

# Circuits in Power Electronics

J. Waldvogel

Research Report No. 94-13  
October 1994

Seminar für Angewandte Mathematik  
Eidgenössische Technische Hochschule  
CH-8092 Zürich  
Switzerland

# Circuits in Power Electronics\*

J. Waldvogel

Seminar für Angewandte Mathematik  
Eidgenössische Technische Hochschule  
CH-8092 Zürich  
Switzerland

Research Report No. 94-13

October 1994

## Abstract

The dynamics of circuits in power electronics may often be modelled by means of systems of linear differential equations with piecewise constant coefficients and sinusoidal inhomogeneities. The Static Var (Volt-Ampere reactive) Compensator (SVC, a device for power network control) is a representative example. Initial value problems of this type may be handled by means of the matrix exponential, whereas the construction of periodic solutions is in itself a linear problem.

The linear algebra package MATLAB is ideally suited for numerically performing the above mentioned tasks, including stability analysis and spectral analysis of periodic solutions. This paper begins with the description of the engineering background and a brief outline of the well known mathematical theory. Then a transparent, but efficient program capable of generating, plotting and analyzing periodic solutions is presented and documented.

**Keywords:** circuit, exponential matrix, linear differential equation, MATLAB, periodic solution, piecewise constant coefficients, power electronics

**Subject Classification:** 15, 34, 65

\* To appear in: W. Gander and J. Hrebicek (eds.): *Solving Problems in Scientific Computing using Maple and MATLAB*, Second Edition, pp. 299 - 311, Springer Verlag, 1995.

# 1 Introduction

Over the last few years high power semi-conductor devices with intrinsic turn-off capability have become available. These devices, called gate turn-off (GTO) thyristors, consist of several layers of silicon with appropriate dopants; they are able to turn off currents of 1000 Amperes at thousands of volts within microseconds. In circuits used in power electronics the usual resistive, inductive and capacitive circuit elements are combined with thyristors which may simply be considered as switches. This technology is still an active field of research, and it has many important applications such as AC/DC conversion (both ways), speed control of locomotives and electric cars, control of power stations and power networks, etc.

For every fixed state of the thyristor switches Kirchhoff's laws must be satisfied, and therefore the dynamical behavior of such a circuit is described by a system of linear ordinary differential equations with constant coefficients, assuming linearity of the circuit elements. If the switches change their positions the structure of the circuit changes, but the final state of the currents in the circuit before switching determines the initial conditions after the switching.

Therefore the mathematical model of a circuit in power electronics is a system of linear differential equations with piecewise constant coefficients if the switching times are neglected. We assume the dynamics of the circuit to be described by  $n$  continuous functions of time  $t$ , which are represented by the vector  $\mathbf{x}(t) \in \mathbb{R}^n$  of dependent variables. By using matrix notation the model may be written as

$$\dot{\mathbf{x}} = A(t)\mathbf{x} + \mathbf{p}(t) \tag{1}$$

where dots denote derivatives with respect to time,  $A(t)$  is a step function, i.e., a given piecewise constant  $n \times n$  matrix, and  $\mathbf{p}(t) \in \mathbb{R}^n$  is a given forcing function. In the environment of AC (alternating current) circuits  $A(t)$  and  $\mathbf{p}(t)$  are often periodic functions. With no loss of generality the period will be normalized to  $2\pi$ , and in view of the Fourier decomposition of  $\mathbf{p}(t)$  we will use the first harmonic as a model case. Usually the solution  $\mathbf{x}(t)$  is specified by initial conditions  $\mathbf{x}(0) = \mathbf{x}_0$ , but other specifications, e.g. periodicity of  $\mathbf{x}(t)$ , will be considered in Section 3.

The initial value problem of systems of linear differential equations with a constant matrix  $A$  is a topic of elementary calculus (cf. [1]) and may be handled via the eigenvalues and eigenvectors of  $A$  or via the matrix exponential  $e^{At}$ . Even if the matrix  $A(t)$  is piecewise constant the explicit solution of the initial value problem is straight-forward, although quite laborious if  $A(t)$  has many discontinuities.

