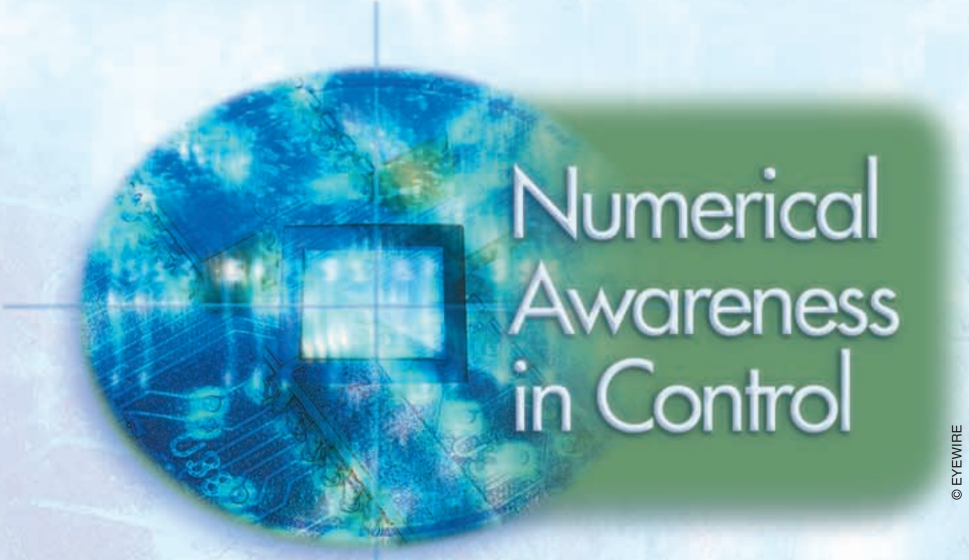
FEATURE

The Basics of Developing Numerical Algorithms



© EYEWIRE

General principles that lead to numerically reliable algorithms for solving a large collection of control problems.

By Paul Van Dooren

The literature of systems and control, especially that of multi-input/multi-output linear systems, has always been rich in numerical linear algebra problems. As a result of fruitful interaction between these areas of expertise, numerous numerical algorithms have been developed for solving specific problems related to systems and control theory. The linear algebra community has been pleased that their tools are of substantial use, and the control community has been pleased that numerical analysts find control problems of sufficient challenge to show interest in them. The interaction between the two groups led to the development of several numerical libraries for reliably solving numerical problems posed by the control community. One of the most advanced numerical software projects addressing such issues is the NICONET project (see http://www.win.tue.nl/niconet/niconet.html), which is partly responsible for the SLICOT library (see [1]).

The interactions are mainly in the area of linear time-invariant systems, which can be modeled as a set of differential equations

0272-1708/04/$20.00©2004IEEE

$$P\left(\frac{d}{dt}\right)y(t) = Q\left(\frac{d}{dt}\right)u(t), \qquad (1)$$

or difference equations

$$P(\mathcal{D})y(k) = Q(\mathcal{D})u(k). \qquad (2)$$

Here $P(\cdot)$ and $Q(\cdot)$ are polynomial matrices of appropriate dimensions, $u(\cdot) \in \mathcal{R}^m$ is the vector of inputs, $y(\cdot) \in \mathcal{R}^p$ is the vector of outputs, and both are functions of time. When the time variable is the continuous variable t, the operator is the differential operator d/dt; when the time variable is the discrete variable k, the operator is the advance operator $\mathcal{D}$.

In systems and control theory, one often uses state-space models as an equivalent representation of these dynamical systems. In such models the relation between inputs and outputs is described by means of a state $x(\cdot) \in \mathcal{R}^n$ and a system of first-order differential or difference equations

$$\begin{aligned} \lambda E x(\cdot) &= Ax(\cdot) + Bu(\cdot), \\ y(\cdot) &= Cx(\cdot) + Du(\cdot), \end{aligned} \qquad (3)$$

where λ represents either the differential operator d/dt or the difference operator $\mathcal{D}$. Here E and A are real or complex $n \times n$ matrices and B, C, and D are real or complex $n \times m$, $p \times n$, and $p \times m$ matrices, respectively. By applying the Laplace transform or the z-transform to (3) and eliminating the state $x(\cdot)$, one obtains the transfer function $T(\lambda)$, which, for zero initial state vector $x(0)$, describes the relation between inputs and outputs

$$T(\lambda) = C(\lambda E - A)^{-1}B + D, \quad y(\cdot) = T(\lambda)u(\cdot).$$

Comparing this expression with the system of differential/difference equations described in (1) and (2) yields the relation

$$T(\lambda) = P^{-1}(\lambda)Q(\lambda).$$

Polynomial matrices, rational matrices, and state-space models are key representations of linear time-invariant systems, and they obviously lead to matrix problems with more structure than, for example, the classical standard eigenvalue problems encountered in numerical linear algebra. In the following sections, we review some of the most important developments in this area and discuss their numerical aspects.

Conditioning and Stability

To explain the concepts of numerical stability of an algorithm and conditioning of a problem, we consider the problem of computing the eigenvalues of an $n \times n$ complex matrix A. It is well known that there always exists an invertible transformation T such that

$$A = TA_JT^{-1},$$

where the bidiagonal Jordan form A_J has the eigenvalues of A on its diagonal and possibly ones on the first superdiagonal corresponding to each nontrivial Jordan block of A [2]. It is also well known that any algorithm attempting to compute this decomposition yields computed quantities $\overline{A}_J$ and $\overline{T}$ (that is, the results that are stored in the computer) for which the backward error ΔA_J defined by

$$A + \Delta A_J = \overline{T}\,\overline{A}_J\overline{T}^{-1} \qquad (4)$$

cannot be bounded for all matrices A [3], [4]. This property is mainly due to the fact that the transformation matrix T has a condition number

$$\kappa(T) := \left\|T\right\|_2\left\|T^{-1}\right\|_2$$

that can be arbitrarily large for some matrices A.

