{k+1} + a_{k+1}, \\ K^* \ni y_{k+1} \perp \lambda_{k+1} \in K, \end{cases} \tag{14.33}$$

with $\theta \in [0, 1]$.

The discretized system (14.33) amounts to solving at each time-step the following one-step problem:

$$\begin{cases} y_{k+1} = M\lambda_{k+1} + q, \\ K^* \ni y_{k+1} \perp \lambda_{k+1} \in K, \end{cases} \tag{14.34}$$

with

$$\begin{aligned} M &= D + hC(I - h\theta A)^{-1}B, \\ q &= a_{k+1} + C(I - h\theta A)^{-1}\big[\big(I + h(1-\theta)A\big)x_k + hu_{k+\theta}\big]. \end{aligned} \tag{14.35}$$

The scheme (14.33) can be viewed as a particular case of the scheme (14.25) using $\gamma = 1$. This small discrepancy between the schemes is nevertheless crucial. In the time-discretization (14.33), the weighted rule with $\gamma < 1$ is not applied to the

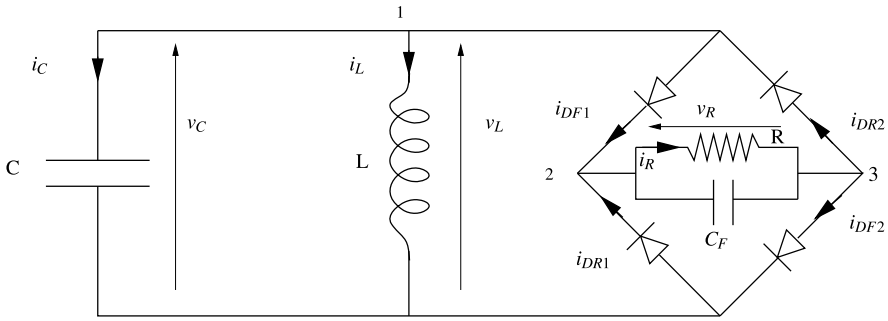


Fig. 14.5 The four-diode bridge rectifier: LC oscillator with a load resistor filtered by a capacitor

variables such as r and λ which are BV. Indeed, higher-order approximations of the BV functions by the mid-point rule, for instance, yield severe instabilities that can disturb the stability of the scheme.

Example 14.4.2 (four-diode bridge rectifier with a capacitive filter supplied by an LC oscillator) Let us consider a second configuration of the four-diode bridge illustrated in Fig. 14.5, written in the form (14.24) by choosing

$$x = \begin{pmatrix} v_L \\ i_L \\ v_R \end{pmatrix}, \qquad y = \begin{pmatrix} v_2 \\ i_{DF2} \\ v_2 - v_1 \\ v_L - v_3 \end{pmatrix}, \qquad \lambda = \begin{pmatrix} i_{DR1} \\ -v_3 \\ i_{DF1} \\ i_{DR2} \end{pmatrix}, \tag{14.36}$$

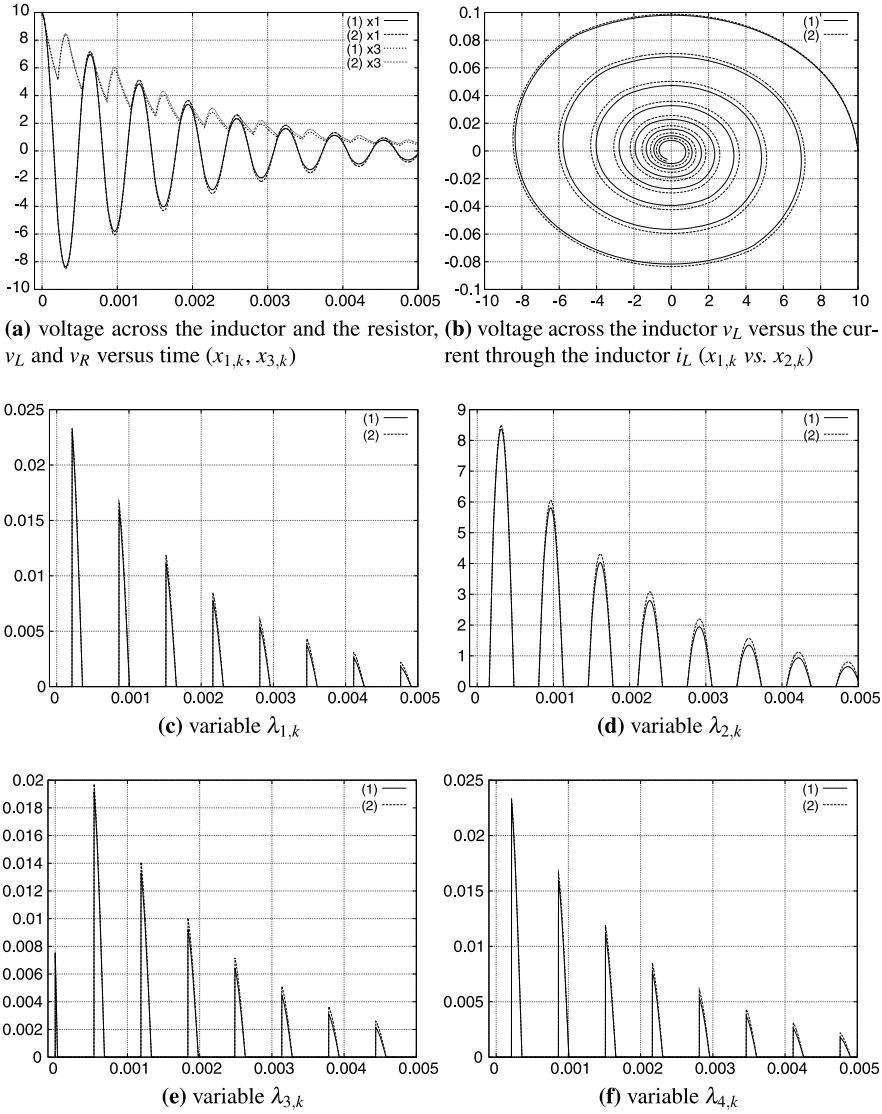
and with

$$A = \begin{pmatrix} 0 & -1/C & 0 \\ 1/L & 0 & 0 \\ 0 & 0 & -1/(RC_F) \end{pmatrix}, \qquad B = \begin{pmatrix} 0 & 0 & -1/C & 1/C \\ 0 & 0 & 0 & 0 \\ 1/C_F & 0 & 1/C_F & 0 \end{pmatrix},$$

$$u = 0,$$

$$C = \begin{pmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ -1 & 0 & 1 \\ 1 & 0 & 0 \end{pmatrix}, \qquad D = \begin{pmatrix} 0 & -1 & 0 & 0 \\ 1 & 0 & 1 & -1 \\ 0 & -1 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix}, \qquad a = 0. \tag{14.37}$$

For this second configuration, the matrix D does not have full rank ($\operatorname{rank}(D) = 2$). The solution $x(t)$ is expected to be a BV solution, and a jump can be encountered if the initial conditions are not consistent. The physical parameters are identical to those in Example 14.4.1 adding $C_F = 300\,\text{pF}$. The initial conditions are chosen as $x_0 = (10\ 0\ 10)^\top$ such that there is no jump at the initial time. In Fig. 14.6, the results of the simulations with the scheme (14.33) are reported for two values of θ. We note that the main qualitative difference between $\theta = 1$ and $\theta = 1/2$ lies in the numerical damping. In Fig. 14.7, the scheme (14.25) dedicated to C^1 solutions



(a) voltage across the inductor and the resistor, v_L and v_R versus time $(x_{1,k}, x_{3,k})$

(b) voltage across the inductor v_L versus the current through the inductor i_L $(x_{1,k}$ *vs.* $x_{2,k})$