It turns out that the features of MATLAB allow a very elegant construction of the solution  $\mathbf{x}(t)$  of this initial value problem. Also, periodic solutions may be easily calculated and plotted. In this article we will use a specific device from the field of electric power network control, the so-called Static Var Compensator (SVC), to explain the use of MATLAB in power electronics. Var stands for Volt-Ampere reactive.

SVCs are used in electric power networks to compensate for the voltage drop due to the losses in the power lines and due to a variable user load. In the usual three-phase AC system six switching operations per period are needed, in order to transfer an impulse

of reactive power from one phase to another during an appropriate time interval in each period.

fig1-circuits.eps  
0 × 0 mm

Figure 1: Simplified SVC circuit

<i>Description of the SVC circuit</i>	
Variable	Description
$U_R(t), U_S(t), U_T(t)$	AC voltages generated by the power station
$R_M, L_M$	Resistance and inductance of the mains
$R_C, L_C, C$	Resistance, inductance and capacitance of the SVC
$I_R(t), I_S(t), I_T(t)$	Currents injected in the mains
$U(t)$	Voltage across the DC capacitance of the SVC

In Figure 1 a simplified SVC circuit including the mains is shown, the thyristors being represented by switches. For simplicity, no loads of the electric network are taken into consideration (they would branch off between  $L_M$  and  $R_C$ ). In each circuit corresponding to a single phase only one switch may be closed at any time. For the circuit of Figure 1 it is sufficient to choose  $n = 3$  independent variables, e.g.

$$x_1(t) = U(t), \quad x_2(t) = I_R(t), \quad x_3(t) = \frac{1}{\sqrt{3}}(I_S(t) - I_T(t)). \quad (2)$$

Then the dynamics of the circuit is described by Equation (1) with

$$\mathbf{p}(t) = \frac{1}{L}(0, \cos t, \sin t)^T, \quad L = L_M + L_C, \quad R = R_M + R_C \quad (3)$$

and

$$A(t) = B(\varphi(t)), \quad B(\varphi) = \begin{bmatrix} 0 & \frac{1}{C} \cos \varphi & \frac{1}{C} \sin \varphi \\ -\frac{2}{3L} \cos \varphi & -\frac{R}{L} & 0 \\ -\frac{2}{3L} \sin \varphi & 0 & -\frac{R}{L} \end{bmatrix}. \quad (4)$$

The switching angle  $\varphi(t)$  is a given piecewise constant function that controls the operation of the thyristors. In the 6-pulse SVC it is chosen as

$$\varphi(t) = \frac{\pi}{3} \text{round} \left( \frac{3}{\pi}(t - \tau) \right), \quad (5)$$

where the shift  $\tau$  is a parameter of the SVC to be chosen in  $|\tau| \leq R$ . Typical values of the parameters  $C, L, R$ , expressed in normalized units such that the period of the AC (often 0.02 sec) becomes  $2\pi$  are

$$C = 0.2, \quad L = 0.15, \quad R = 0.005. \quad (6)$$

The practical orders of magnitude of  $C, L$  and  $R$  are 150 nF, 2Hy and  $20\Omega$  respectively. For more technical details the reader is referred to the textbooks [2], [3].

## 2 Linear Differential Equations with Piecewise Constant Coefficients

We consider the differential equation (1),

$$\dot{\mathbf{x}} = A(t)\mathbf{x} + \mathbf{p}(t),$$

for the unknown function  $\mathbf{x}(t) \in \mathbb{R}^n$ . For simplicity the given  $2\pi$ -periodic forcing function  $\mathbf{p}(t)$  is assumed to contain the first harmonic only,

$$\mathbf{p}(t) = \mathbf{b}e^{it} + \bar{\mathbf{b}}e^{-it}, \quad \mathbf{b} \in \mathbb{C}^n. \quad (7)$$

The use of complex notation turns out to be advantageous since it greatly simplifies the equations, and it is fully supported by MATLAB. The piecewise constant real matrix  $A(t)$  is assumed to be  $2\pi$ -periodic as well. Therefore the  $m + 1$  (possible) discontinuities (jumps)  $t_k$  of  $A(t)$  will be introduced as

$$0 = t_0 < t_1 < t_2 < \dots < t_{m-1} < t_m = 2\pi,$$

and the discrete values of the matrix are denoted by

$$A(t) = A_k \text{ in } t_k \leq t < t_{k+1}, \quad k = 0, \dots, m-1. \quad (8)$$