But there also exists a unitary matrix U such that

$$A = UA_SU^*,$$

where the triangular Schur form A_S has the eigenvalues of A on its diagonal as well. The condition number $\kappa(U) := \|U\|_2\|U^*\|_2$ now equals one [3]. This fact is crucial for showing that the backward error ΔA_S defined by

$$A + \Delta A_S = \overline{U}\,\overline{A}_S\overline{U}^* \qquad (5)$$

can be bounded by $\|\Delta A_S\|_2 < \epsilon c\|A\|_2$ for all matrices A, provided it is computed with an appropriate algorithm [3]. Here c is a constant reasonably close to one, and ϵ is the precision of the machine used for the computations (see [5] for more details). In this case, the computed decomposition corresponds exactly to a slightly perturbed matrix $A + \Delta A_S$. An algorithm for which such a property holds is said to be backward stable. If one is interested in computing the eigenvalues of A, one is better off using the Schur form (5) rather than the Jordan form (4), since there is a backward stable algorithm for computing (5) from which the eigenvalues of (the slightly perturbed) A can be directly obtained. By doing so, we computed the exact eigenvalues of a nearby matrix $A + \Delta A_S$!

Backward stability does not imply that we obtained the desired result (the eigenvalues of A in this case) to high accuracy. Rather, the accuracy depends on how perturbations on the data (namely ΔA) affect the result X we are interested in (the eigenvalues of A in our example). The size of the perturbation is commonly measured by the (absolute) condition $\kappa[f(A)]$ of the computed object $X = f(A)$ given by

$$\kappa[f(A)] = \lim_{\delta\to 0} \sup_{\|\Delta A\|_2 \le \delta} \left[\left\| X - \overline{X} \right\|_2 / \delta \right], \quad \overline{X} = f(A + \Delta A).$$

One can view the quantity $\kappa[f(A)]$ as the worst case derivative of the computed function $f(A)$ in any direction ΔA, or in other words, the sensitivity of this function versus small perturbations. When the above limit exists, one derives an approximate bound for the error ΔX on the computed result of the form

$$\|\Delta X\|_2 \le \kappa[f(A)].\|\Delta A\|_2 + O\left(\|\Delta A\|_2^2\right). \tag{6}$$

When $\kappa[f(A)]$ is not bounded, the above inequality does not make sense and the problem of determining X from A is ill posed (as opposed to well posed when $\kappa[f(A)]$ is bounded). When $\kappa[f(A)]$ is finite and relatively large or relatively small, the problem is badly conditioned and well conditioned, respectively. For the eigenvalue problem it is known that the eigenvalues of a symmetric matrix A are well conditioned, whereas the eigenvalue problem of a matrix with a nontrivial Jordan structure (for example, without a full set of eigenvectors) is ill posed [3].

Notice that backward stability is a property of an algorithm, while conditioning is associated with a problem and the specific data for that problem. As indicated in (6), the errors ΔX in the result depend on both the stability of the algorithm used and the conditioning of the problem being solved. A backward stable algorithm has a uniform bound for $\|\Delta A\|_2$, which is good since the error ΔX depends essentially on the condition of the problem only. An unstable algorithm, on the other hand, may yield a large error ΔX even when the problem is well conditioned. For more details on this section, see [4]–[6].

State-Space Models and Coordinate Transforms

State-space models are far from unique for a given transfer function and its associated input/output behavior. A class of transformations that does not affect the transfer function are the system equivalence transformations

$$\{E, A, B, C, D\} \Longrightarrow \left\{\hat{E}, \hat{A}, \hat{B}, \hat{C}, \hat{D}\right\} := \{SET, SAT, SB, CT, D\}, \tag{7}$$

where S and T are constrained to be nonsingular. For standard state-space models (where $E = I$), equivalence transformations (7) become similarity transformations since $\hat{E} = ST$ must be the identity

$$\{A, B, C, D\} \Longrightarrow \left\{\hat{A}, \hat{B}, \hat{C}, \hat{D}\right\} := \left\{T^{-1}AT, T^{-1}B, CT, D\right\}. \tag{8}$$

Although these state-space models are not the only ones used for systems and control purposes, they are the models that have been most heavily studied as far as numerical algorithms are concerned (see [7]). If the system is given in one of these forms, one typically has to analyze its properties (such as the frequency response, poles/zeros, stability, and robustness) and then design a particular controller to improve some characteristics or to satisfy certain design criteria (tracking, robustness, and optimality criteria). Many analysis and design problems are now well understood and their theoretical solution is often described in terms of canonical forms (such as the Jordan or the Kronecker canonical form [7]), which have been defined for state-space models of multivariable linear systems. These forms are typically sparse since they are described with a minimum number of parameters. Therefore, they often allow one to efficiently characterize all solutions to a particular problem, which is appealing. Unfortunately, these forms are also sensitive to computation, and they can invoke a coordinate transformation that is poorly conditioned [7].

For most analysis and design problems encountered in linear system theory, one can make use of special forms that can be obtained under transformations that are well conditioned, such as unitary or orthogonal transformations [8]. Such transformations have become a major tool in the development of reliable numerical linear algebra algorithms. One motivation for well-conditioned transformations is the numerical sensitivity of the problem at hand. The sensitivity (or conditioning) of problems in linear algebra can often be expressed in terms of norms, singular values, or angles between spaces, and each of these quantities is invariant under orthogonal transformations. These transformations therefore allow one to reformulate the problem in a new coordinate system that is more appropriate for solving the problem, without affecting its sensitivity. A second reason is the numerical stability of the algorithm used for solving the problem. Most decompositions involving orthogonal transformations can be obtained by a sequence of Givens or Householder transformations that can be performed in a numerically stable manner. The concatenation of such transformations can also be applied in a backward stable manner because numerical errors resulting from previous steps are maintained in norm throughout subsequent steps. In fact, these transformations (and their inverse) have two-norm equal to one.

