Periodic steady-state analysis of free-running oscillators

EINDHOVEN UNIVERSITY OF TECHNOLOGY
Department of Mathematics and Computing Science

RANA 00-16
December 2000

Periodic Steady-State Analysis of Free-running Oscillators

by

S.H.M.J. Houben and J.M. Maubach

Reports on Applied and Numerical Analysis
Department of Mathematics and Computing Science
Eindhoven University of Technology
P.O. Box 513
5600 MB Eindhoven, The Netherlands
ISSN: 0926-4507

Periodic Steady-State Analysis of Free-running Oscillators

ir. Stephan H.M.J. Houben (stephanh@win.tue.nl)
dr. Joseph M. Maubach (maubach@win.tue.nl)
Eindhoven University of Technology

1 Introduction

A common problem in the simulation of electric circuits for RF (Radio Frequency) applications is finding a periodic steady-state (PSS) of such a circuit. Several approaches exist for solving this problem. For non-autonomous circuits, i.e. circuits that are driven by an input source with an a priori known period T, many methods exist (see [1], [5], [7]). However, when dealing with autonomous circuits, the situation is less satisfactory. For an autonomous circuit, the period T becomes an additional unknown, which makes the resulting system under-determined. A common solution method is harmonic balance, a frequency-domain method (see [3], [8]). Harmonic balance performs well if the waveform to be computed contains mostly low harmonics, but it becomes very expensive if a large number of harmonics is present. Therefore, there has been much interest in hybrid (see [4]) and pure time-domain methods (see [3]), such as shooting or finite difference. However, convergence of these methods is often problematic; typically there is only convergence when the initial guess for the period T_0 is already very close to the actual solution T^*.

In this paper, two novel methods will be presented. The first, called Poincaré-map method, has very strong convergence properties, but converges only linearly for many real-world circuits. The second, the accelerated Poincaré-map method, converges super-linearly when the Poincaré-map method converges linearly, but has somewhat weaker convergence properties. Some numerical results comparing both methods will be presented.

The algorithms discussed in this paper are described in more detail in the upcoming paper [2]. This paper also contains more numerical experiments and a discussion of how differential-algebraic equations (DAE) can be handled. In this paper, we restrict ourselves to ordinary differential equations (ODE).

2 Periodic steady-state

Definition 1. *Consider an autonomous ordinary differential equation (ODE) of the form:*

$$\frac{d\mathbf{x}}{dt} = \mathbf{f}(\mathbf{x}), \quad \mathbf{x} \in \mathbf{R}^n. \tag{1}$$

A function $\mathbf{x} : \mathbf{R} \to \mathbf{R}^n$ *is called a* periodic steady-state *(PSS) of* (1) *if:*

1. $\mathbf{x}$ *is a solution to* (1).
2. $\mathbf{x}$ *is periodic, i.e. there is a* $T > 0$ *such that for all* $t \in \mathbf{R}$, $\mathbf{x}(t) = \mathbf{x}(t+T)$.

Note that according to this definition, a stationary solution, i.e. a solution of the form $\mathbf{x}(t) \equiv \mathbf{x}_0$, is also a PSS.

Definition 2. *The* limit cycle $\mathcal{C}(\mathbf{x})$ *of a PSS* $\mathbf{x}$ *is the range of the function* $\mathbf{x}(t)$, *i.e.*

$$\mathcal{C}(\mathbf{x}) = \{\mathbf{x}(t) \,|\, t \in \mathbf{R}\}. \tag{2}$$

A set $\mathcal{C}$ *is called a limit cycle of* (1) *if there is a PSS* $\mathbf{x}$ *of* (1) *so that* $\mathcal{C} = \mathcal{C}(\mathbf{x})$.