(c) variable $\lambda_{1,k}$

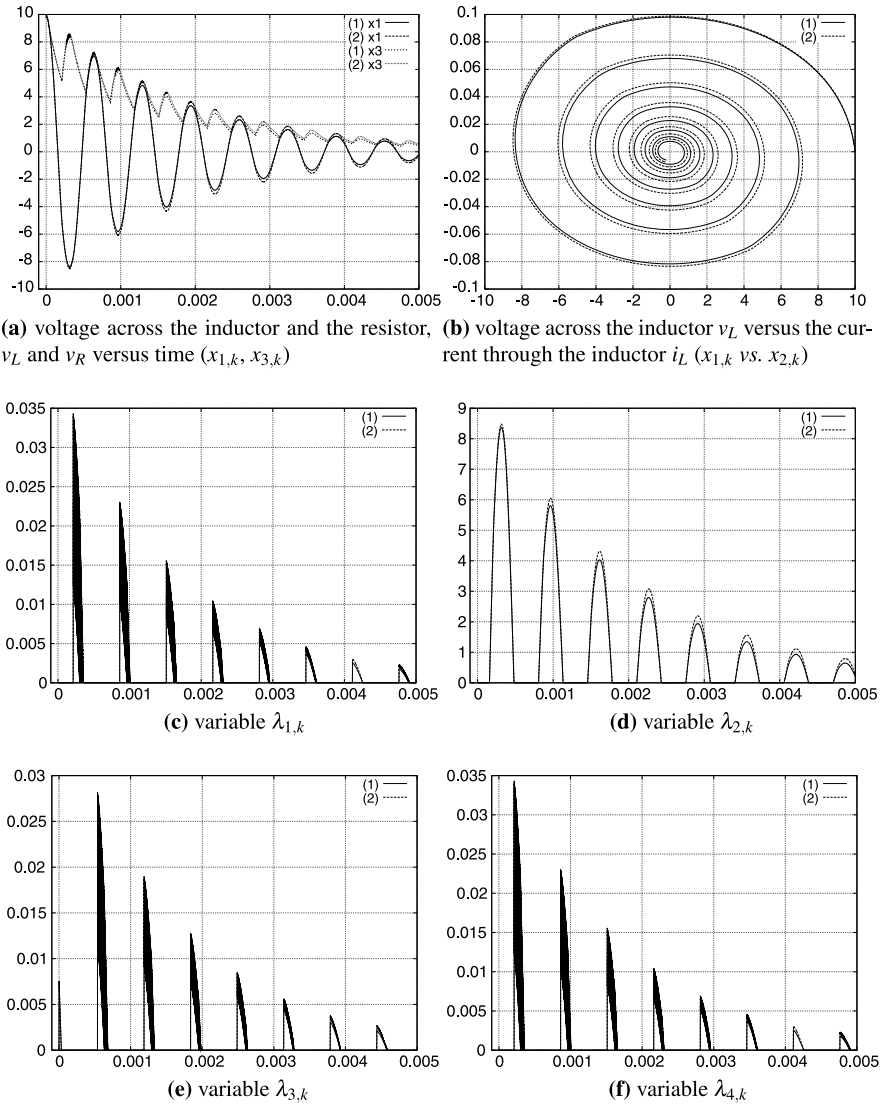
(d) variable $\lambda_{2,k}$

(e) variable $\lambda_{3,k}$

(f) variable $\lambda_{4,k}$

Fig. 14.6 Simulation of the configuration in Fig. 14.5 with the scheme (14.33) for two values of parameters: (1) $\theta = 1$, (2) $\theta = 1/2$

is used. If the trajectory $x(t)$ is very similar to those obtained with the previous scheme, we notice that the scheme (14.25) develops some instabilities on the values λ_k and y_k which approximate BV functions. This illustrates that the mid-point rule cannot consistently approximate λ since it requires more regularity.



(a) voltage across the inductor and the resistor, v_L and v_R versus time $(x_{1,k}, x_{3,k})$

(b) voltage across the inductor v_L versus the current through the inductor i_L $(x_{1,k}$ *vs.* $x_{2,k})$

(c) variable $\lambda_{1,k}$

(d) variable $\lambda_{2,k}$

(e) variable $\lambda_{3,k}$

(f) variable $\lambda_{4,k}$

Fig. 14.7 Simulation of the configuration in Fig. 14.5 with the scheme (14.25) for two values of parameters: *(1)* $\theta = 1$, $\gamma = 1$; *(2)* $\theta = 1/2$, $\gamma = 1/2$. Some instabilities are generated for the approximation of the BV functions

14.4.3 Time-Stepping for Bounded-Variation Solutions

The time discretization of (14.19) has to take into account the nature of the solution to avoid point-wise evaluations of measures that are not mathematically well defined. Only the measure of the time-intervals $(t_k, t_{k+1}]$ must be considered such

that

$$\mathrm{d}x\big((t_k, t_{k+1}]\big) = \int_{t_k}^{t_{k+1}} Ax(t) + u(t)\,\mathrm{d}t + B\,\mathrm{d}i\big((t_k, t_{k+1}]\big). \tag{14.38}$$

By definition of the differential measure, we get

$$\mathrm{d}x\big((t_k, t_{k+1}]\big) = x\big(t_{k+1}^+\big) - x\big(t_k^+\big). \tag{14.39}$$

The measure of the time-interval by $\mathrm{d}i$ is kept as an unknown variable denoted by

$$\sigma_{k+1} = \mathrm{d}i\big((t_k, t_{k+1}]\big). \tag{14.40}$$

Finally, the remaining Lebesgue integral in (14.38) is approximated by an implicit Euler scheme

$$\int_{t_k}^{t_{k+1}} Ax(t) + u(t)\,\mathrm{d}t \approx h(Ax_{k+1} + u_{k+1}). \tag{14.41}$$

The matrix D needs to be at least rank-deficient to expect some jumps in the state. Let us start with the simplest case of $D = 0$.

The following time-stepping scheme is used for (14.24) when a bounded-variation solution $x(t)$ with $\mathrm{d}i$ a measure is expected and $D = 0$:

$$\begin{cases} x_{k+1} - x_k = h(Ax_{k+\theta} + u_{k+\theta}) + B\sigma_{k+1}, \\ y_{k+1} = Cx_{k+1} + a_{k+1}, \\ 0 \in y_{k+1} + N_K(\sigma_{k+1}), \end{cases} \tag{14.42}$$

with $\theta \in [0, 1]$.

The discretised system (14.42) amounts to solving at each time-step the following one-step non-smooth problem:

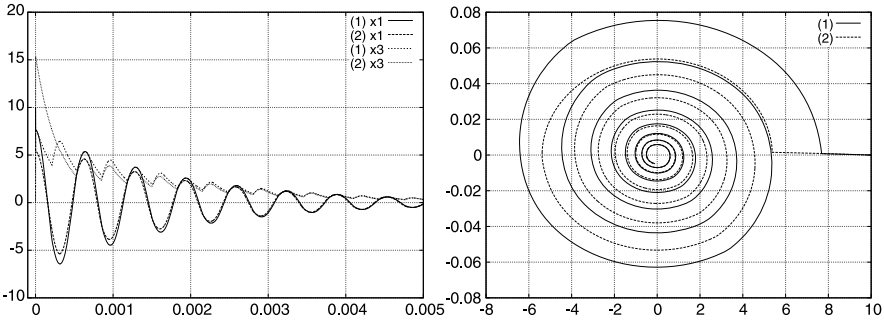
$$\begin{cases} y_{k+1} = M\sigma_{k+1} + q, \\ 0 \in y_{k+1} + N_K(\sigma_{k+1}), \end{cases} \tag{14.43}$$

with

$$\begin{aligned} M &= C(I - h\theta A)^{-1}B, \\ q &= a_{k+1} + C(I - h\theta A)^{-1}\big[\big(I + h(1-\theta)A\big)x_k + hu_{k+\theta}\big]. \end{aligned} \tag{14.44}$$

It is worth noting that the matrix M remains consistent when the time-step h vanishes if CB is assumed to be regular. This is not necessarily the case in (14.28). If $D \neq 0$, the second line of (14.42) is augmented in the following way:

$$y_{k+1} = Cx_{k+1} + a_{k+1} + \frac{1}{h}D\sigma_{k+1}. \tag{14.45}$$



(a) state $x_{1,k}$ and $x_{3,k}$ **(b)** phase portrait $x_{1,k}$ *vs.* $x_{2,k}$

Fig. 14.8 Simulation of the configuration in Fig 14.5 with (*1*) the scheme (14.43) and (*2*) the scheme (14.25) $\theta = 1/2$, $\gamma = 1/2$

More enhanced schemes are described in [8, Sect. 5.2.3], where a decomposition of the matrix D is performed to exhibit non-uniform relative degree.