We explain this stability property below by analyzing the poles of a single-input/single-output system given in standard state-space form, where $E = I, S = T^{-1}$ and $m = p = 1$. For such models the poles are the eigenvalues of the matrix A (provided the realization is minimal), and the classical form describing the fine structure of these eigenvalues is the Jordan canonical form. We choose the similarity transformation (8) where $\hat{A}_J = T^{-1}AT$ is in Jordan canonical form. For convenience, we give the transformed system $\{\hat{A}_J, \hat{B}_J, \hat{C}_J, \hat{D}_J\}$ in the form of the compound matrix

$$\left[\begin{array}{c|c} \hat{A}_J & \hat{B}_J \\ \hline \hat{C}_J & \hat{D}_J \end{array}\right] := \left[\begin{array}{ccccccc|c} \lambda_1 & 1 & 0 & 0 & 0 & 0 & 0 & \times \\ 0 & \lambda_1 & 1 & 0 & 0 & 0 & 0 & \times \\ 0 & 0 & \lambda_1 & 0 & 0 & 0 & 0 & \times \\ 0 & 0 & 0 & \lambda_2 & 1 & 0 & 0 & \times \\ 0 & 0 & 0 & 0 & \lambda_2 & 0 & 0 & \times \\ 0 & 0 & 0 & 0 & 0 & \lambda_3 & 0 & \times \\ 0 & 0 & 0 & 0 & 0 & 0 & \lambda_4 & \times \\ \hline \times & \times & \times & \times & \times & \times & \times & \times \end{array}\right].$$

Since we chose a single-input/single-output example, there is only one Jordan block associated with each individual eigenvalue; for multiple-input/multiple-output systems this does not have to be the case. This form not only describes the poles of the system but contains more information, such as the partial fraction description of the transfer function. A disadvantage of the form is that it requires a state-space transformation T to transform A into its Jordan form $\hat{A}_J$, and that the norms of T and T^{-1} cannot be bounded in general. On the other hand, when one restricts T to be orthogonal, then so is T^{-1}, and both are bounded in norm. Under such transformations, one can always reduce A to triangular form, called the Schur form, which also has the eigenvalues on its diagonal, given by

$$\left[\begin{array}{c|c} \hat{A}_S & \hat{B}_S \\ \hline \hat{C}_S & \hat{D}_S \end{array}\right] := \left[\begin{array}{ccccccc|c} \lambda_1 & \times & \times & \times & \times & \times & \times & \times \\ 0 & \lambda_1 & \times & \times & \times & \times & \times & \times \\ 0 & 0 & \lambda_1 & \times & \times & \times & \times & \times \\ 0 & 0 & 0 & \lambda_2 & \times & \times & \times & \times \\ 0 & 0 & 0 & 0 & \lambda_2 & \times & \times & \times \\ 0 & 0 & 0 & 0 & 0 & \lambda_3 & \times & \times \\ 0 & 0 & 0 & 0 & 0 & 0 & \lambda_4 & \times \\ \hline \times & \times & \times & \times & \times & \times & \times & \times \end{array}\right]. \qquad (9)$$

If one is interested in computing only the poles of the system, it is well known that the latter form is numerically more reliable and actually requires less computation than the Jordan form [3].

There are other classes of orthogonal coordinate transformations such as the Hessenberg and staircase forms [8]. These forms have been proposed for solving a variety of problems such as computing zeros, finding minimal realizations, solving pole-placement problems, constructing observers, and computing frequency responses (see [7] and references therein). In each of these problems, the use of orthogonal or unitary transformations is crucial for guaranteeing the reliability of the computed result.

Trading Speed for Accuracy

For reasons of efficiency, one may want to use nonorthogonal but well-conditioned transformations that reduce the dynamics matrix A to a special form, such as block-diagonal [9]. One example is the tridiagonal realization of an nth order scalar transfer function $G(\lambda) = C(\lambda I - A)^{-1}B + D$ given by

$$\left[\begin{array}{c|c} \hat{A}_T & \hat{B}_T \\ \hline \hat{C}_T & \hat{D}_T \end{array}\right] := \left[\begin{array}{cccc|c} \alpha_1 & \gamma_2 & & & \beta_1 \\ \beta_2 & \alpha_2 & \ddots & & 0 \\ & \ddots & \ddots & \gamma_n & \vdots \\ & & \beta_n & \alpha_n & 0 \\ \hline \gamma_1 & 0 & \ldots & 0 & \alpha_0 \end{array}\right]. \qquad (10)$$

Almost all scalar transfer functions have such a realization; a necessary and sufficient condition is that the $n \times n$ Hankel matrix H built on the moments $H_{i,j} := C A^{i+j-1} B$ of every system realization $\{A, B, C, D\}$ of $G(\lambda)$ must have an LU factorization without pivoting [10]. Let us compare realizations for the system

$$\left[\begin{array}{c|c} A & B \\ \hline C & D \end{array}\right] := \left[\begin{array}{cccc|c} -3.01 & -3.03 & -1.03 & -(0.01+10^{-11}) & 1 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ \hline 1 & 4.5\cdot 10^{-2} & 6.75\cdot 10^{-4} & 3.3375\cdot 10^{-6} & 0 \end{array}\right]. \qquad (11)$$

Since the four (rounded) eigenvalues λ_i of A are $\{-0.0100, -0.9998, -1.0001 \pm 0.0002_{\mathrm{J}}\}$, there is a cluster of three eigenvalues around -1. Although these three eigenvalues are close to each other, they are by no means identical, relative to the machine precision ϵ, which, for this example, is approximately 10^{-16}. A Jordan form realization of this system is given by

$$\left[\begin{array}{c|c} \hat{A}_J & \hat{B}_J \\ \hline \hat{C}_J & \hat{D}_J \end{array}\right] := \left[\begin{array}{cccc|c} \lambda_1 & 0 & 0 & 0 & r_1 \\ 0 & \lambda_2 & 0 & 0 & r_2 \\ 0 & 0 & \lambda_3 & 0 & r_3 \\ 0 & 0 & 0 & \lambda_4 & r_4 \\ \hline 1 & 1 & 1 & 1 & 0 \end{array}\right], \qquad (12)$$

where the coefficients r_i turn out to be large (r_2, r_3, and r_4 are of the order 10^6 and r_3 and r_4 are complex). The cause of these large entries is the proximity of the three eigenvalues. Even with methods that try to recognize clusters of eigenvalues and hence identify the Jordan canonical form in a more robust manner [11], [12], there is still the problem that the transformation to Jordan canonical form is not a continuous function of the parameters of the system, and therefore gives rise to numerical difficulties. On the other hand, the tridiagonal realization has entries of reasonable size because the transformation yielding (10) is well conditioned. In other words, the transformation for placing the system (11) in Jordan form is badly conditioned, while the transformation for placing it in tridiagonal form has an acceptable condition number. When computing the step responses of systems (10) and (12), the response calculated from the Jordan form representation in this example is at least six digits less accurate than the response calculated using the tridiagonal representation.