Definition 3. *A periodic steady-state* $\mathbf{x}$ *is called* stable[1] *if there is a* $\delta > 0$ *so that the following holds: For every solution* $\mathbf{x}^*$ *to* (1) *which has the property that*

$$\exists_{\tau_1 > 0} ||\mathbf{x}^*(0) - \mathbf{x}(\tau_1)|| < \delta, \tag{3}$$

there exists a $\tau_2 > 0$ *so that*

$$\lim_{t \to \infty} ||\mathbf{x}^*(t)(0) - \mathbf{x}(t + \tau_2)|| = 0 \tag{4}$$

A limit cycle is called stable when one of its periodic steady-states is stable; for an ODE of the form (1), *this implies that* ***all*** *of its periodic steady-states are stable.*

In this paper, we will concentrate on methods for finding a stable periodic steady-state. Periodic steady-states that are not stable are not interesting for the IC designer, since they do not correspond to any physical behaviour of the modelled circuit. In fact, we want to actively avoid non-stable periodic steady-states for this reason.

3 Autonomous oscillating circuits

The circuits in which we are interested are so-called *autonomous* or *free-running* oscillators. Such oscillators have the property that they do not have any time-dependent input signals. This implies that they can be mathematically described by an *autonomous* ordinary differential equation or differential-algebraic equation. In this paper, we restrict ourselves to circuits that can be described

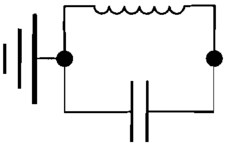

Figure 1: The LC-ring is a very simple free-running oscillator. It consists of a capacitor and an inductor.

with an autonomous ODE of the form (1). An example of such an oscillator is given in figure 1. The equations describing this particular circuit are:

$$\frac{dv}{dt} = \frac{1}{C}i, \qquad \frac{di}{dt} = \frac{-1}{L}v. \tag{5}$$

Note that these equations are linear. For initial conditions $v(0) = v_0$, $i(0) = i_0$, the solution to (5) is given by:

$$v(t) = v_0 \cos\omega t + \sqrt{\frac{L}{C}} i_0 \sin\omega t, \tag{6a}$$

$$i(t) = i_0 \cos\omega t - \sqrt{\frac{C}{L}} v_0 \sin\omega t, \tag{6b}$$

where $\omega = 1/\sqrt{CL}$. For this problem, it isn't difficult to find a PSS, since *every* solution $\mathbf{x}$ of (5) is periodic and hence a PSS. However, none of these PSS is stable.

A nonlinear example of a free-running oscillator is given by the following equations:

$$\frac{dx}{dt} = y + h(\sqrt{x^2 + y^2})x, \tag{7a}$$

$$\frac{dy}{dt} = -x + h(\sqrt{x^2 + y^2})y, \tag{7b}$$

The function h is chosen so that:

1. h continuous and differentiable.

2. $h(0) > 0$.

3. there are several points $r_k > 0$ so that $h(r_k) = 0$.

Possible choices of h include $h(r) = \cos r$, and $h(r) = \varepsilon(1 - r)$.

The problem (7) has the following properties:

1. It has at least one PSS solution, namely the stationary state with $\mathbf{x} = \mathbf{0}$. However, this solution is unstable.

[1] some authors prefer the term ***strongly stable***

2. For every $r_k > 0$ with $h(r_k) = 0$, we have that the circle described by $x^2 + y^2 = r_k^2$ is a limit cycle. Moreover, if $h'(r_k) < 0$, then the limit cycle is stable.

3. As $h'(r_k) \to 0$ from below, the limit cycle $x^2 + y^2 = r_k^2$ becomes a weaker and weaker attractor for nearby solutions of (5).

As we will see later, finding a stable PSS becomes more difficult when the PSS behaves only as a weak attractor, i.e. convergence towards the PSS is very slow.

4 The Poincaré-map method

The Poincaré-map method is based on the following observation: starting sufficiently close to a stable limit cycle $\mathcal{C}$, a transient simulation will eventually converge towards $\mathcal{C}$. After all, this is implied in the definition of a stable limit cycle. There are, however, two disadvantages to this approach:

1. We have to find a way to detect if we have approached the PSS close enough. If T is known, a "running window" can be used, i.e. the value $\mathbf{x}(t)$ at the current integration time t is compared to the value at $\mathbf{x}(t - T)$. However, T is an unknown in the autonomous case.

