Time domain analysis of linear systems using state space approach

Viera Biolková* and Dalibor Biolek**
* Dept. of Radioelectronics, Technical University Brno, Antonínská 1, 662 09 Brno, Czech Republic
biolkova@urel.fee.vutbr.cz
** Dept. of El. Engineering, VA Brno, Kounicova 65, PS13, 612 00 Brno, Czech Republic
biolek@cs.vabo.cz

Abstract

An accurate and fast method of time domain analysis of system from its transfer function is proposed. This method is based on the efficient solution of state equation of the corresponding dynamical system in the first canonical form. Algorithm also provides stable solution in case of periodical or divergent responses.

1 Introduction

During the computer supported design of selective networks as continuous-time filters, the task to compute time response from its Laplace transformation often appears. In case of linear lumped networks, this Laplace transformation is in the form of rational fractional function. The problem of Laplace inversion of rational fractional transfer function is well described in the literature. We can mention primitive *Zakian's* solution [1], *Vlach-Singhal's* method of numerical inversion [2] (further *V-S* method), *Valsa's* method [3], method of Laplace transform replacement by *z*-transform [4], semi-symbolical time-domain analysis using partial fraction expansion [5] etc. Last mentioned approach provides analytical expressions of time response. However, its accuracy strongly depends on the accuracy how to find all poles of Laplace transform. In case of multiple poles or poles of close values, this method can fail. Other mentioned methods are not based on the pole computation, but their accuracy droops with the increased simulation time. This problem can be usually overcome by so-called „*resetting principle*“ [2]. The *V-S* method completed by resetting principle is powerful. Their certain disadvantages are as follows: memory-exacting algorithm based on complex arithmetic and relatively long-time computation in case of high demanded accuracy.

In case we do not intend to utilize algorithms of time domain analysis directly in own design software, for instance in the iterative loops, we can consider to use well-known analog simulators as *SPICE* or *MICROCAP*. However, these programs are not directly established to work with transfer functions of linearized circuits. Nevertheless, we can use controlled Laplace sources in the programs *MC3* and *MC4*. The time analysis of these circuits is performed in following way: The Laplace operator s is replaced by complex frequency $j2\pi F$ to obtain frequency transfer; then the *FFT* is applied to calculate the samples of circuit impulse response. The required time response is obtained using convolution of impulse response and input signal.

Simulation results are distorted by the error caused by the „*FFT* window effect“, because only the limited number of samples of frequency transfer is available for the impulse response computation. Likewise, only limited (and often „non-representative“) number of samples of impulse response is available for response computation. These imperfections can be observed especially in the time intervals et the end of simulation. The simulation results are quite invalid if the Laplace transform contains poles in the ride - side complex plain.

The method of time - domain analysis developed for the design system *NAF* [6] will be introduced in this paper. This method is based on „*resetting principle*“. However, the computational algorithm is not so complicated as in the *V-S* method. In this way, the accelerating of calculation is reached with contemporaneous accuracy preserving. The memory claims are minimum. The described method provides correct results independently of the pole location.

2 Basis of the method

Consider that the Laplace transform of signal is contemporaneously the transfer function of some linear system. Than the time - domain representation of Laplace transform is the impulse response this system. Let us denote the system input/output as $v(t)\,/\,y(t)$ and their Laplace transforms as $V(s)\,/\,Y(s)$, respectively. Consider following transfer function:

$$K(s)=\frac{Y(s)}{V(s)}=\frac{\sum_{i=0}^{n} a_i s^i}{s^n+\sum_{i=0}^{n-1} b_i s^i}, n \geq 1. \tag{1}$$

The coefficients of transfer function are known. We search the zero - state response $y(t)$ to the Dirac impulse.

It is possible to adjoin infinite number of n - order systems with different inside structures to the transfer function (1). We choose classical first canonical structure according to the Fig.1.

