EFFICIENT ALGORITHMS OF DIRECT AND INVERSE FIRST-ORDER S-Z TRANSFORMATIONS

Karel ZAPLATÍLEK[1], Michal LARES[2]
[1]Department of Electrical Engineering and Electronics
Brno Military Academy
Kounicova 65, 612 00 Brno, Czech Republic
[2] Department of Telecommunications
Brno University of Technology
Purkyňova 118, 612 00 Brno, Czech Republic

Abstract

In the article, we describe principles of numerical algorithms by means of which coefficients of continuous-time and discrete-time frequency filters are possible to transform each other for several s-z transformations. The basis of algorithms is so-called Pascal matrix for calculation of which an original procedure is used. We discuss problems of numerical conditionality of Pascal matrix and possibility of its practical usage. Illustrative examples of numerical calculations programmed in Matlab 5.1 are the constituent part of the article.

Keywords

Continuous-time and discrete-time linear filters, Pascal matrix, s-z transformations, numerical conditionality

1. Introduction

Coefficients of continuous-time filters and discrete-time filters are possible to transform each other with application of some known s-z transformations. Transformation problems of analog frequency filters´ coefficients from the *s-plane* to the *z-plane* are not new problems. Application of s-z transformations is widely cited in the 60´s and 70´s [1], [8]. We also put attention to this problem [4], [12]. The design of discrete, above all numerical, filters on the basis of analog prototypes is one of possible design methods [10]. Advantages and disadvantages of such procedure were published many times.

Contemporary state can be characterized in this way:

- sufficiently elaborated and mastered methodology of discrete-time filters` design on the basis of analog prototypes,
- mapped out features of basic types of s-z transformations, especially linear first-order transformations,
- developed numerical algorithms for transformation of filter coefficients in the s→z direction namely for bilinear transformation (BL),
- recently came out articles devoted to the basis of coefficients` transformation in the s←z direction,
- developed numerical algorithms for compilation of Pascal matrix for bilinear transformation, newly also for some other transformations.

Thanks to thorough study of literature, it became evident that for effective numerical transformation of coefficients of frequency filters in the both directions (s→z and s←z) already existing procedures and algorithms will have to be unified and non-solved parts - work out. Works described in this article were intended on these tasks:

- develop efficient and exact algorithm composition of Pascal matrix for several s-z transformations and for both design directions (derive relations for inversion of Pascal matrix),
- in the abstract, work out methodology of calculation of objective coefficients for both design directions with regard to possibility to transform both the structures IIR and FIR in the direction s ← z,
- developed algorithms complete with algorithms of calculation of all frequency characteristics for prototypes and equivalent structures as a constituent part of project methods,
- work out methodology by means of which will be possible to compare results with features of filters, designed by classical procedures in the prototype level,
- on the basis of result - determine possible restrictions with application of developed procedures in practice.

2. Pascal matrix

Bilateral relationship between vectors of continuous-time filter coefficients and its discrete equivalent is presented by equation (1). The relationship was derived in [9], however only for BL transformation

$$\mathbf{V}_D = \mathbf{P}_M . \tilde{\mathbf{V}}_A , \qquad (1)$$

where $\tilde{\mathbf{V}}_A = [a_0\ ca_1\ c^2 a_2\ .....\ c^N a_N]^T$ is vector of modified coefficients of analog prototype, $\mathbf{V}_D = [d_0\ d_1\ d_2\ .....\ d_N]^T$ is vector of coefficients of discrete equivalent, $\mathbf{P}_M$ is Pascal matrix and N is order of filter. Name of matrix was introduced in [6].

Modification of vector $\tilde{\mathbf{V}}_A$ consists in multiplication of element of vector a_k by constant c^k, where value c depends on chosen s-z transformation. Pascal matrix is square matrix of order $(N+1) \times (N+1)$.