2. Convergence will be linear at best, which means that excessive computing time is needed to arrive at the solution.

The first problem will be addressed in this section, leading to the (unaccelerated) Poincaré-map method. The second problem will be addressed in the next section when considering the Accelerated Poincaré-map method.

The length of the period can be estimated by looking for periodic recurring features in the computed circuit behaviour. A possible recurring feature is the point at which a specific condition (the so-called *switch condition*) is satisfied. This is equivalent to carrying out a Poincaré-map iteration. The switch condition has to be chosen in such a way that the solution becomes locally unique. Moreover, the switch condition has to be satisfied at some point during the periodic steady state. In [2], some heuristics for finding a suitable switch condition are given.

The unaccelerated Poincaré-map method can now be described as follows.

Algorithm 1. *Provide the algorithm with the following inputs: an initial state* $\mathbf{x}_0$*, a switch condition of the form* $(\mathbf{v}, \mathbf{x}(t)) = \alpha$*, and a tolerance* $\varepsilon > 0$*. The algorithm will iteratively produce approximations for the period* T *and for a point on the periodic waveform* $\mathbf{x}^*$*.*

1. *Set* $i \leftarrow 0$ *and* $t_0 \leftarrow 0$.

2. *Starting with* $t = t_i$, $\mathbf{x}(t_i) = \mathbf{x}_i$, *integrate* (1) *until* $(\mathbf{v}, \mathbf{x}(t)) = \alpha$ *and* $d(\mathbf{v}, \mathbf{x}(t))/dt > 0$.

3. *Set* $\mathbf{x}_{i+1} \leftarrow \mathbf{x}(t)$ *and* $t_{i+1} \leftarrow t$.

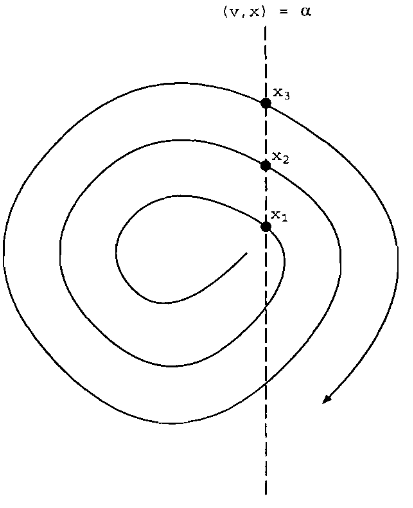

Figure 2: The trajectory of a solution $\mathbf{x}(t)$ to the Initial Value Problem. The points $\mathbf{x}_n$ are chosen so that they satisfy $(\mathbf{x}, \mathbf{v}) = \alpha$, for some given $\mathbf{v}$ and α.

4. *Compute* $\delta_i := ||\mathbf{x}_{i+1} - \mathbf{x}_i||$. *If* $\delta_i > \varepsilon$, *set* $i \leftarrow i + 1$ *and proceed to step 2. If* $\delta_i \leq \varepsilon$, *proceed to step 5.*

5. *Set* $T \leftarrow t_{i+1} - t_i$ *and* $\mathbf{x}^* \leftarrow \mathbf{x}_{i+1}$. *Done.*

This method seems promising for two reasons:

1. It has rather good convergence properties.
2. It is simple to implement in an existing simulator, since it can essentially be considered as a post-processing step to an ordinary transient simulation.