Example 14.4.3 (Four-diode bridge rectifier with inconsistent initial conditions) The second configuration of the four-diode bridge illustrated in Fig. 14.5 is used in this example, but the initial conditions are chosen as $x_0 = (10\,0\,0)^\top$. The initial conditions are inconsistent, and we expect to have a jump at the initial time. The results are depicted in Fig. 14.8. The curves labelled by (1) are obtained with the scheme (14.42). We note that the jump at the initial time is correctly approximated. Indeed, the scheme approximates the energy-based jump rule presented in [25, 26, 34] and [8, Proposition 2.65]. The curves labelled by (2) are obtained with the scheme (14.25). We note that the trajectory does not seem to be consistently approximated since the magnitude σ is proportional to $1/\gamma$. For a mathematical analysis of the consistency of the scheme (14.42) with respect to a state jump law, we refer to [28].

14.5 Some Possible Extensions

The schemes (14.25), (14.33) and (14.42) proposed in the previous sections can be extended to simulate a more general class of systems defined by (14.5). Let us start with the relay systems in the form

$$\begin{cases} \dot{x}(t) = f(t, x(t), \lambda(t)), \\ y(t) = h(t, x(t), \lambda(t)), \\ -\lambda(t) \in \operatorname{sgn}(y(t)). \end{cases} \tag{14.46}$$

Since $\operatorname{range}(N_{[-1,1]^m}(\cdot)) = \mathbb{R}^m$, there is no question about the consistency of the initial conditions. If the mappings f and h are continuous, only C^1 and absolutely

continuous solutions are expected if the relative degree is less than of equal to one. In the relative degree zero case, the scheme (14.25) is extended for system (14.46) as

$$\begin{cases} x_{k+1} - x_k = hf(t_{k+\theta}, x_{k+\theta}, \lambda_{k+\gamma}), \\ y_{k+\gamma} = h(t_{k+\gamma}, x_{k+\gamma}, \lambda_{k+\gamma}), \\ -\lambda_{k+\gamma} \in \operatorname{sgn}(y_{k+\gamma}), \end{cases} \tag{14.47}$$

where $\theta \in [0, 1]$ and $\gamma \in [0, 1]$. In the relative degree one case, for instance $D = 0$, γ should be chosen equal to 1 to avoid numerical chattering on the sliding surface, $y(t) = 0$. For more details, see [2, 7]. The general case (14.5) is treated in [44]. The authors proposed the following time-stepping scheme:

$$\begin{cases} x_{k+1} - x_k = hf(t_{k+\theta}, x_{k+\theta}, \lambda_{k+\theta}), \\ y_{k+\theta} = h(t_{k+\theta}, x_{k+\theta}, \lambda_{k+\theta}), \\ -y_{k+\theta} \in N_X(\lambda_{k+\theta}), \end{cases} \tag{14.48}$$

which is proved to be convergent for the relative degree zero case under the assumption that the mapping $(x, t) \mapsto \mathrm{SOL}(X, h(t, x, \cdot))$ is a single-valued Lipschitz function.

Let us complete this section by considering systems that are obtained in the simulation practise. If an automatic algorithm is used to generate the equations of a circuit, for instance the Modified Nodal Analysis (MNA) adapted to non-smooth circuits, we get the following very general switched system that models the network:

$$\begin{cases} M(X,t)\dot{X} = F(X,t) + U(t), & \text{DAE} \\ \left.\begin{array}{l} 0 = H(X, \lambda, t), \\ y = G(X, \lambda, t), \end{array}\right] & \begin{array}{l}\text{Input/output relations on} \\ \text{non-smooth components}\end{array} \\ 0 \in \Phi(y, \lambda, t) + \mathbb{N}_K(\lambda), & \text{“Inclusion rule”} \end{cases} \tag{14.49}$$

where $X \in \mathbb{R}^n$ is the state composed of all the potentials, and the currents in inductive, voltage-defined and non-smooth branches. The vectors $y, \lambda \in \mathbb{R}^m$ are the slack variables expressing the non-smooth multi-valued models of the components. The functions $M : \mathbb{R}^n \times \mathbb{R} \to \mathbb{R}^{p\times n}$, $H : \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R} \to \mathbb{R}^{n-p}$, $G : \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R} \to \mathbb{R}^m$ and $\Phi : \mathbb{R}^m \times \mathbb{R}^m \times \mathbb{R} \to \mathbb{R}^m$ are assumed to be continuously differentiable. One recognises two basic ingredients in (14.49): the DAE part that is coupled to the non-smooth electrical devices represented by inclusions into normal cones as those developed in Sect. 14.2. The scheme (14.25) can be extended to (14.49) by

$$\begin{cases} M(X_{k+\theta}, t_{k+\theta})(X_{k+1} - X_k) = h(F(X_{k+\theta}, t_{k+\theta}) + U_{k+\theta}), \\ 0 = H(X_{k+\gamma}, \lambda_{k+\gamma}, t_{k+\gamma}), \\ y = G(X_{k+\gamma}, \lambda_{k+\gamma}, t_{k+\gamma}), \\ 0 \in \Phi(y_{k+\gamma}, \lambda_{k+\gamma}, t_{k+\gamma}) + \mathbb{N}_K(\lambda_{k+\gamma}). \end{cases} \tag{14.50}$$

In general, the matrix M in not invertible, and it is very difficult to obtain a separable form of H and G. Furthermore, guessing the relative degree of system (14.49) and then the degree of smoothness of the solution is a hard task for a given circuit. Without any further information on the smoothness of the solution, we first use the scheme (14.50) with $\theta = 1$ and $\gamma = 1$. In order to improve this basic scheme, further knowledge is needed. For instance, if the variable X is supposed to be absolutely continuous, a θ-method with $\theta \neq 1$ can be used. In the same vein, if the system encounters jumps, the dynamics and the inclusion in (14.49) have to be written in terms of measures, and the numerical integration should be performed with impulses. In practise, the scheme (14.50) with $\theta = \gamma = 1$ is employed as a compromise between the mid-point rules developed in the previous sections and the scheme with impulses. After a first simulation, the numerical integration may be often improved with a more dedicated scheme.

14.6 Implementation Issues

In this section, some technical aspects on the implementation issues are given. They are at least as important as the theoretical aspects developed before. We will mainly discussed the algorithm for the automatic equation generation, the methods for solving the one-step problem and the software development.

14.6.1 Automatic Equation Generation for Switched Systems

An automatic circuit equation generation system extending the MNA has been developed at the INRIA (Institut National de Recherche en Informatique et en Automatique) yielding the following formulation:

$$\begin{cases} \left.\begin{aligned} \dot{x} &= f_1(x,z,t) + U(t), \\ 0 &= f_2(x,z,t), \end{aligned}\right] & \text{Semi-Explicit DAE} \\ \left.\begin{aligned} 0 &= h(x,z,\lambda,t), \\ y &= g(x,z,\lambda,t), \end{aligned}\right] & \begin{aligned}&\text{Input/output relations on} \\ &\text{non-smooth components}\end{aligned} \\ 0 \in \Phi(y,\lambda,t) + \mathbb{N}_K(\lambda), \quad] & \text{“Inclusion rule”} \end{cases} \tag{14.51}$$

where $x \in \mathbb{R}^k$ corresponds to the currents in the inductive branches and the voltages in the capacitive branches, $z \in \mathbb{R}^l$ collects all the node potentials, the currents in the voltage-defined and non-smooth branches, and the currents in a subset of the capacitive branches. The choice and the construction of the latter subset of branches is described in details in the patent [5] and [8, Chap. 6]. To avoid troubles related to the index in DAE and the presence of capacitive loops and inductive cutsets, we exploit similar techniques as in [9, 10].