Trading Simplicity for Accuracy

A typical control action used to influence the behavior of the system is linear state feedback, which consists of feeding back a linear function $Fx(\cdot)$ of the state to the input $u(\cdot)$. Replacing the input $u(\cdot)$ by $Fx(\cdot)+u(\cdot)$ yields

$$\begin{aligned}\lambda x(\cdot) &= (A+BF)x(\cdot)+Bu(\cdot)\\ y(\cdot) &= (C+DF)x(\cdot)+Du(\cdot).\end{aligned}$$

This control action is typically used to modify the dynamics of the system and, more particularly, the eigenvalues of the matrix $A+BF$. If the pair (A,B) is single input and controllable, then it is well known that one can arbitrarily choose the closed-loop spectrum, or, equivalently, the characteristic polynomial of $A+BF$, and the feedback matrix F is unique [13]. For such systems there always exists a similarity transformation T yielding a transformed pair $(A_t,B_t):=(T^{-1}AT,T^{-1}B)$, which is in controller canonical form

$$A_t=\begin{bmatrix}-a_{n-1} & -a_{n-2} & \cdots & -a_1 & -a_0\\ 1 & 0 & & & \\ & \ddots & \ddots & & \\ & & \ddots & \ddots & \\ & & & 1 & 0\end{bmatrix},\quad B_t=\begin{bmatrix}1\\0\\ \vdots\\ \vdots\\0\end{bmatrix}.$$

The characteristic polynomial of A_t (and hence of A) is then given by

$$a(z):=\det(zI_n-A_t)=z^n+a_{n-1}z^{n-1}+\cdots+a_1z+a_0.$$

Transforming also the feedback $F_t:=FT$ yields the transformed closed-loop matrix $A_t+B_tF_t=T^{-1}(A+BF)T$, which has the same eigenvalues as $A+BF$ since both matrices are related by a similarity transformation. It is clear that by choosing

$$F_t=[a_{n-1}-f_{n-1},\ \ldots\ ,a_0-f_0]$$

the matrix $A_t-B_tF_t$ has the same form as A_t, but with a_i replaced by f_i. Hence

$$f(z):=\det(zI_n-(A_t+B_tF_t))=z^n+f_{n-1}z^{n-1}+\cdots+f_0.$$

This discussion shows that the closed-loop eigenvalues of every controllable single input pair can be arbitrarily assigned and at the same time suggests a simple method for constructing F. A computational method for implementing this technique is due to Ackermann and is implemented in the MATLAB Control Toolbox in the function `acker`. The algorithm first constructs the controllability matrix

$$T_1=\left[B\ AB\ \cdots\ A^{n-1}B\right],$$

which transforms (A,B) to

$$\begin{aligned}[\ A_1\mid B_1\ ] &:= \left[\ T_1^{-1}AT_1\mid T_1^{-1}B\ \right]\\ &= \begin{bmatrix} & & & -a_0 & 1\\ 1 & & & \vdots & 0\\ & \ddots & & \vdots & \vdots\\ & & 1 & -a_{n-1} & 0\end{bmatrix}\end{aligned}$$

giving the characteristic polynomial of A. A further transformation

$$T_2=\begin{bmatrix}1 & a_{n-1} & \cdots & a_1\\ & \ddots & \ddots & \vdots\\ & & \ddots & a_{n-1}\\ & & & 1\end{bmatrix}$$

yields the requested form (A_t,B_t). This algorithm is a typical example of the use of canonical forms requiring a badly conditioned transformation ($T=T_1T_2$ in this case). It is well known that the transformation T_1 can be badly conditioned. Let us take a random (A,B) pair with only ten states and one input. If we then try to assign random eigenvalues that are symmetric with respect to the real axis, Ackermann's method of the MATLAB Control Toolbox yields a feedback matrix of poor quality. Indeed, when recomputing the eigenvalues of $(A+BF)$ one finds only three digits of accuracy! One can show that the main reason for this loss of accuracy is the poor condition number of the controllability matrix T_1. The culprit is thus the choice of method and not the implementation of the algorithm. Moreover, if one assigns eigenvalues for a 20th-order system, not a single digit of accuracy can be expected anymore, and the stabilizing feedback does not work at all!

There are now better algorithms that have been shown to possess good numerical properties [14], [15] with complexity comparable to that of Ackermann's method. Moreover, the sensitivity is now much better understood [5], and reliable implementations are available in SLICOT [1].

Invariant Subspaces and Design Problems

Many design problems in systems and control can be reduced to the solution of an eigenvalue problem [16], [17] or, more precisely, to the calculation of certain spaces spanned by eigenvectors and generalized eigenvectors. It is crucial to construct orthogonal bases for such spaces, and the basic decomposition that leads to this construction is the generalized Schur form for general $n \times n$ pencils $\lambda E - A$. Of principal interest are regular pencils (that is, with $\det(\lambda E - A)$ not identically zero), for which one can define generalized eigenvalues as the roots of $\det(\lambda E - A) = 0$. We state the decomposition for the case that E and A are real matrices.