5 The accelerated Poincaré-map method

The Poincaré-map method essentially leads us to find the fixed point of a function $F : \mathbf{R}^n \to \mathbf{R}^n$. This function F can be formally defined as:

$$F(\mathbf{x}_0) := \mathbf{x}(T), \tag{8}$$

where $\mathbf{x}(t)$ is the solution of (1) with $\mathbf{x}(0) = \mathbf{x}_0$, and T is the smallest $t > 0$ such that $(\mathbf{v}, \mathbf{x}(t)) = \alpha$ and $d(\mathbf{v}, \mathbf{x}(t))/dt > 0$. Given $\mathbf{x}_0$, the vector $F(\mathbf{x})$ can effectively be computed by using Algorithm 1, i.e. by applying the ordinary Poincaré-map method.

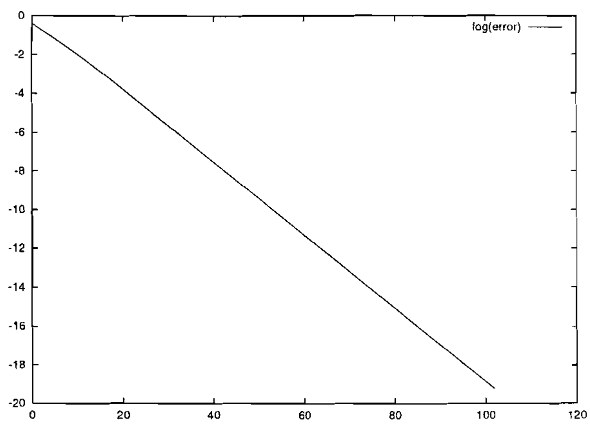

Figure 3: log(error) after each iteration for the Poincaré-map method applied to (7) with $h(r) := \varepsilon(1-r), \varepsilon = 3 \cdot 10^{-2}$.

The successive approximations of the Poincaré-map method satisfy the recursion:

$$\mathbf{x}_{n+1} = F(\mathbf{x}_n) \tag{9}$$

Note that the n-th iteration $\mathbf{x}_n$ does *not* include the period T. Suppose that this sequence converges linearly to some fixed point $\mathbf{x}^*$ of F. As said in the previous section, convergence might be slow. Hence we are interested in accelerating convergence using an acceleration method.

An acceleration method operates on the first k vectors of a sequence $\{\mathbf{x}_n\}$, and produces an approximation $\mathbf{y}$ to the limit of $\{\mathbf{x}_n\}$. This approximation can then be used to restart (9) and generate the beginning of a new sequence $\mathbf{y}_0, \mathbf{y}_1, \mathbf{y}_2, \ldots$. Again, the acceleration method can be applied to this new sequence, resulting in a new approximation $\mathbf{z}$ of the limit. The idea is that the sequence $\mathbf{x}, \mathbf{y}, \mathbf{z}, \ldots$ converges much faster to the limit of $\{\mathbf{x}_n\}$ than the sequence $\{\mathbf{x}_n\}$ itself. Typically, if $\{\mathbf{x}_n\}$ converges linearly, then $\{\mathbf{x}, \mathbf{y}, \mathbf{z}, \ldots\}$ converges super-linearly.

We applied the accelerated Poincaré-map method based on the well-known minimal polynomial extrapolation (MPE) method. Rather than describing MPE here in detail, the reader is referred to [6]. With MPE, we obtain a super-linear converging sequence , provided that the original sequence produced has in the limit linear convergence. This is typically the case for periodic circuits, and it is in particular the case for our test problems.

6 Numerical results

In this section we compare the ordinary Poincaré-map method and the accelerated Poincaré-map method. They have been applied to problem (7), where h

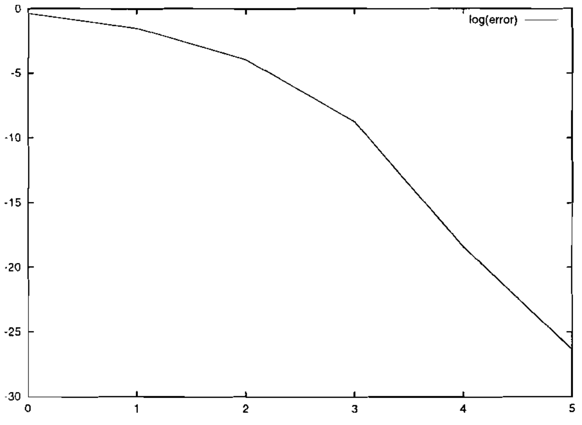

Figure 4: log(error) after each outer loop iteration for the accelerated Poincaré-map method applied to (7) with $h(r) := \varepsilon(1-r), \varepsilon = 3 \cdot 10^{-2}$.