Thanks to relation (1), we can efficiently numerically transform coefficients from the *s-plane* to the *z-plane*. For reverse procedure, that means - calculation of coefficients′ vector of continuous-time filter on the basis of discrete-time prototype, it is possible to use in two ways:

1. to make simple numerical inversion of Pascal matrix in relation (1) with following calculation of coefficients,
2. to try to derive of similar relation as (1) but for procedure in s←z direction.

The first procedure is simple, if we use program systems of Matlab type [7], MathCad, etc. Disadvantage could be (potentially) bad numerical conditionality of inverse matrix, first of all for higher orders of prototypes. We proved that with application of the second way it is possible to gain the relation

$$\tilde{\mathbf{V}}_A' = \mathbf{P}_M' . \mathbf{V}_D . \qquad (2)$$

The meaning of symbols in (2) is analogic to eqn. (1). However, modification of vector of coefficients of analog equivalent is changed, $\tilde{\mathbf{V}}_A' = [c^{-N} a_0\ c^{-(N-1)} a_1\ .....\ a_N]^T$.

Matrix $\mathbf{P}_M'$ has similar function as Pascal matrix. Tab.1 summarizes relations between matrices $\mathbf{P}_M'$ and $\mathbf{P}_M$ in eqn. (1) and (2) for selected types of linear first-order s-z transformations [4].

Type of s-z transformation		
bilinear (BL) $s = 2 f_{SAM} \frac{z-1}{z+1}$	$\mathbf{P}_M' = \mathbf{P}_M$	$c = 2 f_{SAM}$
backward-differ. (BD) $s = f_{SAM}\left(1 - z^{-1}\right)$	$\mathbf{P}_M' = \mathbf{P}_M$	$c = f_{SAM}$
forward-differ. (FD) $s = f_{SAM}\left(z - 1\right)$	$\mathbf{P}_M' = inv\ \mathbf{P}_M$	$c = f_{SAM}$
parametric (BD-BL) $s = \frac{1+r}{T_{SAM}} \frac{z-1}{z+r}$	$\mathbf{P}_M' = \mathbf{P}_M$	$c = \frac{1+r}{T_{SAM}},\ r \subset \langle 0,1 \rangle$

Tab. 1 Comparison and meaning of some quantities in eqn. (1) and (2)

Looking on Tab. 1, it is evident, that with application of the second method it is possible to reduce the number of operations. Except transformation FD, algorithm for composition of Pascal matrix is possible to apply to both projected directions without necessity of carry out its inversion. Constant c is proportional to the sampling frequency f_{SAM} and the relations for its calculation are very simple maybe except of parametric transformation BD-BL [4].

In the past, several procedures for proper numerical calculation of Pascal matrix were worked out. As we didn't have at disposal their descriptions, we worked out our own efficient exact algorithm, which base is Pascal's triangle. On the Fig. 1, we can see Pascal matrix $\mathbf{P}_M$ for the second-order filter together with calculated coefficient of numerical conditionality C_C, for which calculation procedure defined in [11] was used. In the case of transformation BD-BL, numerical value of coefficient $r = 0.5$ is introduced.

$$\mathbf{P}_{M(BL)} = \begin{bmatrix} 1 & 1 & 1 \\ 2 & 0 & -2 \\ 1 & -1 & 1 \end{bmatrix} \qquad \mathbf{P}_{M(BD)} = \begin{bmatrix} 1 & 1 & 1 \\ 0 & -1 & -2 \\ 0 & 0 & 1 \end{bmatrix}$$

$$C_{C(BL)} = 1 \qquad C_{C(BD)} = 4$$

$$\mathbf{P}_{M(FD)} = \begin{bmatrix} 1 & 1 & 1 \\ 2 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix} \qquad \mathbf{P}_{M(BDBL,0.5)} = \begin{bmatrix} 1 & 1 & 1 \\ 1 & -0.5 & -2 \\ 0.25 & -0.5 & 1 \end{bmatrix}$$

$$C_{C(FD)} = 4 \qquad C_{C(BD-BL)} = 1.7778$$

Fig. 1 Pascal matrix of the second-order continuous-time filter and its coefficient of numerical conditionality