Generalized Schur Form [18]

There exist orthogonal transformations Q and Z that transform a regular pencil $\lambda E - A$ to

$$Q^T(\lambda E - A)Z = \lambda E_S - A_S,$$

where E_S is upper triangular and A_S is block upper triangular with a 1×1 diagonal block corresponding to each real generalized eigenvalue and a 2×2 diagonal block corresponding to each pair of complex conjugate generalized eigenvalues (such matrices are called quasi-triangular). This decomposition exists for every ordering of eigenvalues in the quasi-triangular form. ■

If $E = I$ one recovers the standard (quasi-triangular) Schur decomposition $A_S = U^T AU$ based on an orthogonal similarity transformation by taking $U = Z = Q$. Notice that if E is invertible one also has

$$Q^T AE^{-1}Q = A_S E_S^{-1}, \quad Z^T E^{-1}AZ = E_S^{-1}A_S,$$

which are both quasi-triangular matrices. Then Q and Z of the generalized Schur form can be obtained from the standard real Schur forms of AE^{-1} and $E^{-1}A$, but this detour should be avoided when E is badly conditioned. One of the most important uses of this form is the computation of orthogonal bases for eigenspaces. Consider a (block) triangular decomposition where we partition the invertible matrix X conformably as in

$$X^{-1}AX = \begin{bmatrix} A_{11} & A_{12} \\ 0 & A_{22} \end{bmatrix}, \quad X = [X_1 X_2].$$

Then $AX_1 = X_1 A_{11}$, which implies that the range space $\mathcal{X} := ImX_1$ of the matrix X_1 satisfies the invariant subspace condition

$$A\mathcal{X} \subset \mathcal{X}.$$

When X is an orthogonal matrix (as in the Schur decomposition), the columns of X_1 are orthogonal. The corresponding concept for the generalized eigenvalue problem $\lambda E - A$ is that of a deflating subspace defined by the condition

$$\dim(A\mathcal{X} + E\mathcal{X}) = \dim \mathcal{X}.$$

For invertible E this condition is equivalent to $E^{-1}A\mathcal{X} \subset \mathcal{X}$ and hence each deflating subspace of $\lambda E - A$ is an invariant subspace of $E^{-1}A$. The first k columns of the right transformation Z [18] are therefore an orthogonal basis for a deflating subspace of the pencil $\lambda E - A$. We refer to [18], [3], and [19] for a rigorous discussion. The use of these eigenspaces in control shows up in the solution of several matrix equations.

We illustrate this technique with a standard eigenvalue problem (that is, $E = I$). Suppose one wants to solve the $q \times p$ quadratic matrix equation

$$M_{21} - XM_{11} + M_{22}X - XM_{12}X = 0 \tag{13}$$

for the $q \times p$ matrix X. This equation is equivalent to

$$\begin{bmatrix} I_p & 0 \\ -X & I_q \end{bmatrix} \begin{bmatrix} M_{11} & M_{12} \\ M_{21} & M_{22} \end{bmatrix} \begin{bmatrix} I_p & 0 \\ X & I_q \end{bmatrix} = \begin{bmatrix} \hat{M}_{11} & \hat{M}_{12} \\ 0 & \hat{M}_{22} \end{bmatrix}, \tag{14}$$

where $\hat{M}_{11} := M_{11} + M_{12}X, \hat{M}_{12} = M_{12}, \hat{M}_{22} := M_{22} - XM_{12}$, and $\hat{M}_{21} = 0$. But (14) is a similarity transformation on the $(p+q) \times (p+q)$ matrix M partitioned in the four blocks $M_{ij}\ i = 1, 2,\ j = 1, 2$. The block triangular decomposition says that the eigenvalues of M are the union of those of $\hat{M}_{11}$ and of $\hat{M}_{22}$ and that the columns of $\left[\begin{smallmatrix} I_p \\ X \end{smallmatrix}\right]$ span an invariant subspace of the matrix M corresponding to the p eigenvalues of $\hat{M}_{11}$ [17]. Let us suppose for simplicity that M is simple, that is, that it has distinct eigenvalues. Then every invariant subspace of a particular dimension p is spanned by p eigenvectors of M. Therefore, let $\left[\begin{smallmatrix} X_{11} \\ X_{12} \end{smallmatrix}\right]$ be a matrix whose columns are p eigenvectors of M, and thus is a basis for the corresponding invariant subspace. If, moreover, X_{11} is invertible then the columns of $\left[\begin{smallmatrix} I_p \\ X \end{smallmatrix}\right]$ with $X = X_{21}X_{11}^{-1}$ span the same subspace and hence X is a solution of the quadratic matrix equation (13). One shows that the eigenvalues corresponding to the selected eigenvectors are the eigenvalues of $\hat{M}_{11}$ after applying the transformation (14). This approach actually yields *all solutions* X provided M is simple and the matrices X_{11} defined above are invertible. However, the approach requires the computation of all eigenvectors, which are obtained from a diagonalizing similarity transformation. One shows that when M has repeated eigenvalues, one should compute its Jordan canonical form to find all solutions of the quadratic matrix equation (13) [7]. The disadvantage of this approach is that it involves the construction of a transformation T that may be badly conditioned.

Every invariant subspace also has an orthogonal basis, and in general these basis vectors are not eigenvectors, since eigenvectors need not be orthogonal to each other. Such a basis is exactly obtained by the Schur decomposition (9). One can always compute an orthogonal similarity transformation that quasi-triangularizes the matrix M. If we then partition the triangular matrix with a $p \times p$ leading block as in

$$\begin{bmatrix} U_{11} & U_{12} \\ U_{21} & U_{22} \end{bmatrix}^T \begin{bmatrix} M_{11} & M_{12} \\ M_{21} & M_{22} \end{bmatrix} \begin{bmatrix} U_{11} & U_{12} \\ U_{21} & U_{22} \end{bmatrix} = \begin{bmatrix} \tilde{M}_{11} & \tilde{M}_{12} \\ 0 & \tilde{M}_{22} \end{bmatrix}, \tag{15}$$

then it follows that the columns of $\begin{bmatrix} U_{11} \\ U_{21} \end{bmatrix}$ also span an invariant subspace of M, while the columns of $\begin{bmatrix} I_p \\ X \end{bmatrix}$ with $X = U_{21}U_{11}^{-1}$ span the same subspace, provided U_{11} is invertible [17]. The advantage of this approach is that it uses numerically reliable coordinate transformations in (15), while the disadvantage is that only one invariant subspace is directly obtained that way. In certain applications, it turns out that one needs only a particular invariant subspace. Typical examples arise in applications involving continuous time systems:

- the algebraic Riccati equation $XBR^{-1}B^TX - XA - A^TX - Q = 0$ from optimal control. Here the relevant matrix is
$$M = \begin{bmatrix} A & -BR^{-1}B^T \\ -Q & -A^T \end{bmatrix}$$
and the matrix $\tilde{M}_{11}$ must contain all eigenvalues of M in the open left-half plane
- the Lyapunov equation $AX + XA^T + Q = 0$ occurring in stability analysis. Here
$$M = \begin{bmatrix} -A^T & 0 \\ Q & A \end{bmatrix}$$
and $\hat{M}_{11} = -A^T$
- the Sylvester equation $AX - XB + C = 0$ where
$$M = \begin{bmatrix} B & 0 \\ C & A \end{bmatrix}$$
and $\hat{M}_{11} = B$.