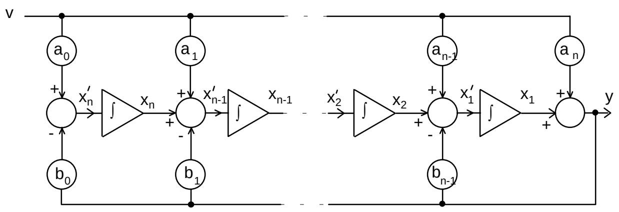

Figure 1: First canonical structure of the dynamical n-order system with transfer function (1).

Selecting output signals of integrators as state variables, the state equations will be in the well-known form

$$\frac{d}{dt}\begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_{n-1} \\ x_n \end{bmatrix} = \underbrace{\begin{bmatrix} -b_{n-1} & 1 & 0 & \cdots & 0 \\ -b_{n-2} & 0 & 1 & \cdots & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ -b_1 & 0 & 0 & \cdots & 1 \\ -b_0 & 0 & 0 & \cdots & 0 \end{bmatrix}}_{\mathbf{A}} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_{n-1} \\ x_n \end{bmatrix} + \underbrace{\begin{bmatrix} a_{n-1}-a_n b_{n-1} \\ a_{n-2}-a_n b_{n-2} \\ \vdots \\ a_1-a_n b_1 \\ a_0-a_n b_0 \end{bmatrix}}_{\mathbf{B}} v$$

$$y = x_1 + a_n v. \tag{2}$$

The main idea how to compute time response from Laplace transform is as follows:

1) Inversion of Laplace transform (1) is replaced by the problem how to find the impulse response of the system in Fig.1.
2) The impulse response will be obtained as the solution of state equations (2).

Aforementioned procedure is also used in the *V-S* method. However, the contents of both items will be different in our method. In the *V-S* method, the second canonical form is used because it is advantageous for the implementation of Laplace inversion algorithm in item **2**. Our solution of state equations does not utilize Laplace inversion and the first canonical structure leads to the state equations in more suitable form for our purposes.

3 Solution of the state equations

The state equations (2) have well-known solution

$$\mathbf{x}(t)=e^{\mathbf{A}t}\mathbf{B}*v(t)+e^{\mathbf{A}t}\mathbf{x}(0), y(t)=x_1(t)+a_n v(t), \tag{3}$$

where the symbol * denotes convolution.

Considering computation of impulse response, i.e. $v(t)=\delta(t)$, $\mathbf{x}(0)=\mathbf{0}$ and utilizing „*resetting principle*“, the vector $\mathbf{x}(t)$ can be computed in the single time points $t_1 < t_2 < .. < t_k < ..$ with the constant step Δt as follows:

$$\mathbf{x}(t_1)=e^{\mathbf{A}\Delta t}\mathbf{B} \qquad ; v(t)=\delta(t); \mathbf{x}(0)=0$$
$$\mathbf{x}(t_k)=e^{\mathbf{A}\Delta t}\mathbf{x}(t_{k-1}), k>1 \quad ; v(t)=0; \mathbf{x}(0)\rightarrow \mathbf{x}(t_{k-1})$$

It should be noted that there is sufficient to compute matrix $e^{\mathbf{A}\Delta t}$ once for all and to calculate state vector in the k-th step from ones in previous step. The required response in given step is then first coordinate of the state vector (see Eq.3):

$$y(t_k)=x_1(t_k), k>0.$$

It is also necessary to determine initial value $y(0)$. As follows from Fig.1 and Eq. (2), the input Dirac impulse penetrates into the output with the scale a_n. The important value is ride - side limit $y(0^+)$. The simple computation from the transfer function leads to the conclusion

