Circuit theory for computational algebra

Patrick Dewilde*
dedicated to the memory of Vitold Belevitch

Abstract **- Algebra, be it linear or discrete, has had great importance for circuit theory. It has been used to solve problems in circuit analysis and circuit synthesis. But is the opposite true also? Has circuit theory, and its twin brother, system theory, been able to produce new results in algebra, and in particular in computational algebra? In this lecture we shall show that that is indeed the case: We shall concentrate on core algebraic problems of approximation, algebraic minimality and inversion. We shall also show that the scope can be larger than what is customary in linear algebra, also infinite systems can be handled by circuit theoretic methods. The approach opens a variety of new perspectives in linear algebra which we shall briefly describe at the end of the lecture.**

1 Introduction: some history

The introduction of complex function theory in Electrical Engineering in the early years of the 20th century has been a watershed that allowed the solution of major problems in electrical circuit analysis and synthesis. It allowed the treatment of many problems considered 'of analytic nature' by algebraic methods. The connection could be laid between desired signal behavior at the input and output ports of circuits, the properties of the constituting elements and the circuit structure. Propagation of a signal through a linear medium is described by a convolution, which after Laplace transformation converts into an algebraic product:

$$\boldsymbol{y(s) = T(s)u(s)}$$

here *u(s)* is the Laplace transform of the signal, *T(s)* characterizes the transfer behavior of the system and *y(s)* is the Laplace transform of the resulting output. Much of circuit analysis and synthesis has to do with the properties of the transfer operator *T(s)*. We give two common examples:

- *Filtering:* it is desired that *T(s)* exhibits as closely as desired a given frequency behavior, e.g. *T(jw)* must be a low pass filter;
- *Model reduction:* an operator $T_a(s)$ is desired which approximates *T(s)* within some tolerance given, with minimal computational complexity.

These examples are put here in a linear context, but the problems are of course more general: the questions make sense in a non-linear context as well. Classical circuit theory has been able to find solutions to these, and to many more problems. We give some quick comments on the two problems mentioned:

- *filtering:* the classical approach has been to approximate an ideal behavior with a rational transfer function *T(s)=p(s)/q(s)* keeping desired physical circuit behavior, e.g. contractivity or positive definiteness;
- *model reduction:* the classical approach has been to define a Hankel operator connected to the large system behavior and then to approximate that Hankel operator with one of low complexity, leading to so called 'balanced model reduction' or 'Hankel norm model reduction'.

Much of classical circuit theory is highly dependent on the properties of analytic functions in a region of the complex plane. So what could be the import of these theories for computational algebra?

The most basic problems in linear algebra are connected to the prototype linear equation:

$$\boldsymbol{y = Tu}$$

where *u* and *y* are vectors and *A* is a matrix of appropriate dimensions. The questions we raised make sense also in this case: can *A* exhibit a desired behavior with low computational complexity? or, can *A* be approximated by a another matrix A_a of low computational complexity?

The algebraic link was first provided by a number of great circuit and system theorists in the 1960's, with Vitold Belevitch playing a central role [1]. I do not want to leave unmentioned the contributions of a number of other researchers, in particular: D.Youla, R.Kalman, R.W.Newcomb, Ch. Desoer, L. Zadeh and B.D.O.Anderson, where the last one probably made the most comprehensive linkage between circuit theory and linear algebra in his book [2].

The contributions just mentioned in fact amounted to the introduction of the notion of state and state transitions, leading to a representation of the transfer behavior of a system using state equations:

$$\begin{cases} \partial \boldsymbol{x(t)} = \boldsymbol{Ax(t)} + \boldsymbol{Bu(t)} \\ \boldsymbol{y(t)} = \boldsymbol{Cx(t)} + \boldsymbol{Du(t)} \end{cases}$$

where the ∂ indicates a time evolution (a 'next' operator: a time derivative in the time-continuous case or the next value in the time discrete case), and *{A,B,C,D}* is a quadruple of matrices. Although linear algebra was ushered in circuit theory in a new way, a

* DIMES, TU Delft, POB 5031, 2600GA Delft, the Netherlands, tel. +31 15 278 5089, email dewilde@dimes.tudelft.nl

number of unexpected problems were introduced as well. While it was relatively easy to characterize physical circuit behavior using Laplace transforms, that certainly was not the case with the state description. For example, contractiveness and positive reality were hard to characterize, leading eventually to the PR-lemma, known otherwise as the Kalman-Yacubovitch-Popov criterium. Also complexity gave its host of problems. The state space representation is not unique, it utilizes an intermediate state quantity *x* which is certainly not unique. Depending on its choice, a realization may have very different computational properties. One may safely say that the question of the impact of circuit and system theory on computational algebra was still open. What was needed was a way of handling the opposite direction: that will be the main goal of this exposition.