Given initial conditions  $\mathbf{x}(0) = \mathbf{x}_0$ , Equation (1) has a unique solution  $\mathbf{x}(t)$ ; its values at the jumps are denoted by

$$\mathbf{x}_k := \mathbf{x}(t_k), \quad k = 0, \dots, m.$$

First, we construct the explicit solution  $\mathbf{x}(t)$  of (1) in the  $k$ -th subinterval  $t \in [t_k, t_{k+1}]$  satisfying the appropriate differential equation and initial condition:

$$\left. \begin{array}{l} \dot{\mathbf{x}}(t) = A_k \mathbf{x}(t) + \mathbf{p}(t), \quad t \in [t_k, t_{k+1}] \\ \mathbf{x}(t_k) = \mathbf{x}_k \end{array} \right\} \quad k = 0, \dots, m-1. \quad (9)$$

As usual we use the decomposition

$$\mathbf{x}(t) = \mathbf{y}(t) + \mathbf{z}(t) \quad (10)$$

into a conveniently chosen particular solution  $\mathbf{z}(t)$  and the solution  $\mathbf{y}(t)$  of the homogeneous problem satisfying

$$\dot{\mathbf{y}}(t) = A_k \mathbf{y}(t), \quad \mathbf{y}(t_k) = \mathbf{x}_k - \mathbf{z}(t_k). \quad (11)$$

Using the well-known matrix exponential we obtain

$$\mathbf{y}(t) = e^{A_k t} \mathbf{c}_k \quad (12)$$

with  $\mathbf{c}_k$  determined from the second equation of (11).

The computation of the matrix exponential is a non-trivial problem with a long history, as is seen from the title of the survey paper [4], “Nineteen Dubious Ways to Compute the Exponential of a Matrix”. In the MATLAB command `expm` the “least dubious” way, the method of Padé approximation, is implemented and works reliably, accurately and quickly in almost all cases. It is an expensive operation, however, requiring up to  $30n^3$  flops for an  $n \times n$  matrix.

If the matrix  $B$  is diagonalizable  $e^B$  may easily be computed via the eigenvalue factorization  $B = TDT^{-1}$ ,  $D = \text{diag}(\lambda_1, \dots, \lambda_n)$  as

$$e^B = Te^DT^{-1}, \quad e^D = \text{diag}(e^{\lambda_1}, \dots, e^{\lambda_n}), \quad (13)$$

where  $\lambda_j$  is the  $j$ -th eigenvalue of  $B$ .

In the case of the matrix  $A(t) = B(\varphi(t))$  defined in (4) the eigenvalues turn out to be independent of  $\varphi$  and may be written explicitly as

$$\lambda_1 = -r, \quad \lambda_{2,3} = -r/2 \pm i\omega \quad (14)$$

where

$$r = -R/L, \quad \omega = \sqrt{\frac{2/3}{LC} - \frac{R^2}{4L^2}}.$$

Therefore,  $\lambda_1, \lambda_2, \lambda_3$  are also the eigenvalues of all the matrices  $A_k$  for any choice of the switching times  $t_k$ . This is seen by means of the similarity relation

$$B(\varphi) = S(\varphi)B(0)S(\varphi)^{-1}$$

with the orthogonal matrix

$$S(\varphi) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\varphi) & -\sin(\varphi) \\ 0 & \sin(\varphi) & \cos(\varphi) \end{bmatrix}.$$

We then have

$$e^{B(\varphi)} = S(\varphi)e^{B(0)}S(\varphi)^{-1},$$

with the great advantage that  $e^{B(0)}$  has a simple explicit representation due to the block diagonal structure of  $B(0)$ :

$$e^{B(0)} = e^{r/2} \begin{bmatrix} \cos(\omega)I + \sin(\omega)/\omega B_0 & 0 \\ 0 & e^{r/2} \end{bmatrix}.$$

Here  $I$  is the  $2 \times 2$  unit matrix,  $B_0$  is the upper left  $2 \times 2$  block of  $B(0)$ , and  $r, \omega$  are defined above. The reader is invited to derive or verify these relations by means of MAPLE. An implementation of  $e^{B(\varphi)}$  is possible with about 30 elementary operations and 6 functions calls (such as `exp` or `sin`).