For effective calculation of Pascal matrix, we made original functions (m-files) in Matlab 5.1. Syntax of calling such function for example of parametric BD-BL transformation follows:

$$[P_M, C_C] = \text{pascalbdbl}(\ N, r)$$

Input parameters are filter order and coefficient BDBL, output parameters then mentioned Pascal matrix together with coefficient of numerical conditionality C_C.

2.1 Numerical conditionality of Pascal matrix

After application of direct s-z transformations, the order of analog prototypes is quite low, rarely are used filters of higher then tenth or twelfth order. If there is necessity of transformation of coefficients from *z-plane* to *s-plane*, then orders of discrete-time prototypes can be much higher. That is why in case of inverse s-z transformations should be paid attention to numerical conditionality of Pascal matrix, because Pascal matrix in the high degree influences size of numerical errors. If the above described functions are used for calculation of Pascal matrix, it is easy to gain spread of coefficients of numerical conditionality of this matrix depending on the prototypes order, see Fig. 2.

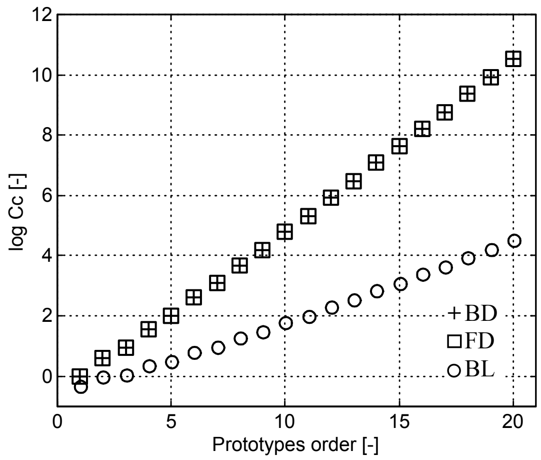

Fig. 2 Numerical conditionality of Pascal matrix

If coefficient of numerical conditionality of square matrix assumes order of units, it is possible to consider this matrix as well numerical conditioned one [11]. Fig. 2 demonstrates clear differences between s-z transformations. The best result gives transformation BL, where the boundary ones are considered orders of discrete-time prototypes about five at BL and about two at BD and FD. This claim, however, could limit possible range of prototypes. Fortunately, in practice it is proved that e.g. for BL filters of twelfth-order and more are possible to transform precisely without making any significant numerical mistakes.

The details, connected with composing of Pascal matrix for various s-z transformations, are possible to look up in the recent publications, e.g. [13], [14]. Derivation of relations for calculation of its elements is not completely trivial and is based on binomial expansions. Functions enabling calculation of Pascal matrix we developed also for transformation LDI [3], however with regard to the character of this transformation (doubles prototype´s order and sampling frequency) we do not give details. According to us, in the future, bigger attention should be paid to working out of features of this transformation.

3. Transformation of coefficients

If Pascal matrix is calculated for given linear system and for chosen s-z transformation, according to eqn. (1), resp. (2) it is easy to gain numerical form of resulting coefficients. Calculation should be done separately for numerator and denominator of transfer (system) function of the prototype. For these calculations were also created original algorithms in Matlab 5.1 environment. Partial functions for the calculation of Pascal matrix and coefficient of conditionality are parts of them. Algorithm was named *PZ*.