2 From circuit to matrix

Since we are interested in translating circuit properties to properties of matrices, we put ourselves squarely in the context of discrete time systems. Since the great contributions of A. Fettweiss, we know that most if not all continuous time properties can be translated to discrete time properties via the bilinear transformation $\boldsymbol{z} = \frac{1-\boldsymbol{s}}{1+\boldsymbol{s}}$ (which transforms the right half plane to the unit disc - we use here the convention that a delay is represented by *z* rather than the common engineering convention z^{-1}).

A transfer operator of the type $\boldsymbol{T}(\boldsymbol{z}) = \sum \boldsymbol{T}_{\boldsymbol{k}} \boldsymbol{z}^{\boldsymbol{k}}$ will now correspond to an input/output map of the type:

$$\begin{bmatrix} \vdots \\ \boldsymbol{y}_{-1} \\ \boldsymbol{y}_0 \\ \boldsymbol{y}_1 \\ \vdots \end{bmatrix} = \begin{bmatrix} \ddots & \ddots & \ddots & \ddots & \ddots \\ \ddots & \boldsymbol{T}_0 & \boldsymbol{T}_{-1} & \boldsymbol{T}_{-2} & \ddots \\ \ddots & \boldsymbol{T}_1 & \boldsymbol{T}_0 & \boldsymbol{T}_{-1} & \ddots \\ \ddots & \boldsymbol{T}_2 & \boldsymbol{T}_1 & \boldsymbol{T}_0 & \ddots \\ \ddots & \ddots & \ddots & \ddots & \ddots \end{bmatrix} \begin{bmatrix} \vdots \\ \boldsymbol{u}_{-1} \\ \boldsymbol{u}_0 \\ \boldsymbol{u}_1 \\ \vdots \end{bmatrix}$$

in other words: a doubly infinite Toeplitz matrix. Classical operators do translate to a very special type of non-standard (because of non finite dimensions) matrix. How could the normal matrix case be fitted in? A first step would be to take the individual components of the transfer *time varying*: at position *(i,j)* we would put a $T_{i,j}$ rather than a T_{i-j}. A second important step is to make also the dimensions time varying, including 'no dimensions at all', when an input or output is not present. Hence, the input dimensions are indexed by numbers $\{m_j\}$, the output dimensions by numbers (n_i) and the block $T_{i,j}$ has dimensions $n_i \times m_j$. When $m_j=0$, then $T_{i,j}$ has zero columns (but does have rows!). We extend the classical matrix calculus with matrices of dimension zero and utilize the convention that the product of a matrix of dimension $m \times 0$ with one of dimension $0 \times n$ is an $m \times n$ matrix with entries zero. In this way, a finite matrix can be interpreted as an infinite one, whereby dimensions of entries become zero for small and large values of the indices. One is free to position the 0,0 or central element where it is convenient. We indicate the sequence of input dimensions by a sequence $\{\cdots, \boldsymbol{m}_{-1}, \langle \boldsymbol{m}_0 \rangle, \boldsymbol{m}_1, \cdots\}$ and likewise (with *n*'s) for the output sequence, where the central, zero'th element is distinguished with triangular brackets for identification. For example: a map $\{\langle 2 \rangle, 1, 1\}$ to $\{1, \langle 2 \rangle\}$ will have the form

$$\begin{bmatrix} * & * & * & * \\ \langle * & * \rangle & * & * \\ * & * & * & * \end{bmatrix}$$

where the central element is put between triangular brackets. Elements in positions *k,k* for arbitrary *k* belong to the 'main diagonal'. In the example shown, the main diagonal consists of a single non-trivial element, a 2×2 block matrix in position 0,0 at the left lower corner. Lower diagonal matrices or operators play a special role: they correspond to classical 'causal' maps, i.e. maps for which inputs only influence future outputs. A state space representation of such causal maps now becomes time varying, and it has the form (we specialize to the discrete time case)