In each of these cases one has a well-defined spectrum in mind for the matrix $\tilde{M}_{11}$ after transformation, and so only one invariant subspace must be computed. We point out that Lyapunov and Sylvester equations can be viewed as special linear cases of the quadratic Riccati equation, which is extensively discussed in [17] and [7]. The efficient calculation of the linear equations is discussed in [20] and is based on the Schur forms of A and B. This so-called Schur approach was proposed in a number of papers (see references in [7]) and is now the recommended technique for solving these problems, although improvements are still being found in this area. The generalized eigenvalue counterpart involves deflating subspaces and arises in applications involving discrete-time systems and generalized state space systems. We refer to [7] for more details.

Structured Matrix Problems

In systems and control theory there are many linear algebra problems that have a special structure and for which one would like to have numerical algorithms that not only have a small backward error (in the sense given earlier) but at the same time preserve the structure of the problem. The following examples are typical:

- Hamiltonian and symplectic eigenvalue problems originating from optimal control. The optimal control, optimal filtering, and spectral factorization problems can all be reduced to generalized eigenvalue problems with the specific structure
$$\lambda \begin{bmatrix} 0 & I & 0 \\ -I & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} - \begin{bmatrix} 0 & A & B \\ A^T & Q & S \\ B^T & S^T & R \end{bmatrix};$$
$$\lambda \begin{bmatrix} 0 & I & 0 \\ F^T & 0 & 0 \\ G^T & 0 & 0 \end{bmatrix} - \begin{bmatrix} 0 & F & G \\ I & Q & S \\ 0 & S^T & R \end{bmatrix}.$$
Although stable algorithms have been developed for these problems [21], [22], [19], it is only recently that structure-preserving stable algorithms were obtained as well [23].
- Matrices with Toeplitz and Hankel structure
$$T = \begin{bmatrix} T_1 & T_2 & \cdots & T_n \\ T_2 & \ddots & & \vdots \\ \vdots & & \ddots & \vdots \\ T_n & \cdots & \cdots & T_1 \end{bmatrix}, H = \begin{bmatrix} H_1 & H_2 & \cdots & H_n \\ H_2 & & \iddots & \vdots \\ \vdots & \iddots & & \vdots \\ H_n & \cdots & \cdots & H_{2n-1} \end{bmatrix}$$
often arise in identification and least squares fitting of system models to input/output or spectral data collected from system measurements. For such problems there are fast algorihms that have been proven to be backward stable [24], [25], but unfortunately not in a structured sense [26]. There is still much work going on in this area [27].
- Eigenvalue problems with cyclic structure
$$\lambda\mathcal{E} - \mathcal{A} := \begin{bmatrix} -A_1 & \lambda E_1 & & \\ & \ddots & \ddots & \\ & & -A_{K-1} & \lambda E_{K-1} \\ \lambda E_K & & & -A_K \end{bmatrix} \tag{16}$$
appear in the study of periodic systems. There are now stable structure-preserving algorithms for a large subclass of these problems (see [28] for a survey). We discuss this problem further below.
- Polynomial models can be reduced to generalized state-space models in which the poles and zeros are described by pencils of the type
$$\begin{bmatrix} \lambda I & -I & & \\ & \ddots & \ddots & \\ & & \lambda I & -I \\ P_0 & \ldots & P_{K-2} & P_{K-1} + \lambda P_K \end{bmatrix}. \tag{17}$$
There are only partial results available for such problems as we indicate below.

We now take a closer look at the last two items, which illustrate that exploiting structure allows one to significantly reduce the complexity of the algorithms. If, moreover, the algorithms are backward stable in a structured sense (that is, the backward error has the same structure as the original data) then the sensitivity can only improve relative to unstructured problems, and often does so significantly [26].

Periodic Eigenvalue Problems

These problems typically occur in the context of discrete-time, linear time-varying systems

$$\begin{aligned} E_k x_{k+1} &= A_k x_k + B_k u_k, \\ y_k &= C_k x_k + D_k u_k, \end{aligned} \tag{18}$$

arising, for example, from the discretization of a continuous-time periodic system. A periodic system is a set of difference equations (18) where the coefficient matrices vary periodically with time, that is, $E_k = E_{k+K}$ for all k and likewise for A, B, C, and D. The period is the smallest value of K for which these identities hold. It was shown in [29] that a periodic system of period K has a well-defined solution for appropriately defined boundary conditions provided the pencil $\lambda\mathcal{E} - \mathcal{A}$ in (16) is regular. In case E_k is invertible for all k, it follows that, for some constant c,

$$\det(\lambda\mathcal{E} - \mathcal{A}) = c \det\left(\lambda^K I_n - \Phi_{K,1}\right),$$

where $\Phi_{K,J} := E_K^{-1} A_K \cdots E_2^{-1} A_2 E_J^{-1} A_J$ and $\Phi_{K,1}$ is the so-called monodromy matrix of the periodic system. For more details on the relation between generalized eigenvectors and eigenvalues of these pencils we refer to [28]. A key decomposition for computing these generalized eigenvalues and eigenvectors is the periodic Schur form, which we state again for the case of real matrices.