Example 1

The syntax of calling *PZ* program follows:

$$[A,B] = \text{pz}([10000\ \ 0\ \ 1],[10000\ \ 10\ \ 1],2)$$

Parentheses on the right side of the expression specify input parameters, that means list of coefficients of numerator and denominator and its order. The prototype is continuous-time second-order filter with transfer function

$$K(s) = \frac{s^2 + 10000}{s^2 + 10s + 10000}$$

It is band-stop filter with parameters ω_0 = 100 rad/s, and Q = 10. Output parameters of the whole process are 2 matrices A and B, which clearly demonstrate resulting coefficients of equivalent either directly calculated (matrix A) or normalized (matrix B). Yet, at the same time normalization is made in both directions of transform differently, as it is common in the theory of continuous-time filters and discrete-time filters. Tab. 2 demonstrates matrix B as an output of *PZ* program. During running of the function the user is appealed to introduce several parameters, they are clearly demonstrated in the block diagram below. In this case there was chosen direction s→z (prototype is continuous-time system), sampling frequency was f_{SAM} = 10 kHz and coefficient of BD-BL transformation is r = 0.5.

analog prototype		BL equivalent	
10000	10000	0.9995	1
0	10	-1.9989	-1.9989
1	1	0.9995	0.9900

BD equivalent		FD equivalent		BD-BL equival.	
0.999	1	1	1	0.9993	1
-1.9978	-1.9988	-2	-1.999	-1.9985	-1.9989
0.9989	0.9989	1	0.9991	-0.9993	0.9989

Tab. 2 Example of the output of *PZ* program

From Tab. 2, we can easily way put down semi-symbolic form of the system functions of discrete equivalents, e.g. for transformation FD we get

$$H(z) = \frac{1 - 2z^{-1} + z^{-2}}{1 - 1.999z^{-1} + 0.9991z^{-2}} .$$

Coefficients are inserted in the columns of A and B matrices with increasing power of reciprocal value of operator z.

Example 2

Function *PZ* can be used for reverse procedure of transformation, i.e. in the direction s←z. So the prototype is discrete-time system, which can be designed in Matlab:

$$[a,b] = \text{cheby1}(\ 8,\ 1,\ 0.068)$$
$$[A,B] = \text{pz}(\ a,\ b,\ 8)$$

Using built-in function *cheby1*, a discrete-time filter of the

discrete-time prototype		BL continuous-time equivalent	
2.37614818E-10	1.00000000E+00	1.55723247E+32	1.74724369E+35
1.90091855E-09	-7.71386074E+00	0	5.07711122E+31
6.65321491E-09	2.61230547E+01	0	9.87711580E+27
1.33064298E-08	-5.07249582E+01	0	8.70962737E+23
1.66330373E-08	6.17685417E+01	0	8.81003176E+19
1.33064298E-08	-4.83000866E+01	0	3.70182075E+15
6.65321491E-09	2.36838922E+01	0	2.52708490E+11
1.90091855E-09	-6.65820130E+00	0	4.47348072E+06
2.37614818E-10	8.21618305E-01	0	2.29794214E+02

Tab. 3 Part of matrix of output coefficients of Example 2

8^{th} order with Chebyshev approximation and ripple -1 dB in pass-band and f_{CAT} = 3.4 kHz is designed. A sampling frequency was f_{SAM} = 100 kHz. The standardized cut-off frequency was determined according to the relation

$$f_{CATNORM} = \frac{f_{CAT}}{0{,}5 f_{SAM}} = 0{,}068\,[-].$$

Coefficients of system function of the prototype are inserted in vectors *a* and *b* and serve as introductory parameters of the function *PZ*. Part of the output coefficient matrix of continuous-time equivalent is demonstrated in Tab. 3.

4. Possibilities of developed algorithms

Examples mentioned above illustrate application of some functions of developed algorithms. Fig. 3 demonstrates simplified developing diagram of *PZ* program. Advantages of this program are:

- rather high accuracy and stability of all functions,
- testing of algorithms also for filters of higher orders,
- simple user's interface,
- outputs in the textual and graphic form,
- open architecture, possibility to complete with other required functions,
- easy program implementation into another programs, and functions, created within Matlab,
- all partial functions are completed with helps to simplify usage even for non-trained operators,
- in the case of transformation in the s←z direction, it is possible to transform both structures IIR and FIR (automatic recognition of types),
- decomposition of output transfer function of analog system into the partial second-order sections in purpose of cascade synthesis.