$$\begin{cases} \boldsymbol{x}_{\boldsymbol{k}+1} = \boldsymbol{A}_{\boldsymbol{k}} \boldsymbol{x}_{\boldsymbol{k}} + \boldsymbol{B}_{\boldsymbol{k}} \boldsymbol{u}_{\boldsymbol{k}} \\ \boldsymbol{y}_{\boldsymbol{k}} = \boldsymbol{C}_{\boldsymbol{k}} \boldsymbol{x}_{\boldsymbol{k}} + \boldsymbol{D}_{\boldsymbol{k}} \boldsymbol{u}_{\boldsymbol{k}} \end{cases}$$

in which all matrices are block matrices and dimensions are also time-varying, in particular, they may disappear. State space representations are not unique, in fact and with $\boldsymbol{R}_{\boldsymbol{k}}$ an invertible matrix for each *k*, we have another adequate representation as

$$\begin{cases} \boldsymbol{x}_{\boldsymbol{k}+1} = \boldsymbol{R}_{\boldsymbol{k}+1}^{-1} \boldsymbol{A}_{\boldsymbol{k}} \boldsymbol{R}_{\boldsymbol{k}} \boldsymbol{x}_{\boldsymbol{k}} + \boldsymbol{R}_{\boldsymbol{k}+1}^{-1} \boldsymbol{B}_{\boldsymbol{k}} \boldsymbol{u}_{\boldsymbol{k}} \\ \boldsymbol{y}_{\boldsymbol{k}} = \boldsymbol{C}_{\boldsymbol{k}} \boldsymbol{R}_{\boldsymbol{k}} \boldsymbol{x}_{\boldsymbol{k}} + \boldsymbol{D}_{\boldsymbol{k}} \boldsymbol{u}_{\boldsymbol{k}} \end{cases}$$

We shall denote the state space dimensions by $\boldsymbol{d}_{\boldsymbol{k}}$. Introducing a generic causal shift

$$\boldsymbol{Z} = \begin{bmatrix} \ddots & & & & \\ \ddots & 0 & & & \\ & \boldsymbol{I} & \langle 0 \rangle & & \\ & & \boldsymbol{I} & 0 & \\ & & & \ddots & \ddots \end{bmatrix}$$

mapping state sequences $\{\cdots, \boldsymbol{d}_{-1}, \langle \boldsymbol{d}_0 \rangle, \boldsymbol{d}_1, \cdots\}$ to $\{\cdots, \boldsymbol{d}_{-2}, \langle \boldsymbol{d}_{-1} \rangle, \boldsymbol{d}_0, \cdots\}$, and block diagonal global representations for the state space matrices as e.g.

$$\boldsymbol{A} = \begin{bmatrix} \ddots & & & & \\ & \boldsymbol{A}_{-1} & & & \\ & & \langle \boldsymbol{A}_0 \rangle & & \\ & & & \boldsymbol{A}_1 & \\ & & & & \ddots \end{bmatrix}$$

we find the global state space representation for the causal operator *T* (assuming I-ZA invertible)

$$\boldsymbol{T} = \boldsymbol{D} + \boldsymbol{C}(\boldsymbol{I} - \boldsymbol{Z}\boldsymbol{A})^{-1}\boldsymbol{Z}\boldsymbol{B}\,.$$

One should realize that every computing system can be represented by state space equations, the state stands for what the computing system remembers from the past of the process. When the computation is linear, then the state space equations will be linear as well. The dimension of the state space controls the size of the memory needed. The number of algebraically independent parameters in the state space matrices will control the complexity of the calculation at each point in time *k*. In our theory here we do not allow reshuffling of the inputs and outputs across time points, we assume that when an input u_k comes in, the corresponding y_k has to be computed.