$$y(0^+)=a_{n-1}-a_n b_{n-1} \tag{4}$$

4 Algorithm

The algorithm of time - domain representation of Laplace transform (1) with constant time step can be described as follows:

```
begin
   INPUT;      {entering of vectors NUM and DEN
                 of coefficients of the transfer
                 function, order n, step DELTA and
                 number of steps NS}
   MATRIX_COMPILATION;
               {compilation of matrix A and vector
                 B from the coefficients of the
                 transfer function}
   EXPMAT(A,DELTA,);
               {computation of matrix
                 exp(A.DELTA), result is stored in
                 A}
   if (num[n] >< 0) then
            writeln('there is Dirac impulse in the
            origin with strength of',num[n]);
   Y[0]:= num[n-1]-num[n]*den[n-1];
   X:=B;
               {initiate vector X}
```

```
    for k:= 1 to NS do
            begin
                MATVECMULTI(A,X);
                  { multiplication A by X;
                    result is stored in X}
                Y[k]:= X[1];
            end;
end.
```

Algorithm of the matrix **A** and vector **B** compilation is obvious of Eq. (2). Computation of matrix $e^{\mathbf{A}\Delta t}$ is most time consuming procedure but this procedure is called only once in case of constant time step. The sum of finite number M of Taylor set elements of exponential matrix function belongs to the often used methods:

$$e^{\mathbf{A}\Delta t} \approx \sum_{k=0}^{M} \frac{(\mathbf{A}\Delta t)^k}{k!} \tag{5}$$

After each addition, the test of certain error criterion is performed, e.g. relative difference between norms of computed matrix in succeeding steps must be less then demanded error.

Utilizing special form of matrix **A**, computation of matrix function (5) can be accelerated. Matrix **A** (see Eq.2) can be decompose to the sum of matrices $\mathbf{A}_1$ and $\mathbf{A}_2$:

$$\underbrace{\begin{bmatrix} -b_{n-1} & 1 & 0 & \cdots & 0 \\ -b_{n-2} & 0 & 1 & \cdots & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ -b_1 & 0 & 0 & \cdots & 1 \\ -b_0 & 0 & 0 & \cdots & 0 \end{bmatrix}}_{\mathbf{A}} =$$

$$= \underbrace{\begin{bmatrix} 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & 0 & \cdots & 1 \\ 0 & 0 & 0 & \cdots & 0 \end{bmatrix}}_{\mathbf{A}_1} + \underbrace{\begin{bmatrix} -b_{n-1} & 0 & 0 & \cdots & 0 \\ -b_{n-2} & 0 & 0 & \cdots & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ -b_1 & 0 & 0 & \cdots & 0 \\ -b_0 & 0 & 0 & \cdots & 0 \end{bmatrix}}_{\mathbf{A}_2}$$

Computation of k-th power of matrix $\mathbf{A}\Delta t$ is realized using the precalculated k-1-th power:

$$\frac{(\mathbf{A}\Delta t)^k}{k!} = \frac{(\mathbf{A}\Delta t)^{k-1}}{(k-1)!}\frac{\mathbf{A}\Delta t}{k} = \mathbf{M}_{k-1}\frac{\mathbf{A}\Delta t}{k}, \tag{6}$$

where the symbol $\mathbf{M}_{k-1}$ designates auxiliary matrix stored in memory. To utilize (6) we can write

$$\frac{(\mathbf{A}\Delta t)^k}{k!} = \mathbf{M}_{k-1}(\mathbf{A}_1 + \mathbf{A}_2)\frac{\Delta t}{k} = (\mathbf{M}_{k-1}\mathbf{A}_1 + \mathbf{M}_{k-1}\mathbf{A}_2)\frac{\Delta t}{k}.$$

After multiplication of matrices $\mathbf{M}_{k-1}$ and $\mathbf{A}_1$, the resulting matrix arises which columns are columns of the original matrix $\mathbf{M}_{k-1}$ shifted to one position to the right. The first column of resulting matrix will contain zeros.

After multiplication of matrices $\mathbf{M}_{k-1}$ and $\mathbf{A}_2$, the resulting matrix arises which all columns except first one are zero. Element in the m-th row of first column is equal to the sum

$$\sum_{i=1}^{n} \mathbf{M}_{k-1}[m,i]\mathbf{A}_2[i,1] = -\sum_{i=1}^{n} \mathbf{M}_{k-1}[m,i]b_{N-i}. \tag{7}$$

We can now formulate algorithm how to compute k-th element of Taylor set (5):

Shifting of columns No. 1 to n-1 of matrix $\mathbf{M}_{k-1}$ to one position to the right.
Compilation of first column according to (7).
Multiplication of all elements by the expression $\Delta t / k$.