was chosen as $h(r) := \varepsilon(1-r)$. A switch condition of the form $y = 0$ was taken. For all $\varepsilon > 0$, the resulting problem has exactly one stable limit cycle, namely the unit circle. The parameter $\varepsilon > 0$ affects the speed of convergence towards the limit cycle; as ε approaches 0, speed of convergence also goes to 0.

The number of iterations needed for decreasing values of ε is shown in Table 1. From this, it is easy to see that the unaccelerated Poincaré-map method

ε	$1 \cdot 10^0$	$1 \cdot 10^{-1}$	$3 \cdot 10^{-2}$	$1 \cdot 10^{-2}$	$3 \cdot 10^{-3}$	$1 \cdot 10^{-3}$
Unaccel. Poincaré	6	35	104	290	899	2516
Accel. Poincaré	7	11	17	17	20	23

Table 1: The numbers of iterations (i.e. the number of evaluations of F) needed by both methods for decreasing values of ε

becomes impractical when ε approaches 0. On the other hand, the accelerated Poincaré-map method performs well even for very small values of ε.

For $\varepsilon = 3 \cdot 10^{-2}$, the errors after each iteration for both methods have been plotted in Figures 3 and 4. Note that in Figure 4, only the error after each outer loop iteration has been plotted, whereas the number in Table 1 indicates the total number of iterations.

From Figures 3 and 4, it is clear that the unaccelerated Poincaré-map method gives linear convergence, whereas accelerated Poincaré gives super-linear convergence for this test problem.

7 Conclusions

The following conclusions can be drawn:

1. Unaccelerated Poincaré is impractical for finding the PSS, because of its slow convergence.
2. Unaccelerated Poincaré probably is a good way to generate an initial approximation. After that, we might switch to the accelerated Poincaré method.
3. Accelerated Poincaré gives super-linear convergence towards the solution in all the test cases.
4. Both methods are simple to implement in existing simulators, since they can be implemented as a post-processing step to an ordinary transient simulation. Implementation details for both methods are given in [2].

References

[1] T. J. Aprille and T.N. Trick. Steady state analysis of nonlinear circuits with periodic inputs. *Proceedings IEEE*, 60(1):108–114, 1972.

[2] Stephan H.M.J. Houben and Joseph M. Maubach. Periodic steady-state analysis of free-running oscillators. *pre-print*, 2000.

[3] Ken Kundert. Simulation methods for RF integrated circuits. In *Proceedings of ICCAD'97*, 1997.

[4] A. Semlyen and A. Medina. Computation of the periodic steady state in systems with nonlinear components using a hybrid time and frequency domain method. *IEEE Transactions on Power Systems*, 10(3):1498–1504, 1995.

[5] Stig Skelboe. Time-domain steady-state analysis of nonlinear electrical systems. *Proceedings of the IEEE*, 70(10):1210–1228, 1982.

[6] David A. Smith, William F. Ford, and Avram Sidi. Extrapolation methods for vector sequences. *SIAM Review*, 29(2):199–233, 1987.

[7] R. Telichevesky, K. Kundert, I. Elfadel, and J. White. Fast simulation algorithms for RF circuits. In *Proceedings of the IEEE 1996 Custom Integrated Circuits Conference*, pages 437–444. IEEE, New York,USA, 1996.

[8] E.J.W. ter Maten. Numerical methods for frequency domain analysis of electronic circuits. *Survey on Mathematics for Industry*, 8:171–185, 1999.