14.6.2 Solving the One-Step Problem

The problem (14.50) is a VI written in the form of an inclusion into a normal cone to a convex set. The choice of the numerical solver for (14.50) depends mainly on the structure of the convex set K. Indeed, from a very general convex set K to a particular choice of K, the numerical solvers range from the numerical methods for VI to nonlinear equations, passing through various complementarity problem solvers. The convergence and numerical efficiency are improved in proportion as the structure of K becomes simpler. In the sequel, majors choices of K will be given leading to various classes of well-known problems in mathematical programming theory. We refer to [24] for a thorough presentation of available numerical solvers and to [1, Chap. 12] for a comprehensive summary of numerical algorithms.

For the numerical purposes, let us rewrite problem (14.50) as a global inclusion

$$0 \in \mathsf{F}(\zeta) + \mathbb{N}_C(\zeta), \tag{14.52}$$

where the variable $\zeta = [X_{k+1}^\top, y_{k+1}^\top, \lambda_{k+1}^\top]^\top \in \mathbb{R}^{n+2m}$, and the function $\mathsf{F} : \mathbb{R}^{n+2n} \to \mathbb{R}^{n+2m}$ is defined by

$$\mathsf{F}(\zeta) = \begin{bmatrix} M(\theta X_{k+\theta}, t_{k+\theta})(X_{k+1} - X_k) - hF(\theta X_{k+\theta}, t_{k+\theta}) - hU(t_{k+\theta}) \\ H(X_{k+\gamma}, \lambda_{k+\gamma}, t_{k+\gamma}) \\ G(X_{k+\gamma}, \lambda_{k+\gamma}, t_{k+\gamma}) - y_{k+\gamma} \\ \Phi(y_{k+\gamma}, \lambda_{k+\gamma}, t_{k+\gamma}) \end{bmatrix}. \tag{14.53}$$

The normal cone $\mathbb{N}_C$ is the normal cone to the following convex set:

$$C = \mathbb{R}^n \times \mathbb{R}^m \times K \subset \mathbb{R}^{n+2m}. \tag{14.54}$$

In practise, the convex set is finitely represented by

$$K = \left\{\lambda \in \mathbb{R}^m \mid h(\lambda) = 0, g(\lambda) \geq 0\right\}, \tag{14.55}$$

where the functions $h : \mathbb{R}^m \to \mathbb{R}^m$, $g : \mathbb{R}^m \to \mathbb{R}^m$ are assumed to be smooth with non-vanishing Jacobians. More precisely, we assume that the following constraints qualification holds:

$$\forall \lambda \in K, \exists d \in \mathbb{R}^m \quad \text{such that} \quad \begin{cases} \nabla^\top h_i(\lambda) d < 0, & i = 1, \ldots, m, \\ \nabla^\top g_j(\lambda) d < 0, & j \in \mathscr{I}(\lambda), \end{cases} \tag{14.56}$$

where $\mathscr{I}(\lambda)$ is the set of active constraints at λ, that is

$$\mathscr{I}(\lambda) = \left\{j \in 1, \ldots, m,\ g_j(\lambda) = 0\right\}. \tag{14.57}$$

In this case, general algorithms for VI can be used. To cite a few, the minimisation of the so-called regularised gap function [27, 54, 55] or generalised Newton methods [24, Chaps. 7 and 8] can be used. If $F(\cdot)$ is affine (possibly after the linearization Newton step) and the functions $h(\cdot)$ and $g(\cdot)$ are also affine, the VI is

said to be an affine VI for which the standard pivoting algorithms for LCP [22] are extended in [19].

Let us consider the case that K is a generalised box, that is

$$K = \left\{\lambda \in \mathbb{R}^m \mid a_i \leq \lambda_i \leq b_i, a_i \in \bar{\mathbb{R}}, b_i \in \bar{\mathbb{R}}, i = 1 \ldots m\right\}, \tag{14.58}$$

with $\bar{\mathbb{R}} = \{\mathbb{R} \cup \{+\infty, -\infty\}\}$. In this case, problem (14.52)–(14.54) can be recast into a Mixed Complementarity Problem (MCP) by defining $p = n + m + m + m$ and the bounds l, u as $l = [\,0_n\ 0_m\ 0_m\ a\,]^\top$ and $u = [\,0_n\ 0_m\ 0_m\ b\,]^\top$.

The MCP can be solved by a large family of solvers based on Newton-type methods and interior-points techniques. In contrast to the interior-point methods, it is not difficult to find comparisons of numerical methods based on Newton's method for solving MCPs. We refer to [12] for an impressive comparison of the major classes of algorithms for solving MCPs. If $\mathsf{F}(\cdot)$ is affine, the MLCP is equivalent to a box-constrained affine VI. For this problem, the standard pivoting algorithm such Lemke's method is extended in [47]. A special case of a generalised box is the positive orthant of $\mathbb{R}^m$, that is $K = \mathbb{R}^m_+$. Standard theory and most of the numerical algorithms for LCPs apply in this MCLP case.

14.6.3 Open-Source Software Platform

Finally, some insights are given on the software implementation of the methods. A netlist is a circuit textual description used by many SPICE-like simulators. From a netlist, the automatic generator builds all the components defined in (14.51). The open-source SICONOS/KERNEL [3] library performs the time-discretisation following the Moreau time-stepping scheme (14.50) and formulates at each time-step one instance of the inclusion problem (14.52)–(14.54). The numerical algorithms for the latter problem are in the open-source SICONOS/NUMERICS library. The output of the simulation is a file containing the potential and current values in the SPICE format.

The implementation is object-oriented and mainly in C++. For each electrical component, group of equations and inclusions in (14.50), a corresponding instance of a class is built. The system is updated in memory at each iteration by the stamp method of each component. In the linear case, these methods are called only once; in the nonlinear case, they may be called at any time to update the system. The open-source platform SICONOS is under General Public License and can be freely used. The equation generator is under private license and can be obtained freely on demand for an academic use. Note that the actual implementation of the software does not exploit the inherent sparsity of the system. This is mainly due to development effort restriction. It is clear that substantial gain may be expected by using standard sparse library.

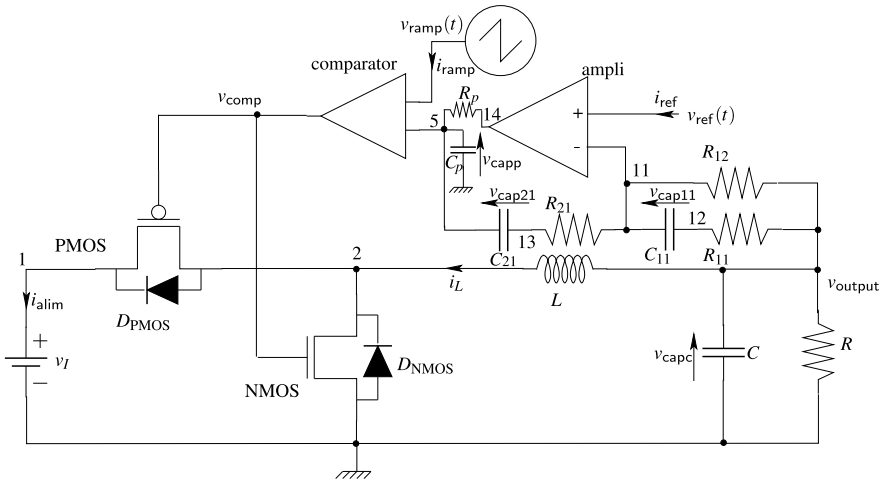


Fig. 14.9 Buck converter

14.7 Applications and Examples on the Buck Converter

We conclude the numerical experiments with the less academic example of a buck converter described on Fig. 14.9. The components are modelled with either linear, or piecewise-smooth, or set-valued relations yielding a non-smooth dynamical system of the linear time-invariant complementarity systems class. The features of the models are given thereafter.