To complete the construction of the solution in the  $k$ -th time interval a particular solution  $\mathbf{z}(t)$  of (9) has to be chosen. The simplest choice is a harmonic oscillation of the same frequency as  $\mathbf{p}(t)$ , i.e.

$$\mathbf{z}(t) = -\mathbf{u}_k e^{it} - \bar{\mathbf{u}}_k e^{-it} \quad (15)$$

where the complex vector  $\mathbf{u}_k \in \mathbb{C}^n$  must be determined such that (9) is satisfied. Inserting (15) into (9) yields the condition

$$(A_k - iI)\mathbf{u}_k = \mathbf{b}, \quad (16)$$

a system of linear equations in  $C$  for  $\mathbf{u}_k$ , where  $I$  is the  $n \times n$  unit matrix. Therefore, the necessary and sufficient condition for the existence of a solution of the form (15) is  $\det(A_k - iI) \neq 0$ , i.e.  $\pm i$  must not be an eigenvalue of  $A_k$ . From (14) there follows that this is satisfied as long as

$$R \neq 0 \quad \text{or} \quad \frac{2L}{3C} - \frac{R^2}{4} \neq L^2, \quad (17)$$

which is true for the specific data given in (6). The resonant case (Condition (17) violated) may be handled by augmenting (15) with terms such as  $\mathbf{v}_k t e^{it}$ , but this case will not be pursued further.

Finally, combining (10), (12) and (15) yields the explicit solution

$$\mathbf{x}(t) = e^{A_k t} \mathbf{c}_k - 2 \operatorname{Re}(\mathbf{u}_k e^{it}), \quad t \in [t_k, t_{k+1}] \quad (18)$$

with

$$\mathbf{u}_k = (A_k - iI)^{-1} \mathbf{b} \quad (19)$$

and

$$\mathbf{c}_k = e^{-A_k t_k} (\mathbf{x}_k + 2 \operatorname{Re}(\mathbf{u}_k e^{it_k})), \quad (20)$$

as follows from (18) with  $t = t_k$ . Putting  $t = t_{k+1}$  in (18) yields the value of  $\mathbf{x}(t)$  at the next jump,

$$\mathbf{x}_{k+1} = e^{A_k t_{k+1}} \mathbf{c}_k - 2 \operatorname{Re}(\mathbf{u}_k e^{it_{k+1}}). \quad (21)$$

For evaluating  $\mathbf{x}(t)$  at many points it is best to pre-compute and store  $\mathbf{u}_k$ ,  $\mathbf{c}_k$ ,  $\mathbf{x}_{k+1}$  according to Equations (19), (20) and (21) in a loop running over  $k = 0, \dots, m-1$ . Then (18) yields  $\mathbf{x}(t)$  involving at most one matrix exponential.

### 3 Periodic Solutions

In technical applications such as SVCs one is often interested in periodic solutions of the corresponding differential equations. However, periodic solutions are of practical significance only if they are attractive; then they arise naturally after a long time from an arbitrary initial state in their basin of attraction.

In linear problems such as (1) the principle of superimposition holds; therefore the stability of periodic solutions is determined by the corresponding homogeneous problem defined by  $\mathbf{p}(t) = 0$  or  $\mathbf{b} = 0$  (see Equation (7)). From Equations (19), (20), (21) with  $\mathbf{b} = 0$  and  $\mathbf{x}_k$  replaced by  $\mathbf{y}_k$  we obtain

$$\mathbf{y}_{k+1} = e^{A_k(t_{k+1}-t_k)} \mathbf{y}_k \quad (22)$$

since the matrices  $A_k t_{k+1}$  and  $A_k t_k$  commute. Here  $\mathbf{y}_k$  denotes the value of a solution of the homogeneous equation at the jump  $t_k$  ( $k = 0, \dots, m$ ). Therefore, after a full period  $t_m = 2\pi$ , the value of  $\mathbf{y}_m$  is given by the linear map

$$\mathbf{y}_m = M \mathbf{y}_0, \quad (23)$$

where

$$M = \prod_{k=0}^{m-1} e^{A_k(t_{k+1}-t_k)} \quad (24)$$

(the product taken from right to left) is the so-called monodromy matrix. There follows that a periodic solution of (1) is globally attractive if  $|\mu_j| < 1$  holds for all eigenvalues  $\mu_j$  of  $M$ .