Program was intentionally drafted in purpose of easy and comprehensive tend. This results in some simplifications and restrictions:

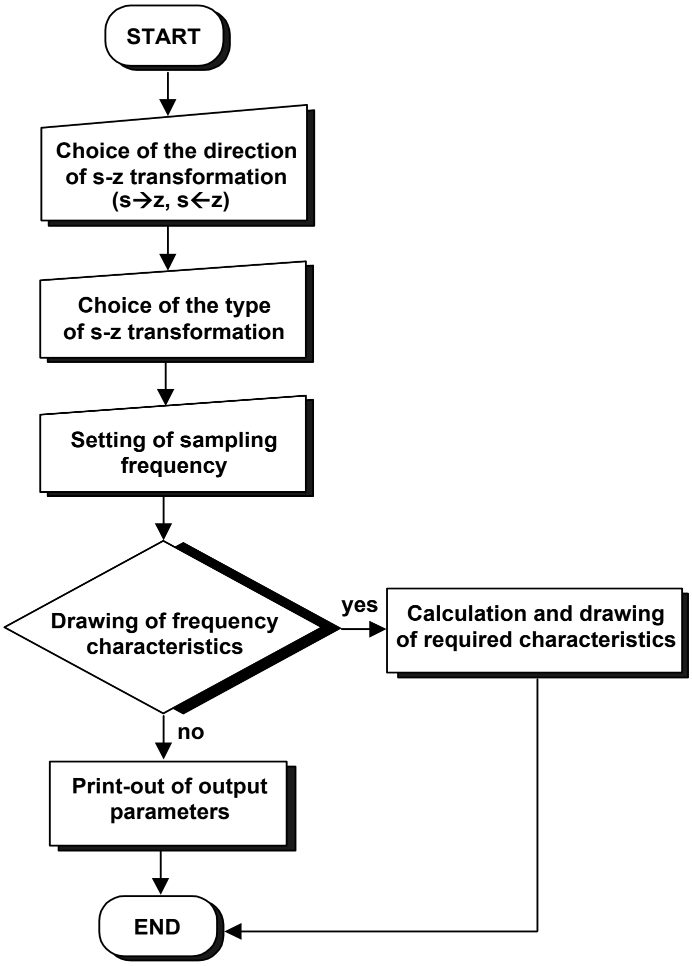

Fig. 3 Simplified developing diagram of *PZ* program

- there was not developed user´s graphic interface, all data are entered from computer keyboard,
- working of the program is dependent on the Matlab system environment,
- meanwhile missed functions for predistortion of frequency axis for individual types of s-z transformations (is possible to solve by the change of sampling frequency).

Considerable emphasis was put on the robustness of used algorithms.

5. Conclusion

PZ program, enabling bilateral transformation of coefficients of continuous-time and linear discrete-time systems, summarizes priorities of up to this time known algorithms and supplements them with new functions. The work of this system was examined during solution of the problem of optimization of continuous-time filters from point of view of the course of characterization of group delay. Continuous-time system was projected on the basis of FIR structure with linear phase and transformed into the *s-plane*. In spite of some imperfections of this solution (potential higher order etc.) designed filter of twelfth-order had favorable course of group delay. System was compared with filters, projected in the classical procedures in the *s-domain* (Feistel-Unbehauen), which comply with given tolerance field [5], [13]. In addition there appeared several interesting features of these selective structures, called in [2] FIR-BL (system is on the boundary of aperiodicity, cascade synthesis goes to low-pass, resp. high-pass filter of second-order with zeros of transfer function and others). System can be used also in transformation of already optimized discrete structures IIR.