The non-smooth DAE has been generated using the automatic circuit equation formulation. It leads to a dynamical system described in (14.51) composed of 5 dynamical equations and 12 algebraic equations. The unknowns are

$$x = (v_{\text{capp}} \quad v_{\text{cap21}} \quad v_{\text{cap11}} \quad v_{\text{capc}} \quad i_L)^\top$$

and

$$z = (v_1 \quad v_2 \quad v_{\text{comp}} \quad v_{\text{ramp}} \quad v_{\text{ref}} \quad v_{11} \quad v_{14} \quad i_{\text{ref}} \quad i_{\text{ramp}} \quad i_{\text{alim}} \quad i_{\text{ampli}})^\top,$$

where the unknowns are depicted on Fig. 14.9.

14.7.1 Nonsmooth MOSFET Transistors

Following [36], let us consider the Sah model of the nMOSFET (n metal–oxide–semiconductor field–effect transistor) static characteristic

$$i_{\text{ds}} = \frac{\mathsf{K}}{2}\big(f(v_g - v_s - v_t) - f(v_g - v_d - v_t)\big) \tag{14.59}$$

with K is the transconductance of the transistor. The voltage v_t is the threshold voltage depending on the technology. The notation for the currents and the potentials is as follows: the subscript g stands for the gate, d for the drain, and s for the source. The function $f : \mathbb{R} \to \mathbb{R}$ in (14.59) is defined as

$$f(x) = \begin{cases} 0 & \text{if } x < 0, \\ x^2 & \text{if } x \geq 0. \end{cases} \tag{14.60}$$

14.7.1.1 A Piecewise-Linear Model

The piecewise and quadratic nature of the function (14.60) is approximated by the following $(s+2)$-segment piecewise-linear function [36]:

$$f_{\mathsf{pwl}}(x) = \alpha_i x + \beta_i \quad \text{for } a_i \leq x \leq a_{i+1},\ i = -1, \ldots, s+1, \tag{14.61}$$

with $a_{-1} = -\infty$ and $a_{s+1} = +\infty$. The complete model of the piecewise-linear nMOS transistor with $s+2$ segments in (14.61) can be recast under the following mixed linear complementarity form:

$$\begin{aligned} y(t) &= \begin{pmatrix} \underbrace{\begin{matrix} 0 & \ldots & 0 \\ -b & \ldots & -b \end{matrix}}_{\times s+1} & \underbrace{\begin{matrix} -b & \ldots & -b \\ 0 & \ldots & 0 \end{matrix}}_{\times s+1} \end{pmatrix}^\top v(t) \\ &\quad + \lambda(t) + \begin{pmatrix} h_1 & \ldots & h_{s-1} & h_1 & \ldots & h_{s-1} \end{pmatrix}^\top, \\ 0 &= \mathsf{I}_3\, i(t) + \begin{pmatrix} -c_1 & \ldots & -c_{s-1} & c_1 & \ldots c_{s-1} \\ 0 & 0 & 0 & 0 & 0 \\ c_1 & \ldots & c_{s-1} & -c_1 & \ldots -c_{s-1} \end{pmatrix} \lambda(t), \\ 0 &\leq y(t) \perp \lambda(t) \geq 0, \\ v(t) &= \begin{pmatrix} v_{\mathrm{gd}}(t) = v_g(t) - v_d(t) \\ v_{\mathrm{gs}}(t) = v_g(t) - v_s(t) \end{pmatrix}, \quad \text{and} \quad i(t) = \begin{pmatrix} i_d(t) \\ i_g(t) \\ i_s(t) \end{pmatrix}. \end{aligned} \tag{14.62}$$

The parameters are given as follows: $b = \mathsf{K}/2$, $h_i = b(v_t + a_i),\ i = 1, \ldots, s$. The values c_i are computed from the linear approximation in (14.61). Using some basic convex analysis, one obtains the compact formulation of (14.62):

$$\begin{cases} -y(t) \in \mathbb{N}_K(\lambda(t)), \\ y(t) = Bu(t) + \lambda(t) + h(t), \\ 0 = i(t) + C\lambda(t), \end{cases} \tag{14.63}$$

with $K = (\mathbb{R}_+)^{2(s+1)}$. The pMOS transistor is represented in the same way, changing the values of h_i, $i(t)$ to $-i(t)$ and b to $-b$. For more details on the construction and the calibration of such a model, we refer to [36].

14.7.1.2 A Piecewise-Nonlinear Model

The model (14.59) can be modelled using the piecewise-nonlinear model (14.64). This leads to a nonlinear MCP,

$$\begin{cases} i_{\mathrm{ds}} = \frac{\mathsf{K}}{2}(\lambda_4(v_g - v_s - v_t)^2 - \lambda_2(v_g - v_d - v_t)^2), \\ y_1 = 1 - \lambda_2, \qquad y_2 = v_t - v_g + v_d + \lambda_1, \\ y_3 = 1 - \lambda_4, \qquad y_4 = v_t - v_g + v_s + \lambda_3, \end{cases} \tag{14.64}$$

and $0 \leq y \perp \lambda \geq 0$.

14.7.2 Parameters and Simulation Models

Power MOSFETs pMOS/nMOS are described as an assembly of a piecewise-linear current source $i_{\mathrm{ds}} = f(v_{\mathrm{gs}}, v_{\mathrm{ds}})$ and the intrinsic diode (DpMOS and DnMOS) with an ideal characteristic. The capacitors were not taken into account. The diodes residual voltage is 0.8 V. The MOSFETs trans-conductance K was set to 10 A/V^2, and their threshold voltage to respectively $v_t = -2$ V for the pMOS and $v_t = 2$ V for the nMOS. One can notice that the sum of their absolute values largely exceeds the supply voltage $v_i = 3$ V, thus providing non-overlapping conduction times.

For the piecewise-linear model (14.62), we choose six segments ($s = 4$) given by the following data: $c_1 = 0.09$, $c_2 = 0.2238$, $c_3 = 0.4666$, $c_4 = 1.1605$, $c_5 = 2.8863$, $a_1 = 0$, $a_2 = 0.1$, $a_3 = 0.2487$, $a_4 = 0.6182$, $a_5 = 1.5383$. The relative error between $f(\cdot)$ and $f_{\mathsf{pwl}}(\cdot)$ is kept below 0.1 for $0.1 \leq x < 3.82$. The absolute error is less than $2 \cdot 10^{-3}$ for $0 \leq x < 0.1$ and 0 for negative x. In practise, the values of v_g, v_s, v_d, v_t in logic integrated circuits allow a good approximation of $f(\cdot)$ by $f_{\mathsf{pwl}}(\cdot)$.

The compensator amplifier is modelled as an 10^5 gain and an output low-pass filter with a cutoff frequency of 30 MHz, composed of $R_p = 1\,\Omega$ and $C_p = 5.3$ nF. The comparator is modelled as a piecewise-linear continuous function whose value is 0 if $x < -\epsilon$V and 3 if $x > \epsilon$V, with $\epsilon = 0.15$. The ramp voltage is at the 600 kHz frequency, and the bounds are 0 and $0.75 v_i = 2.25$ V. The rise time is 1.655 ns, and the fall time is 10 ns. For the other components, the standard values are $v_i = 3$ V, $L = 10\,\mu$H, $C = 22\,\mu$F, $R_{\mathrm{load}} = 10\,\Omega$, $R_{11} = 15.58\,\mathrm{k}\Omega$, $R_{12} = 227.8\,\mathrm{k}\Omega$, $R_{21} = 5.613\,\mathrm{M}\Omega$, $C_{11} = 20$ pF, $C_{21} = 1.9$ pF.