Periodic Schur Form [30], [31]

Let the $n \times n$ matrices E_k and A_k, $k = 1, \ldots, K$ be such that the pencil $\lambda\mathcal{E} - \mathcal{A}$ is regular. Then there exist orthogonal transformations Q_k and Z_k, $k = 1, \ldots, K$ such that

$$\mathcal{Q}^T(\lambda\mathcal{E} - \mathcal{A})\mathcal{Z} = \begin{bmatrix} -\hat{A}_1 & \lambda\hat{E}_1 & & \\ & \ddots & \ddots & \\ & & -\hat{A}_{K-1} & \lambda\hat{E}_{K-1} \\ \lambda\hat{E}_K & & & -\hat{A}_K \end{bmatrix},$$

where

$$\mathcal{Q} := \mathrm{diag}\{Q_1, \ldots, Q_K\}, \quad \mathcal{Z} := \mathrm{diag}\{Z_1, \ldots, Z_K\},$$

and the transformed matrices $\hat{A}_k$ and $\hat{E}_k$ are all upper triangular, except for one matrix—say, $\hat{A}_1$—that is quasi-triangular. ■

The relation with the standard Schur form is that if the matrices E_k are invertible, then the monodromy matrix $\Phi_{K,1}$ is transformed by the orthogonal similarity Z_1 to its Schur form

$$\begin{aligned} \hat{\Phi}_{K,1} : &= \hat{E}_K^{-1}\hat{A}_K \cdots \hat{E}_1^{-1}\hat{A}_1 \\ &= Z_1^T\left(E_K^{-1}A_K \cdots E_1^{-1}A_1\right)Z_1 = Z_1^T\Phi_{K,1}Z_1. \end{aligned}$$

Since all matrices except one are triangular, it follows that all transformed monodromy matrices $\hat{\Phi}_{K+k-1,k}$ are quasi-triangular as well, and with the same ordering of eigenvalues. From the generalized Schur form it follows that the ordering of the eigenvalues can be chosen arbitrarily and hence that there exists a periodic Schur form associated with every eigenvalue ordering. An important feature here is that the backward errors can be completely mapped back to the original data, that is, we compute the periodic Schur form of a slightly perturbed pencil of the same structure as (16) but with perturbed matrices $A_k + \Delta A_k$ and $E_k + \Delta E_k$, $k = 1, \ldots, K$. This property is known as structured backward stability.

The transformations Z_k and Q_k can also be applied directly to the system (18). Defining a new state $\hat{x}_k := Z_k^T x_k$ and multiplying the state equation of (18) by Q_k^T, yields the equivalent system

$$\begin{aligned} \hat{E}_k\hat{x}_{k+1} &= \hat{A}_k\hat{x}_k + \hat{B}_k u_k, \\ y_k &= \hat{C}_k\hat{x}_k + D_k u_k, \end{aligned} \tag{19}$$

where $\hat{B}_k := Q_k^T B_k$, $\hat{C}_k := C_k Z_k$, and $\hat{E}_k$ and $\hat{A}_k$ are upper triangular, except for $\hat{A}_1$, which is quasi-triangular. This system is expressed in a very special coordinate system, namely, the lower equation in (19) (or the lower two equations if $\hat{A}_1$ has a bottom 2×2 block) is now decoupled from the rest of the system. Since the ordering of the eigenvalues in the Schur form can always be chosen arbitrarily, one can choose this decoupled subsystem to be the subsystem with the smallest eigenvalue in absolute value and hence the easiest to integrate numerically [30]. Once the lower component of the state has been computed, it can be substituted into the next component, which is then decoupled from the rest of the system, and so on. This coordinate system is thus appealing for simulation purposes.

The periodic Schur form has several other applications in control problems involving periodic discrete-time systems [32], [28]. In the optimal control of such a periodic system one considers the problem

$$\begin{aligned} \text{Minimize over } u_k: \quad & J = \sum_{k=1}^{\infty} z_k^T Q_k z_k + u_k^T R_k u_k \\ \text{subject to}: \quad & H_k z_{k+1} = F_k z_k + G_k u_k, \end{aligned}$$

where the matrices Q_k, R_k, F_k, G_k, H_k are periodic with period K. To solve this variational problem, one needs to solve Hamiltonian equations that are periodic homogeneous systems of difference equations (18) in the state z_k and costate λ_k of the system [28]. The correspondences with (18) are

$$x_k := \begin{bmatrix} \lambda_k \\ z_k \end{bmatrix}, \; E_k := \begin{bmatrix} -G_k R_k^{-1} G_k^T & H_k \\ F_k^T & 0 \end{bmatrix}, \; A_k := \begin{bmatrix} 0 & F_k \\ H_k^T & Q_k \end{bmatrix}.$$

For finding periodic solutions to the underlying periodic Riccati equation one has to find the stable invariant subspaces of the monodromy matrices $\Phi_{K+k-1,k}$ [33]. Clearly, the periodic Schur form is useful here as well as the reordering of eigenvalues [30].

In pole placement for periodic systems, the periodic Schur form and reordering are useful for extending Varga's pole placement algorithm [15] to periodic systems. Additional applications of the periodic Schur form are the solution of periodic Lyapunov and Sylvester equations [34], [29]. Since these are special cases of periodic Riccati equations, they can also be solved by using the periodic Schur form. The computation of zeros of periodic systems is related to that of finding a minimal realization of such systems, and appropriate algorithms need to be designed [28].

Polynomial Models

The zero structure of an $m \times n$ polynomial matrix

$$P(\lambda) := P_0 + \lambda P_1 + \cdots + \lambda^K P_K$$

is the set of zeros of the polynomials $e_i(\lambda)$ defined by means of the Smith canonical form of $P(\lambda)$ given by

$$\left[\begin{array}{ccc|c} e_1(\lambda) & & & \\ & \ddots & & 0_{r,n-r} \\ & & e_r(\lambda) & \\ \hline & 0_{m-r,r} & & 0_{m-r,n-r} \end{array}\right] := M(\lambda)P(\lambda)N(\lambda),$$

where each polynomial $e_i(\lambda)$ divides $e_{i-1}(\lambda)$ for $i = 2, \ldots, r$ and $M(\lambda)$ and $N(\lambda)$ are unimodular transformations, that is, polynomial matrices with constant but nonvanishing determinant. The integer r is the normal rank of the polynomial matrix, which is the rank of the polynomial matrix evaluated at every value λ in the complex plane, except the set of zeros.