It turns out that for the 6-pulse SVC given by the matrix (4) and the switching function (5) the eigenvalues  $\mu_j$  are independent of the shift  $\tau$ . In the example (6) the values

$$\mu_1 = 0.84311362558494 \quad (25)$$

$$\mu_{2,3} = -0.77497080502505 \pm 0.42379742896324i \quad (26)$$

are obtained; hence if a periodic solution exists it is globally attractive.

To construct such a solution  $\mathbf{x}(t) = \mathbf{x}_P(t)$ , an initial value vector  $\mathbf{x}_0 \in \mathbb{R}^n$  has to be found such that

$$\mathbf{x}_m = \mathbf{x}_0 \quad \text{or} \quad \mathbf{f}(\mathbf{x}_0) := \mathbf{x}_m - \mathbf{x}_0 = 0 \quad (27)$$

in the notation of (21). Due to the linearity of the problem, the vector valued function  $\mathbf{f}$  defined in (27) is itself linear. Hence it suffices to compute  $n + 1$  values of  $\mathbf{f}$  in order to define the system of linear equations (27). This is necessary because  $\mathbf{f}$  is defined only indirectly by means of the rather complicated algorithm described at the end of Section 2.

The  $n + 1$  points of evaluation are conveniently chosen as the origin and the  $n$  unit points  $\mathbf{e}_j = (0, \dots, 0, 1, 0, \dots, 0)^\top$ , where the non-vanishing component is in position  $j$ , ( $j = 1, \dots, n$ ). If we denote

$$\mathbf{f}_0 := \mathbf{f}(0), \quad \mathbf{f}_j := \mathbf{f}(\mathbf{e}_j), \quad (j = 1, \dots, n) \quad (28)$$

the linear function  $\mathbf{f}(\mathbf{x})$  is explicitly given by

$$\mathbf{f}(\mathbf{x}) = \mathbf{f}_0 + \sum_{j=1}^n (\mathbf{f}_j - \mathbf{f}_0) x_j \quad (29)$$

where  $\mathbf{x} = (x_1, \dots, x_n)^\top$ . The initial value  $\mathbf{x}_0$  satisfying  $\mathbf{f}(\mathbf{x}_0) = 0$  is therefore obtained from the linear system

$$F\mathbf{x}_0 = \mathbf{f}_0 \quad (30)$$

where  $F$  is the matrix

$$F = [\mathbf{f}_0 - \mathbf{f}_1, \dots, \mathbf{f}_0 - \mathbf{f}_n]. \quad (31)$$

A unique periodic solution exists if the matrix  $F$  is regular. In the numerical example (6) we obtain for all  $\tau \in [-R, R]$   $\text{cond}(F) \doteq 11.74$ , hence in this case  $F$  is far away from a singular matrix.

## 4 A MATLAB Implementation

In this section we will present a complete MATLAB program capable of carrying out the tasks listed below for the example of the 6-pulse SVC. It is based on the explicit solution of  $n = 3$  linear differential equations with piecewise constant coefficients as described in the previous sections.



- (a) The periodic solution  $\mathbf{x}_P$  discussed in Section 3 is generated, and its values at the  $m + 1$  jumps of  $A(t)$  are stored. Possible near-degeneracies may be detected by means of  $\text{cond}(F)$ .
- (b) Computation of the monodromy matrix  $M$  associated with  $\mathbf{x}_P$  together with its eigenvalues for discussing the stability of  $\mathbf{x}_P$ .
- (c) Efficient tabulation and plotting of  $\mathbf{x}_P$  (dense output).
- (d) Fourier analysis of the periodic solution  $\mathbf{x}_P$ .