The reference voltage v_{ref} rises from 0 to 1.8 V in 0.1 ms at the beginning of the simulation. The output voltage v_{output} is regulated to track the reference voltage v_{ref} when v_i or v_{ref} or the load current vary. The error voltage v_{error} is a filtered value of the difference between v_{output} and v_{ref}. This voltage signal is converted into a time length thanks to a comparison with the periodic ramp signal. The comparator drives the pMOS transistor which in turn provides more or less charge to the output depending on the error level. The operation of a buck converter involves both a relatively slow dynamics when the switching elements (MOS and diodes) are keeping

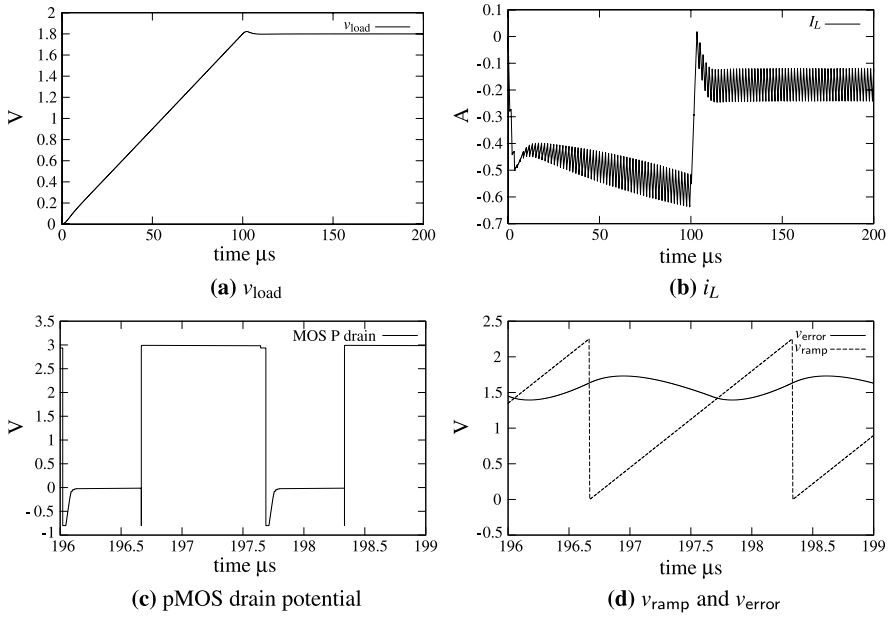


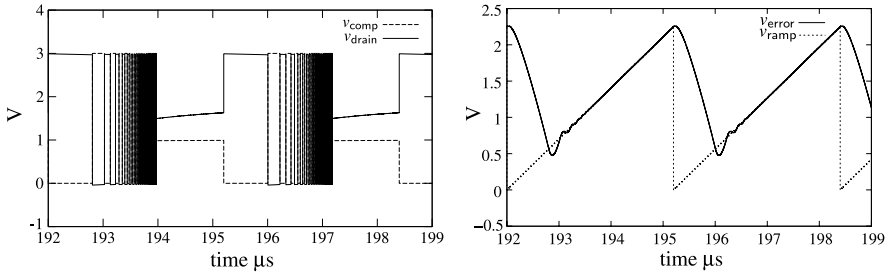
Fig. 14.10 SICONOS buck simulation using standard parameters

their conducting state, and a fast dynamics when the states change. The orders of magnitude are 50 ps for some switching details, 1 µs for a slow variation period and 100 µs at least for a settling period of the whole circuit requiring a simulation.

14.7.3 Simulation with SICONOS *with the Piecewise-Nonlinear Model*

As initial conditions, all state variables are zeroed. The detailed analysis of the switching events requires to use a time step as small as 50 ps. The simulations are carried with a fixed time-step, 4×10^6 steps are then computed for the 200 µs long settling of the output voltage. The one-step problem is solver with our own implementation of a semi-smooth Newton method based on the Fischer–Burmeister reformulation [38] and using a convergence tolerance of 10^{-12}. Let us detail the overall result displayed in Fig. 14.10:

1. Figure 14.10a is the output potential, following the ramp v_{ref}.
2. Figure 14.10b is the current through the inductor. Until 0.0001 s, i_L is loading the capacitor C. After 0.0001 s, i_L has to keep the capacitor charge constant.
3. Figure 14.10c zooms on the pMOS drain potential with standard parameters.
4. Figure 14.10d zooms on the v_{error} and v_{ramp} voltages.



(a) v_{comp} and v_{drain} **(b)** v_{ramp} and v_{error}

Fig. 14.11 SICONOS buck simulation using sliding-mode parameters and multi-valued comparator

The central processing unit (CPU) time required to achieve the simulation is 60 s on a Pentium 4 processor clocked at 3 GHz. It includes 19 s in the MLCP solvers and 40 s in matrices products. The time to export the resulting data is not included.

The simulation has been tested with many values of the parameters. The robustness of the non-smooth modelling and solving algorithms enables one to perform with the same CPU time the simulation of such cases. All the SICONOS simulations presented in this chapter have been obtained in one shot from the dynamical equations automatically generated from the netlist, without any further parameter tuning.

14.7.3.1 Simulations with Sliding Modes

This paragraph focuses on the simulation with sliding parameters and using a multi-valued model for the comparator. The rise time of the ramp voltage has been increased to 3.2 ns. The model used in SICONOS consists in setting the ϵ gap to 0. Figure 14.11 shows the SICONOS simulation using a fully implicit time-stepping. It could be noted that the comparator output is stabilised to an unsaturated value.

14.7.4 Comparison with SPICE Analog Solvers

The simulation of this circuit was done with several versions of SPICE (the open-source NGSPICE from Berkeley and ELDO from Mentor Graphics) and with two kinds of MOS models.

The first model, a MOS level 3, takes more physical effects into account than the piecewise-linear model used in SICONOS simulations, in particular the voltage-dependent capacitors. It is an important issue since these varying capacitors cause some convergence problems when node 2 switches between V_I and ground. Adding a small capacitor of a few picoFarad between this node and ground helps to solve the problem but may yield artefacts (spikes) on the current of the V_I alim and the

MOS transistors. The second model is the MOS simplified model (Sah model) with fixed capacitances and a quadratic static characteristic:

$$I_{DS} = \max(0, V_{GS} - Vt_N)^2 - \max(0, V_{GD} - Vt_N)^2.$$

This model is very close to the piecewise-linear model used in SICONOS simulations. The implementation in netlists was done thanks to voltage-dependent current sources that are very likely not compiled by the various SPICE simulators tested. Thus the measured CPU time is increased with respect to a compiled version. An estimation of the CPU time with a compiled simplified model may be given by multiplying the MOS level 3 CPU time by the ratio of the Newton–Raphson iterations required respectively during the simulations with each model. An additional correction should be done to reflect that the computation of the Jacobian matrix entries linked to a compiled simplified model would require less time than with a MOS level 3 model. Even if the SPICE simulation includes other operations, Jacobian matrix loading time is indeed known to be generally predominant.

Power MOSFETs intrinsic diodes are modelled by the classical Shockley equation with an emission coefficient $N = 1$:

$$I = I_S\left(e^{\frac{qV}{NkT}} - 1\right) \quad \text{when } V > -5N\frac{kT}{q},$$

$$I = -I_S \quad \text{when } V < -5N\frac{kT}{q},$$

where V and I are the voltage and current through the diode, I_S is the saturation current with default value 1×10^{-14} A, q the electron charge 1.6×10^{-19} C, k the Boltzmann constant 1.38×10^{-23} J/K, T temperature in K and N is the emission coefficient.

The comparator is modelled as a nonlinear voltage-controlled voltage source defined as

$$V_{\text{out}} = 1.5\big(\tanh(10V_{\text{in}}) + 1\big).$$

Thus the 3-segment characteristic used as the non-smooth model is regularised to help convergence of SPICE.

The power supply V_I is raised from 0 in 50 ns at the beginning to help the convergence.[3] The SPICE tolerance values used are 1 nA for currents, 1 μV for voltages and 0.00075 for relative differences. The maximum number of Newton–Raphson iterations is set to 100. The default values are 10 for NGSPICE and 13 for ELDO, which are not sufficient.

Usually, SPICE simulators integrate with a time-step adjusted according to different strategies based on an estimation of the local truncation error (LTE) or the number of Newton–Raphson iterations required by previous steps. Since SICONOS

[3]This is not required with the SICONOS algorithms that find a consistent initial solution from scratch.