The classical multiplication of two matrices is performed using n^3 multiplication and n^3 additions. (6) Our procedure is performed using shifting of n-1 columns and n^2 multiplication and n^2 additions. If the matrix is declared as the array of columns, then the shifting of columns can be realized using their readressing.

5 Algorithm testing

Algorithm had been tested wit the aim to compare its accuracy and speed with the *V-S* method. The tests were passed on the Laplace

method	average relative error in % of correct value	computational time [s] on PC AT 486DX2/66
presented	10^{-7}	0,5
V-S, 10 poles	10^{-6}	3,73
V-S, 4 poles	1	2,14
V-S, 2 poles	100 *)	1,59

*) Invalid due to numerical failure of *V-S* method

Tab.1. Test results

transforms with multiple poles and on the transform of periodical signals, particularly harmonic ones. The results are listed in Tab.1. Besides our described method, three variants of *V-S* methods were tested with various number of poles of *Pade approximating function* (see [2]). The exacting inversion of Laplace transform

$$s/\left[s^2 + (2\pi.1000)^2\right] \mathrel{\hat{=}} \cos(2\pi.1000t)$$

was performed with the time step 0,1ms in 10000 steps. This task means the computation of time response over its 1000 periods. As noted from Tab.1, our algorithm coped with this task with the ordinally less error then the most precise *V-S* method and seven times faster. The fastest *V-S* method with two poles of *Pade* approximating function was three times slower. However, its results are invalid due to numerical failure of *V-S* method.

6 Conclusion

The efficient method of computation of the signal samples from its Laplace transform in the form of rational fractional function is introduced. This method provides correct results for arbitrary location of poles of the Laplace transform. It is also very stable during the computation of periodical signals over a lot of periods. Test results proved better speed- and accuracy parameters than for *V-S* method.

Described method is based on the solution of state equations of the system in first canonical form. This system is adjoined the transfer function. This fact enables to utilize given method also for the computation of periodical steady state in linear systems on the assumption that the period of input signal is known [7].

The algorithm has been incorporated to the program *NAF* for the design of frequency filters [6] instead of so far utilized algorithm based on the *V-S* method.

Acknowledgement

This work is supported by the Grant Agency of the Czech Republic under grant No. 102/94/1080.

References

[1] ZAKIAN,V.: Optimization of Numerical Inversion of Laplace Transforms. Electronics Letters, 6, 1970, No.21, pp. 677-679.

[2] VLACH,J.-SINGHAL,K.: Computer Methods for Circuit Analysis and Design. Van Nostrand Reinhold, New York, 1983.

[3] VALSA,J.: Jedna metoda numerického výpočtu originálu *f*(*t*) z Laplaceova obrazu *F*(*p*). Technika elektrických strojů - teoretické číslo, 1975, pp.9-18. Czechoslovakia.

[4] MAYER,D.-ULRYCH,B.: A Contribution to Numerical Analysis of Electric Networks in Periodical Non-Sinusoidal Steady State. Electrical Engineering Journal, 41, 1990, No.10, pp.745-753.

[5] WESTREICH,D.: Partial Fraction Expansion without Derivative Evaluation. IEEE Trans. on CAS, Vol.38, No.6, June 1991, pp.658-660.

[6] BIOLEK,D.-HÁJEK,K.-SEDLÁČEK,J.: Analog Filter Design with User's Optimization by Personal Computer. XXIV GA URSI, Kyoto 1993, Abstracts p.693. Japan.

[7] BIOLEK,D.: Time domain analysis of linear systems using Laplace inversion. Radioengineering, No.1, 1994, pp.17-20.