The program is kept general as much as possible although some particular features of the specific example necessarily appear. This will enable the reader to adapt the program to any other problem involving linear ODEs with piecewise constant coefficients. The main objectives of the code are efficiency and simplicity, not luxury of the input and output. However, compared to the shortness of the program a fair amount of luxury and a high degree of reliability is achieved.

The core of the program is the function `f = solution(x0)` given in Algorithm 1. It solves (1) with the initial vector  $\mathbf{x}_0 = \mathbf{x}_0$  and returns the value  $\mathbf{f} = \mathbf{f}(\mathbf{x}_0)$  of the function defined in (27). According to the number `nargout` of output arguments in the actual call (a permanent variable of MATLAB) the matrices `xx`, `uu`, `M` are also computed. The matrices  $A_k$  specific to this example (see Equations (4), (5)) are generated by the function `A = matrix(k)` in Algorithm 2. Table 1 describes the variables which are passed as global parameters for convenience.

ALGORITHM 1. *Function solution*

```
function [f, xx, uu, M] = solution(x0)
SOLUTION      Solves the SVC problem over a period.
               f == 0 <=> periodic solution
global m n tt b
if (nargout > 1), uu = []; xx = x0; end;
if (nargout > 3), M = eye(n); end;
x = x0;
for k = 1:m,
    A = matrix(k-1);
    E = expm(A*(tt(k+1)-tt(k)));
    u = (A-i*eye(n))\b;
    x = E*(x+2*real(u*exp(i*tt(k))))-2*real(u*exp(i*tt(k+1)));
    if (nargout > 1), uu = [uu,u]; xx = [xx,x]; end;
    if (nargout > 3), M = E*M; end;
end;
f = x - x0;
end solution
```

ALGORITHM 2. *Function matrix*

Table 1: Description of Global Variables

Variable	Description
$m, n$	Dimensional parameters
$C, L, R$	System parameters
$\mathbf{b}$	Inhomogeneity $\mathbf{b}$ from Equations (7),(3)
$\mathbf{tt}(1:m+1)$	Array of jumps, $\mathbf{tt}(1+k) = t_k, (k = 0 \dots m)$

```
function [A] = matrix(k)
generates the matrix A[k], k=0,...,m-1
global m C L R
phi = k*2*pi/(m-1);
A = [
        0, cos(phi)/C, sin(phi)/C
    -cos(phi)/1.5/L,    -R/L,    0
    -sin(phi)/1.5/L,    0,    -R/L
];
end matrix
```

On exit from `solution`, `xx` will contain the values  $\mathbf{x}_k$ , `uu` the intermediate results  $\mathbf{u}_k$  and `M` the monodromy matrix. These variables are initialized in the first lines of `solution`. In the subsequent loop over  $k$  we first generate the matrix  $A_{k-1}$  by calling the function `matrix`. Then Equations (19), (20) and (21) are evaluated as described at the end of Section 2. The only modification is that  $\mathbf{c}_k$  in (21) has been substituted by the expression in (20). This halves the number of matrix exponentials to be computed. Furthermore, it turns out that storing the vectors  $\mathbf{c}_k$  can be avoided at almost no cost.

ALGORITHM 3. *Script per*

```

global m n C L R tt b

m = 7; n = 3;
C = 0.2; L = 0.15; R = 0.005;
b = [0;1;-i]/2/L;
N = 512;

tau = input('tau = ');
tt = [0, 2*pi/(m-1)*[1/2:(m-3/2)]+tau, 2*pi];

f0 = solution(zeros(n,1));
F = []; for x = eye(n),
    F = [F, - solution(x) + f0];
end;
x0 = F f0;
[ff, xx, uu, M] = solution(x0);

delta = 2*pi/N;
k = 0; xtab = [];
for j = 0:N-1,
    t = j*delta;
    if (tt(k+1) <= t),
        k = k+1;
        A = matrix(k-1);
        aux = expm(A*(t-tt(k)))*(xx(:,k)+2*real(uu(:,k)*exp(i*tt(k))));
        E = expm(A*delta);
    else
        aux = E*aux;
    end;
    xtab = [xtab, aux-2*real(uu(:,k)*exp(i*t))];
end;

figure(1); plot(xtab(1,:))';
figure(2); plot(xtab(2:3,:))';
cc = fft(1/N*xtab'); cc(1:32,:)