Table 14.1 Numerical comparison on the buck converter example

Simulator	Model	# Newton iterations	CPU time (s)
Standard compensator values			
NGSPICE	simple	8024814	632
NGSPICE	level 3	8304237	370
ELDO	simple	4547579	388
ELDO	level 3	4554452	356
SICONOS	LCP	–	60
Sliding-mode compensator values			
NGSPICE	simple	8070324	638
NGSPICE	level 3	8669053	385
ELDO	simple	5861226	438
ELDO	level 3	5888994	367
SICONOS	LCP	–	60

simulations were carried with a fixed time step of 50 ps, simulators were forced to use this value as a maximum. Even when SPICE simulators use a fixed time-step, they may compute LTE to assess a solution found by the Newton–Raphson algorithm. This computation of LTE was disabled because it could impair the performance of SPICE with respect to SICONOS.[4]

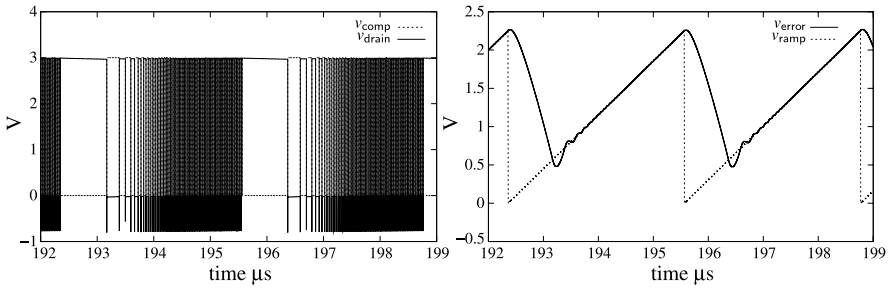
Table 14.1 displays the results with the standard and the sliding-mode values of compensator components. An estimation of the CPU time with a compiled simplified model is added.

These results shall be compared to the 60 s CPU time achieved with the time-stepping via complementarity approach. Depending on the model and the SPICE simulator, the (estimated) CPU time is from 2.8 to 6.1 larger than with SICONOS. Moreover, it was necessary to add a parasitic capacitor on the connection between the pMOS and nMOS transistors to allow the convergence of the NGSPICE simulator with the MOS level 3 model. We recall that all the SICONOS simulations presented have been obtained without any further parameter tuning.

14.7.4.1 Simulations with Sliding Modes

Simulation using ELDO has been done using the model $V_{\text{out}} = 1.5(\tanh(10000 V_{\text{in}}) + 1)$ for the comparator. The MOS level 3 leads to '`Newton no-convergence`' messages, so the MOS SAh model has been used to run the simulation displayed in

[4]For NGSPICE, it implied a slight modification of the source code since no standard option is provided to do it.



(a) v_{comp} and v_{drain} **(b)** v_{ramp} and v_{error}

Fig. 14.12 ELDO buck simulation using sliding-mode parameters and $V_{\text{out}} = 1.5(\tanh(10000V_{\text{in}}) + 1)$ for the comparator

Fig. 14.12. It is noteworthy that it does not handle the stabilisation of the comparator output on the sliding surface.

Remark 14.7.1 In Fig. 14.11a the sliding surface is attained in finite time after an accumulation of switches. This is a classical phenomenon in non-smooth systems, see Filippov's example in [2].

14.7.5 Simulation with PLECS

As we pointed out above, the hybrid approach that consists of an exhaustive enumeration of all the system's modes, soon becomes inefficient and unusable mainly because the simulation duration grows exponentially fast. Let us illustrate this fact with the buck converter, loaded with a resistance. The simulator is PLECS®, version 2.1, a hybrid simulator developed by Plexim.[5] The CPU time required to achieve the simulation of 200 µs ranges between 596 s to 4 hours, depending on the values of the resistors, capacitors and inductors and the existence of sliding modes. This should be compared to the 60 s of the SICONOS simulation, obtained independently of these components values. Moreover, the PLECS simulation performs only 168038 steps comparing the 4×10^6 steps performed during the SICONOS simulation. It can be concluded that the computation of one step of simulation using SICONOS is 250 faster than using PLECS. This demonstrates the robustness and efficiency of the time-stepping scheme via complementarity and the one-step problem algorithms based on mathematical programming techniques of SICONOS.

[5] http://www.plexim.com/.

14.8 Summary

In this chapter, we give an overview of the abilities of the event-capturing time-stepping schemes together with the complementarity framework to efficiently simulate switched electrical circuits. Contrary to the hybrid approach, these schemes avoid the explicit enumeration and the computation of the different modes of the system. They are able therefore to simulate a very large number of events with a very large number of switching components. They also avoid the convergence problem associated with the stiff systems generated by some regularisation procedures. Finally, they are also able to reproduce the finite time convergence of system on sliding-mode surfaces without any chattering. In this chapter, we have mainly focus on the Buck converter architecture. In [8, Chaps. 7 and 8], some other architectures of power converters (Ćuk converter) and a particular structure of analogue-to-digital converters and the delta–sigma converter show that the time-stepping approach via complementarity provides electrical design engineers with a very powerful prototyping and validating tool.

Acknowledgements The author would like to warmly thank Pascal Denoyelle for his contribution in the earlier version of this work and his two main co-workers on this project Olivier Bonnefon and Bernard Brogliato. The authors acknowledge Michael Ferris (University Wisconsin–Madison) for providing us with the PATH solver. Part of this work has been supported by the ANR project VAL-AMS (ANR-06-SETI-018-01).

References

1. Acary, V., Brogliato, B.: Numerical Methods for Nonsmooth Dynamical Systems: Applications in Mechanics and Electronics. Lecture Notes in Applied and Computational Mechanics, vol. 35. Springer, Berlin (2008)
2. Acary, V., Brogliato, B.: Implicit Euler numerical scheme and chattering-free implementation of sliding mode systems. Syst. Control Lett. **59**(5), 284–295 (2010)
3. Acary, V., Pérignon, F.: Siconos: A software platform for modeling, simulation, analysis and control of nonsmooth dynamical systems. Simul. News Eur. **17**(3–4), 19–26 (2007)
4. Acary, V., Brogliato, B., Goeleven, D.: Higher order Moreau's sweeping process: Mathematical formulation and numerical simulation. Math. Program. **113**(1), 133–217 (2008)
5. Acary, V., Bonnefon, O., Brogliato, B.: Improved circuit simulator. Patent number 09/02605 (2009)
6. Acary, V., Bonnefon, O., Brogliato, B.: Time-stepping numerical simulation of switched circuits with the nonsmooth dynamical systems approach. IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst. **29**(7), 1042–1055 (2010)
7. Acary, V., Brogliato, B., Orlov, Y.: Chattering-free digital sliding-mode control with state observer and disturbance rejection. IEEE Trans. Autom. Control (2011). doi:10.1109/TAC.2011.2174676. The Research Report RR-7326, INRIA (2010) is available as a preprint at http://hal.inria.fr/inria-00494417/PDF/RR-7326.pdf
8. Acary, V., Bonnefon, O., Brogliato, B.: Nonsmooth Modeling and Simulation for Switched Circuits. Lecture Notes in Electrical Engineering, vol. 69. Springer, Berlin (2011)
9. Bächle, S., Ebert, F.: Element-based topological index reduction for differential-algebraic equations in circuit simulation. Technical Report Preprint 05-246 (Matheon), Inst. f. Mathematik, TU Berlin (2005)