3. Realization theory

Realization theory is concerned with finding minimal $\boldsymbol{d_k}$'s given the transfer operator or matrix *T*. The answer is given by what is known as 'generalized realization theory', which goes back to the work of Kronecker at the end of the 18th century. We define the 'kth Hankel operator' related to *T* as

$$\boldsymbol{H_k} = \begin{bmatrix} \boldsymbol{T_{k,k-1}} & \boldsymbol{T_{k,k-2}} & \cdots \\ \boldsymbol{T_{k+1,k-1}} & \boldsymbol{T_{k+1,k-2}} & \cdots \\ \vdots & \vdots & \ddots \end{bmatrix}.$$

The result is now that the rank of H_k equals the minimal $\boldsymbol{d_k}$. Any minimal realization can be obtained by computing minimal factorizations for all H_k:

$$\boldsymbol{H_k} = \begin{bmatrix} \boldsymbol{C_k} \\ \boldsymbol{C_{k+1}A_k} \\ \boldsymbol{C_{k+2}A_{k+1}A_k} \\ \vdots \end{bmatrix} [\boldsymbol{B_{k-1}}\ \boldsymbol{A_{k-1}B_{k-2}}\ \boldsymbol{A_{k-1}A_{k-2}B_{k-3}}\ \cdots]$$

The factorization determines the realization.

4. Algebraic minimality

The generalized Kronecker minimality result is as straight as it possibly can be, quite another matter is to obtain algebraic minimality. We wish to find local realization matrices $\{A_k,B_k,C_k,D_k\}$ such that the total number of local multiplications and additions is minimal. This problem can be approached through embedding or Darlington theory. Let us restrict the discussion to the case where the operator *T* is causal and contractive, i.e. $|\boldsymbol{T}| \le 1$. Darlington theory will provide for a realization of an embedding

$$\Sigma = \begin{bmatrix} \boldsymbol{T} & \Sigma_{12} \\ \Sigma_{21} & \Sigma_{22} \end{bmatrix}$$

which is causal and unitary (in mathematical terms: inner) and has a unitary state space realization of the same dimension as the realization for *T*:

$$\begin{bmatrix} \boldsymbol{A} & \boldsymbol{B} & \boldsymbol{B}_2 \\ \boldsymbol{C} & \boldsymbol{D} & 0 \\ \boldsymbol{C}_2 & \boldsymbol{D}_{21} & \boldsymbol{D}_{22} \end{bmatrix}_{\boldsymbol{k}}.$$

The computations can be reduced to the utilization of orthogonal rotations (Jacobi transformations), lossless transformations in circuit theory terms. We would need $q_k*(q_k-1)/2$ Jacobi rotations, when q_k is the dimension of the total realization matrix. This is also the number of free algebraic parameters, in case of real computations. Hence: both state space complexity and algebraic complexity can be minimized simultaneously, just as is the case in classical lossless circuit realization theory.

5. Factorization theory

An important intermediate theory needed in most synthesis questions is factorization theory. We briefly introduce a few main results. Again we consider a causal (i.e. lower triangular) matrix or operator *T* with minimal realization *{A,B,C,D}*. An external factorization is a factorization of the type:

$$\boldsymbol{T} = \Delta_{\boldsymbol{r}}^{*}\boldsymbol{U_r}$$

where Δ_r is causal an U_r is inner, and the factorization is minimal, the index 'r' standing for 'right' since there will be a dual left factorization. External factorizations amount to recursively solving a linear equation known as a 'Lyapunov-Stein' equation. This can be done in a straightforward and numerically stable way on the state space representation for *T*, resulting in state space representations for D_r and U_r, whereby the state representation for *U* shares *A* and *B* matrices with *T*. Another important type of factorization is known as inner-outer factorizations, it has the form:

$$\boldsymbol{T} = \boldsymbol{V_r T_{or}}\,,$$

where now both V_r and T_{or} are causal, and V_r is isometric, i.e. $\boldsymbol{V_r^{*}V_r} = 1$ (it is not necessarily square). The inner-outer factorization is very different from the external factorization, in that it exhibits the 'causal zero structure' of *T*, while the external factorization exhibits its global pole structure. The inner-outer factorization, when expressed in state space terms,

gives rise to a so called 'square root algorithm', first introduced by Morf and Kailath [3] in the context of Kalman filtering. It involves an intermediate sequence $\{Y_k\}$ whose dimensions reflects the causal zero structure of T, and a QL factorization:

$$\begin{bmatrix} \boldsymbol{Y_{k+1}A_k} & \boldsymbol{Y_{k+1}B_k} \\ \boldsymbol{C_k} & \boldsymbol{D_k} \end{bmatrix} = \boldsymbol{W_k} \begin{bmatrix} 0 & 0 \\ \boldsymbol{Y_k} & 0 \\ \boldsymbol{C_{ok}} & \boldsymbol{D_{ok}} \end{bmatrix}$$

in which the known left hand matrix is brought into lower echelon form using a unitary transformation matrix W_k. The realization for V_r is derived from the W_k, while the realization for T_{or} is given by $\{A_k,B_k,C_{ok}\ D_{ok}\}$. The square root algorithm in its various forms may be one of the most powerful algorithms of modern algebra! Its ramifications, especially the so called doubling algorithms utilize notions of circuit theory in unexpected ways.

6. Approximation theory

Both in filter theory and in model reduction theory, approximation theory plays a central role. The central result in complex function approximation theory is called 'Hankel norm approximation' and it was the central topic of the famous 'AAK theory' - a generalization of the classical Schur-Takagi interpolation theory [4]. Given a causal transfer operator T representing a desired behavior, it produces a causal approximating operator T_a, of minimal complexity, which meets the norm constraint

$$\left|\boldsymbol{T} - \boldsymbol{T_a}\right|_{\boldsymbol{H}} \leq \boldsymbol{e}$$

for a given error ε. The norm used is the Hankel norm, it is the sup of the Hankel operators defined earlier (in the classical time invariant case there is just one Hankel matrix). It turns out [5] that the Hankel norm model reduction problem generalizes to the matrix case and involves just the computation of a single external factorization, be it an external factorization of a special type - using a J-unitary matrix with J a signature matrix, rather than a unitary factorization. This appears to be a new results in matrix theory, for a comprehensive analysis see [6].

7. Further results

In the same vein as detailed up to now, many more classical circuit theoretical results translate to results in matrix calculus:

- *Darlington synthesis:* embedding of a contractive matrix or operator in a unitary one and resulting in algebraically minimal operations.
- *System inversion:* in particular channel equalization. In the general case it amounts to an external factorization followed by an inner-outer factorization.
- *Minimal sensitivity control:* this difficult problem (initiated by D.Youla and G.Zames) amounts to a J-inner/outer factorization and results also in coupled square root algorithms, as was shown by Helton and Ball [7].

8. Future perspective

Although quite a few important results have been obtained translating circuit theory to algebra, there remains a lot to do! Foremost there is the extension of the theory to the non-linear case. The differential of a non-linear system is a time varying system, the time varying theory should provide for a new way to approach non-linear problems such as model reduction, system inversion and robust control. In fact, there is already quite a literature showing that that should be the case. On the other hand, even in the linear case there is more work to do. Let me just mention two unsolved problems of potentially great technical importance:

- inversion of outer systems (with applications to the modeling of transmission lines in large integrated circuits);
- sensitivity conditioning of systems.

References

[1] V. Belevitch, *Classical Circuit Theory*, Prentice Hall 1968.

[2] B.D.O. Anderson and S. Vongpanitlerd, *Network Analysis and Synthesis*, Prentice Hall, 1973.

[3] M. Morf and T. Kailath, Square root algorithms for least squares estimation, *IEEE Trans. on Automatic Control*, 20(4): 487-497, 1975.

[4] V.M. Adamyan, D.Z.Arov and M.G.Krein, "Analytic properties of Schmidt pairs for a Hankel operator and the generalized Schur-Takagi problem", *Math. USSR Sbornik* **15**(1) 31-73 (transl. of *Iz. Akad. Nauk. Armjan. SSR Ser. Mat.* **6** (1971).

[5] P. Dewilde and A.-J. van der Veen, "On the Hankel-norm approximation of upper-triangular operators and matrices", *Integral Eq. Operator Th.*, **17**(1):1-45, 1993.

[6] P. Dewilde and A.-J. van der Veen, *Time-Varying Systems and Computations*, Kluwer 1998.

[7] J.A.Ball and J.W.Helton, "A Beurling-Lax theorem for the Lie group *U(m,n)* which contains most classical interpolation theory", *J. Operator Theory*, **9**:107-142, 1983.