```

Since the MATLAB indices are  $\geq 1$  the loop index  $k$  has been shifted by 1. The newly computed vectors  $\mathbf{u} = \mathbf{u}_k$ ,  $\mathbf{x} = \mathbf{x}_{k+1}$  are stored by appending them to the matrices  $\mathbf{u}\mathbf{u}$  and  $\mathbf{x}\mathbf{x}$ , respectively. In the same loop the partial product  $\mathbf{M}$  is updated according to (24). The final statement defines the function value (27).

In Algorithm 3 the main program `per.m` performing the tasks (a) through (d) by means of calls to the function `solution` is given. After initializing some global variables, the input of the shift  $\tau$  is done via a program request. In this way the sensitive dependence of the circuit's behavior on  $\tau$  may easily be studied. In the next statement the array of the jumps `tt(1:m+1)` (with the index shifted by 1) is defined:

$$\begin{aligned} \mathbf{tt}(1) &= 0 \\ \mathbf{tt}(1 + \mathbf{k}) &= t_k = \tau + (k - \frac{1}{2})\frac{\pi}{3}, \quad (k = 1, \dots, 6) \\ \mathbf{tt}(8) &= 2\pi \end{aligned}$$

(see (5)). Then the periodic solution  $\mathbf{x}_P$  together with its initial value  $\mathbf{x}_0 = \mathbf{x}_0$  is computed. The program closely follows Equations (28) through (31) and is self-explanatory.

In the next section of the program  $\mathbf{x}_P(t)$  is tabulated with step  $\Delta = 2\pi/N$ :

$$\mathbf{x}\mathbf{tab}(:, j) = \mathbf{x}_P((j - 1)\Delta), \quad j = 1, \dots, N.$$

In view of the subsequent Fourier analysis  $N$  must be a power of 2. The algorithm is organized to work efficiently if  $N \gg m$  as follows. Given a value of  $t$ , the index  $k$  is such that  $t \in [t_k, t_{k+1})$ . If  $t \in [t_k, t_{k+1})$  is the first evaluation point in this interval,  $\mathbf{x}_P(t)$  must be calculated according to Equation (18). Hence we begin by computing the first term of (18) as

$$\mathbf{aux} := e^{A_k t} \mathbf{c}_k = e^{A_k(t-t_k)} (\mathbf{x}_k + 2 \operatorname{Re}(\mathbf{u}_k e^{it_k})),$$

where the vectors  $\mathbf{x}_k$  and  $\mathbf{u}_k$  are taken from the arrays  $\mathbf{x}\mathbf{x}$  and  $\mathbf{u}\mathbf{u}$ , respectively. Furthermore, the matrix  $\mathbf{E} := e^{A_k \Delta}$  is computed and stored at this point, thus avoiding its frequent re-computation at later points in the same interval. Otherwise, it suffices to update  $\mathbf{aux}$  as  $\mathbf{aux} := \mathbf{E} * \mathbf{aux}$ .

In the final section of the program the first component of  $\mathbf{x}_P(t)$ , i.e.  $U(t)$  (the voltage across the capacitance  $C$ ) is plotted versus  $j = 1 + t/\Delta$  as MATLAB Figure 1. The second and third components, which are both variable currents, are simultaneously plotted as MATLAB Figure 2. Finally, the 3 components of  $\mathbf{x}_P$  are separately Fourier-analyzed by means of the MATLAB command `fft` that requires a column vector as its argument. The matrix `cc` serves for printing the first 32 complex Fourier coefficients of the three components of  $\mathbf{x}(t)$ .