10. Bächle, S., Ebert, F.: Graph theoretical algorithms for index reduction in circuit simulation. Technical Report Preprint 05-245 (Matheon), Inst. f. Mathematik, TU Berlin (2005)
11. Bastien, J., Schatzman, M.: Numerical precision for differential inclusions with uniqueness. Math. Model. Numer. Anal. **36**(3), 427–460 (2002)
12. Billups, S.C., Dirkse, S.P., Ferris, M.C.: A comparison of large scale mixed complementarity problem solvers. Comput. Optim. Appl. **7**, 3–25 (1997)
13. Biolek, D., Dobes, J.: Computer simulation of continuous-time and switched circuits: Limitations of SPICE-family programs and pending issues. In: Proc. of the International Conference Radioelektronika, Brno, Czech Republic, pp. 1–11 (2007)
14. Brogliato, B., Goeleven, D.: Well-posedness, stability and invariance results for a class of multivalued Lur'e dynamical systems. Nonlinear Anal. **74**(1), 195–212 (2011)
15. Brogliato, B., Thibault, L.: Well-posedness results for non-autonomous complementarity systems. J. Convex Anal. **17**(3–4), 961–990 (2010)
16. Camlibel, M.K., Heemels, W.P.M.H., Schumacher, J.M.: Consistency of a time-stepping method for a class of piecewise-linear networks. IEEE Trans. Circuits Syst. I **49**(3), 349–357 (2002)
17. Camlibel, M.K., Heemels, W.P.M.H., Schumacher, J.M.: On linear passive complementarity systems. Eur. J. Control **8**(3), 220–237 (2002)
18. Camlibel, M.K., Heemels, W.P.M.H., van der Schaft, A.J., Schumacher, J.M.: Switched networks and complementarity. IEEE Trans. Circuits Syst. I **50**(8), 1036–1046 (2003)
19. Cao, M., Ferris, M.C.: A pivotal method for affine variational inequalities. Math. Oper. Res. **21**(1), 44–64 (1996)
20. Chung, H.S.H., Ioinovici, A.: Fast computer aided simulation of switching power regulators based on progressive analysis of the switches' state. IEEE Trans. Power Electron. **9**(2), 206–212 (1994)
21. Coddington, E.A., Levinson, N.: Theory of Ordinary Differential Equations. McGraw-Hill, New York (1955)
22. Cottle, R.W., Pang, J., Stone, R.E.: The Linear Complementarity Problem. Academic Press, Boston (1992)
23. Elmqvist, H., Mattsson, S.E., Otter, M.: Object-oriented and hybrid modeling in Modelica. J. Eur. Syst. Autom. **35**(4), 395–404 (2001)
24. Facchinei, F., Pang, J.S.: Finite-Dimensional Variational Inequalities and Complementarity Problems. Springer Series in Operations Research, vols. I & II. Springer, New York (2003)
25. Frasca, R., Camlibel, M.K., Goknar, I.C., Vasca, F.: State discontinuity analysis of linear switched systems via energy function optimization. In: Proc. of the IEEE International Symposium on Circuits and Systems, Seattle, Washington, USA, pp. 540–543 (2008)
26. Frasca, R., Camlibel, M.K., Goknar, I.C., Iannelli, L., Vasca, F.: Linear passive networks with ideal switches: Consistent initial conditions and state discontinuities. IEEE Trans. Circuits Syst. I **57**(12), 3138–3151 (2010)
27. Fukushima, M.: Equivalent differentiable optimization problems and descent methods for asymmetric variational inequality problems. Math. Program. **53**(1–3), 99–110 (1992)
28. Greenhalgh, S., Acary, V., Brogliato, B.: Preservation of the dissipativity properties of a class of nonsmooth dynamical systems with the (θ, γ)-algorithm. Research Report RR-7632, INRIA (2011). URL http://hal.inria.fr/inria-00596961/en
29. Hairer, E., Wanner, G.: Solving Ordinary Differential Equations II. Stiff and Differential-algebraic Problems, 2nd edn. Series in Computational Mathematics, vol. 14. Springer, London (1996)
30. Han, L., Tiwari, A., Camlibel, K., Pang, J.S.: Convergence of time-stepping schemes for passive and extended linear complementarity systems. SIAM J. Numer. Anal. **47**(5), 3768–3796 (2009)
31. Heemels, W.P.M.H., Camlibel, M.K., Schumacher, J.M.: A time-stepping method for relay systems. In: Proc. of the IEEE Conference on Decision and Control, Sydney, Australia, pp. 461–466 (2000)
32. Heemels, W.P.M.H., Schumacher, J.M., Weiland, S.: Linear complementarity systems. SIAM J. Appl. Math. **60**, 1234–1269 (2000)

33. Hiriart-Urruty, J.B., Lemaréchal, C.: Convex Analysis and Minimization Algorithms. Springer, Heidelberg (1993)
34. Iannelli, L., Vasca, F., Camlibel, M.K.: Complementarity and passivity for piecewise linear feedback systems. In: Proc. of the IEEE Conference on Decision and Control, San Diego, California, USA, pp. 4212–4217 (2006)
35. Isidori, A.: Nonlinear Control Systems, 3rd edn. Springer, London (1995)
36. Leenaerts, D.M.W., Bokhoven, W.M.V.: Piecewise Linear Modeling and Analysis. Kluwer Academic Publishers, Dordrecht (1998)
37. Leenarts, D.M.: On linear dynamic complementarity systems. IEEE Trans. Circuits Syst. I **46**(8), 1022–1026 (1999)
38. Luca, T.D., Facchinei, F., Kanzow, C.: A semismooth equation approach to the solution of nonlinear complementarity problems. Math. Program. **75**(3), 407–439 (1996)
39. Maffezzoni, P., Codecasa, L., D'Amore, D.: Event-driven time-domain simulation of closed-loop switched circuits. IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst. **25**(11), 2413–2426 (2006)
40. Marques, M.D.P.M.: Differential Inclusions in Nonsmooth Mechanical Problems. Shocks and Dry Friction. Progress in Nonlinear Differential Equations and Their Applications, vol. 9. Birkhäuser, Boston (1993)
41. Mayaram, K., Lee, D.C., Moinian, D.A., Roychowdhury, J.: Computer-aided circuit analysis tools for RFIC simulation: Algorithms, features, and limitations. IEEE Trans. Circuits Syst. II **47**(4), 274–286 (2000)
42. Moreau, J.J.: Evolution problem associated with a moving convex set in a Hilbert space. J. Differ. Equ. **26**, 347–374 (1977)
43. Moreau, J.J.: Bounded variation in time. In: Moreau, J.J., Panagiotopoulos, P.D., Strang, G. (eds.) Topics in Nonsmooth Mechanics, pp. 1–74. Birkhäuser, Basel (1988)
44. Pang, J.S., Stewart, D.: Differential variational inequalities. Math. Program. **113**(2), 345–424 (2008)
45. Pogromski, A.Y., Heemels, W.P.M.H., Nijmeijer, H.: On solution concepts and well-posedness of linear relay systems. Automatica **39**, 2139–2147 (2003)
46. Rockafellar, R.T.: Convex Analysis. Princeton University Press, Princeton (1970)
47. Sargent, R.W.H.: An efficient implementation of the Lemke algorithm and its extension to deal with upper an lower bounds. Math. Program. Stud. **7**, 36–54 (1978)
48. van Bokhoven, W.M.G.: Piecewise linear analysis and modelling. Ph.D. thesis, Technical University of Eindhoven, TU/e (1981)
49. van Bokhoven, W.M.G., Jess, J.A.G.: Some new aspects of P and P_0 matrices and their application to networks with ideal diodes. In: Proc. of the IEEE International Symposium on Circuits and Systems, New York, USA, pp. 806–810 (1978)
50. van Eijndhoven, W.M.G.: A piecewise linear simulator for large scale integrated circuits. Ph.D. thesis, Technical University of Eindhoven, TU/e (1984)
51. van Stiphout, M.T.: Plato—a piecewise linear analysis for mixed-level circuit simulation. Ph.D. thesis, Technical University of Eindhoven, TU/e (1990)
52. Vandenberghe, L., Moor, B.L.D., Vandewalle, J.: The generalized linear complementarity problem applied to the complete analysis of resistive piecewise-linear circuits. IEEE Trans. Circuits Syst. **36**(11), 1382–1391 (1989)
53. Yuan, F., Opal, A.: Computer methods for switched circuits. IEEE Trans. Circuits Syst. I **50**(8), 1013–1024 (2003)
54. Zhu, D.L., Marcotte, P.: Modified descents methods for solving the monotone variational inequality problem. Oper. Res. Lett. **14**(2), 111–120 (1993)
55. Zhu, D.L., Marcotte, P.: An extended descent framework for monotone variational inequalities. J. Optim. Theory Appl. **80**(2), 349–366 (1994)