Practical verification of *PZ* program is a part of the diploma thesis at the Brno University of Technology, Faculty of Electrical Engineering and Computer Science.

Further possible development associated with presented problems can include:

- combining PZ program with already existing INVAR program [15], which enables the project of discrete structures on the basis of analog prototypes with signal invariant methods and enables to create complete project system,
- completing other types of s-z transformations, first of all LDI (algorithm for the transform is already done), as well as other types (namely some of nonlinear ones),
- thoroughly examining features of LDI transformation (potential possibility of filter optimization).

The aim of the presented development is to offer to possible users alternative means for effective project of selective systems, to state theoretical interesting results and determine limits of possible usage.

Acknowledgement

The Grant Agency of the Czech Republic supports this project under number 102/98/P204.

References

[1] ANTONIOU, A.: Digital Filters: Analysis and Design. McGraw-Hill, New York, 1979.

[2] BIOLEK, D., BIOLKOVÁ, V.: FIR-BL Analog Filters. In: Proc. of MWSCAS '99, Las Cruces, New Mexico (USA), 1999, pp. 259-262.

[3] BIOLKOVÁ, V., BIOLEK, D.: LDI Matrix and its Utilization for the Circuit Analysis and Design. In: Proc. of Radioelektronika '99, FEI VUT Brno (Czech Rep.), 1999, pp. 38-41.

[4] DOSTÁL, T.: Analýza a syntéza obvodů se spínanými kapacitory. VUT Publishing, Brno, 1988.

[5] FEISTEL, K. H., UNBEHAUEN, R.: Tiefpässe mit Tschebyscheff-Charakter der Betriebsdämpfung im Sperrbereich und Maximal Geebneter Laufzeit. Frequenz 19, 1965, pp. 265-282.

[6] KLEIN, W.: Finite Systemtheorie. Teuner Studienbücher, Stuttgart, 1976.

[7] Matlab User's Giude. The Mathworks, Natick, 1996.

[8] POWER, H. M.: The mechanics of the bilinear transformation. IEEE Trans. on CAS, vol. 10, no. 5, 1967, pp. 114-116.

[9] PŠENIČKA, B., PHAM KHAC, Di.: Výpočet přenosové funkce číslicového filtru pomocí Pascalovy matice. Slaboproudý obzor, vol. 46, no. 7., 1985, pp. 348-350.

[10] RABINER, L. R., GOLD, B.: Theory and Application of Digital Signal Processing. Prentice-Hall, Englewood Cliffs, 1975.

[11] RALSTON, A.: A first course in numerical analysis. McGraw Hill, New York, 1965.

[12] VÍCH, R.: Transformace Z a některá její použití. SNTL, Praha, 1983.

[13] ZAPLATÍLEK, K.: Synthesis of transfer functions of analogue filters by means of bilinear transformation In: Proc. of XXVIth General Assembly of URSI. Toronto (Canada), 1999, pp. 230.

[14] BIOLKOVÁ, V., BIOLEK, D.: Generalized Pascal Matrix of First Order s-z Transforms. In: Proc. of the ICECS'99, Pafos, (Cyprus), 1999, pp. 929-931.

[15] BIOLEK, D.: Computer design of IIR filters using signal invariance approach. BIOSIGNAL'94, VUT Brno, 1994.

About authors...

Karel Zaplatílek was born in Svitavy, (Czech Republic) in 1964. He received the Ing. (MSc.) degree from the Brno University of Technology in 1989, and Ph.D. degree from the Brno Military Academy in 1998. His field of professional interest is circuit theory, especially analog and digital filter analysis and design.

Michal Lares was born in Novy Jicin (Czech Republic), in 1975. He received the Ing. (M.S.) degree in Electrical Engineering at the Brno University of Technology in 1998. Since 1998, he has been Ph.D. student at the Dept. of Telecommunications of Brno University of Technology. He focuses on frequency filters with optimized group delay.