The data (6) in the three cases  $\tau = -R$ ,  $\tau = -0.375R$  and  $\tau = R$  produce the plots presented in Figure 2. The sensitivity to small changes in  $\tau$  is obvious. Ideally, the currents  $x_2(t)$ ,  $x_3(t)$  should be sinusoidal. One goal of research in this field is to reduce the disturbances due to the higher harmonics present in the periodic solution  $\mathbf{x}_P$ .

circpowe-fig-2.eps  
0 × 0 mm

Figure 2: Voltage and currents in an SVC

Parameters:  $C = 0.2$ ,  $L = 0.15$ ,  $R = 0.005$  according to Equation (6). The three cases of the shifts  $\tau = -R$ ,  $-0.375R$ ,  $R$  are shown. In the left-hand figures the voltage  $x_1(t) = U(t)$  is plotted versus time  $t$ , whereas the currents  $x_2(t) = I_S(t)$  (solid line) and  $x_3(t) = (I_S(t) - I_T(t))/\sqrt{3}$  (dashed line) are plotted in the right-hand figures.

## 5 Conclusions

A computer simulation of a technical process is a research tool useful in designing and optimizing the process. The above specialized circuit simulator is more than ten times faster than a general-purpose simulator on the same problem, and it produces highly accurate approximations (14 decimals) to the exact solutions of the mathematical model. This enables the user to obtain reliable spectra of periodic solutions to high order, which, in turn, enable the designer to eliminate some of the unwanted harmonics.

Clearly, the success of this simulation is largely due to the high standards of the MATLAB software. Best results are obtained, however, if good software is combined with a careful mathematical analysis.

The author is indebted to Gerald Scheuer of the Institute of Power Electronics (ETH Zürich) for providing the differential equations of the SVC. Helpful comments by Rolf Strebel of the Institute of Scientific Computation (ETH Zürich) are gratefully acknowledged.

## References

- [1] M. BRAUN, *Differential Equations and their Applications*, Fourth ed., Springer, New York, 1993, 578 pp.
- [2] M. MEYER, *Leistungselektronik*, Springer, 1990, 349 pp.
- [3] T. J. E. MILLER, *Reactive Power Control in Electric Systems*, J. Wiley and Sons, New York, 1982, 381 pp.
- [4] C.B. MOLER AND C.F. VAN LOAN, *Nineteen Dubious Ways to Compute the Exponential of a Matrix*, SIAM Review 20, 1978, pp. 801-836.

# Research Reports

No.	Authors	Title
94-13	J. Waldvogel	Circuits in Power Electronics
94-12	A. Williams, K. Burrage	A parallel implementation of a deflation algorithm for systems of linear equations
94-11	N. Botta, R. Jeltsch	A numerical method for unsteady flows
94-10	M. Rezny	Parallel Implementation of ADVISE on the Intel Paragon
94-09	M.D. Buhmann, A. Le Méhauté	Knot removal with radial function interpolation
94-08	M.D. Buhmann	Pre-wavelets on scattered knots and from radial function spaces: A review
94-07	P. Klingenstein	Hyperbolic conservation laws with source terms: Errors of the shock location
94-06	M.D. Buhmann	Multiquadric Pre-Wavelets on Non-Equally Spaced Centres
94-05	K. Burrage, A. Williams, J. Erhel, B. Pohl	The implementation of a Generalized Cross Validation algorithm using deflation techniques for linear systems
94-04	J. Erhel, K. Burrage, B. Pohl	Restarted GMRES preconditioned by deflation
94-03	L. Lau, M. Rezny, J. Belward, K. Burrage, B. Pohl	ADVISE - Agricultural Developmental Visualisation Interactive Software Environment
94-02	K. Burrage, J. Erhel, B. Pohl	A deflation technique for linear systems of equations
94-01	R. Sperb	An alternative to Ewald sums
93-07	R. Sperb	Isoperimetric Inequalities in a Boundary Value Problem in an Unbounded Domain
93-06	R. Sperb	Extension and simple Proof of Lekner's Summation Formula for Coulomb Forces
93-05	A. Frommer, B. Pohl	A Comparison Result for Multisplittings Based on Overlapping Blocks and its Application to Waveform Relaxation Methods
93-04	M. Pirovino	The Inverse Sturm-Liouville Problem and Finite Differences
93-03	R. Jeltsch, X. Wang	Uniqueness of Piecewise Lipschitz Continuous Solutions of the Cauchy-Problem for $2 \times 2$ Conservation Laws
93-02	W.-A. Yong	Difference approximations to the global $W^{1,\infty}$ solutions of the isentropic gas equations