Although the construction of the Smith form is based on elementary column and row operations, these transformations are numerically unstable as shown in the following small example. A typical row operation to be performed in the Smith form calculation of

$$P(\lambda) = \begin{bmatrix} \delta & \lambda \\ \lambda & \lambda \end{bmatrix},$$

where δ is small but nonzero, is

$$\begin{bmatrix} 1 & 0 \\ -\lambda/\delta & 1 \end{bmatrix} P(\lambda) = \begin{bmatrix} \delta & \lambda \\ 0 & \lambda - \lambda^2/\delta \end{bmatrix}.$$

The coefficients in the left transformation matrix, and therefore also the numerical errors induced by them, diverge when δ becomes small. For this low-dimensional example the effect seems tractable, but in general an accumulation of such effects renders the reduction to Smith canonical form numerically unstable.

The alternative here is to compute the zero structure as the generalized eigenvalues of the linearized form (17). It is known [35] that all of the information contained in the Smith form of $P(\lambda)$ can be retrieved from the Kronecker canonical form [16] of (17) and that there exists an orthogonal decomposition that computes this information in a stable manner. However, the backward error perturbs the zero entries of (17), causing potential problems. It is shown in [35] that errors can be mapped back to the individual matrices P_i, $i = 0, \ldots, K$, but the best obtainable bounds

$$\|\Delta P_k\| \le \epsilon c \max_i \|P_i\|_2$$

may be unsatisfactory when the coefficients P_i have strongly varying norms. In [36] a second-order polynomial matrix is given for which the generalized Schur form applied to (17) gives eigenvalues that are not computed in a backward stable manner.

Today, there are several algorithms available that deal directly with polynomial matrices rather than use reductions to generalized eigenvalue problems of the type (17) [37]. Some of these techniques are based on elimination algorithms involving unimodular transformations. As indicated earlier, these algorithms may have very poor stability properties on particular examples. Other techniques are based on the solution of structured linear systems of Toeplitz or Hankel type and generally have good numerical properties [38]. Currently, there is no detailed numerical analysis of these algorithms, and they should therefore be used with care.

Concluding Remarks

In this article we surveyed several numerical methods arising in systems and control and pointed out general principles that lead to numerically reliable algorithms for solving a large range of problems in this area. Algorithms that use orthogonal or unitary transformations were emphasized because they often lead to reliable numerical algorithms. We focused on those problems for which good algorithms are available, and for which robust software implementations are available. It is crucial to have robust numerical software available to implement these ideas, and the NICONET project aims at providing exactly that. Moreover, the underlying software library SLICOT is freely available for noncommercial use. For more details we refer to [1].

Acknowledgment

This work was supported by the National Science Foundation under grant CCR-99-12415. Parts of this article present research results of the Belgian Programme on Inter-university Poles of Attraction, initiated by the Belgian State, Prime Minister's Office for Science, Technology, and Culture.

References

[1] S. Van Huffel, V. Sima, A. Varga, S. Hammarling, and F. Delebecque, "High performance numerical software for control," *IEEE Control Syst. Mag.,* vol. 24, no. 1, pp. 60–76, Feb. 2004.

[2] R. Horn and C. Johnson, *Matrix Analysis*. Cambridge: Cambridge Univ. Press, 1985.

[3] G. Golub and C. Van Loan, *Matrix Computations,* 3rd ed. Baltimore, Maryland: Johns Hopkins Univ. Press, 1996.

[4] N. Higham, *Accuracy and Stability of Numerical Algorithms,* 2nd ed. Philadelphia, PA: SIAM, 2002.

[5] N. Higham, M. Konstantinov, V. Mehrmann, and P. Petkov, "The sensitivity of computational control problems," *IEEE Control Syst. Mag.,* vol. 24, no. 1, pp. 28–43, Feb. 2004.

[6] G.W. Stewart and J.-G. Sun, *Matrix Perturbation Theory*. New York: Academic, 1990.

[7] R. Patel, A. Laub, and P. Van Dooren, Eds., *Numerical Linear Algebra Techniques for Systems and Control.* Piscataway, NJ: IEEE Press, 1993.

[8] P. Van Dooren and M. Verhaegen, "On the use of unitary state-space transformations," *Linear Algebra and Its Role in Linear System Theory,* B.N. Datta, Ed. Providence, RI: AMS Contemporary Mathematics Series, 1985, vol. 47, pp. 447–463.

[9] C. Bavely and G.W. Stewart, "An algorithm for computing reducing subspaces by block diagonalization," *SIAM J. Num. Anal.,* vol. 16, pp. 359–367, 1979.

[10] B. Gragg and A. Lindquist, "On the partial realization problem," *Linear Algebra Appli.,* vol. 50, pp. 277–319, 1983.

[11] J. Demmel and B. Kågström, "Accurate solutions of ill-posed problems in control theory," *SIAM J. Matrix Anal. Appl.,* vol. 9, no. 1, pp. 126–145, 1988.

[12] J. Demmel and B. Kågström, "The generalized Schur decomposition of an arbitrary pencil $A - \lambda B$: Robust software with error bounds and applications," *ACM Trans. Math. Software,* vol. 19, pp. 160–174, 1993.

[13] W.M. Wonham, *Linear Multivariable Control. A Geometric Approach*. New York: Springer-Verlag, 1974.

[14] G. Miminis and C.C. Paige, "A direct algorithm for pole assignment of time-invariant multi-input linear systems using state feedback," *Automatica,* vol. 24, pp. 343–356, 1988.

[15] A. Varga, "A Schur method for pole assignment," *IEEE Trans. Automat. Contr.,* vol. 26, no. 2, pp. 517–519, 1981.

[16] P. Van Dooren, "The generalized eigenstructure problem in linear system theory," *IEEE Trans. Automat. Contr.,* vol. 26, no. 1, pp. 111–129, 1981.

[17] A. Laub, "Invariant subspace methods for the numerical solution of Riccati equations," in *The Riccati Equation,* S. Bittanti, A. Laub, and J. Willems, Eds. New York: Springer Verlag, 1990.

[18] C. Moler and G. Stewart, "An algorithm for the generalized matrix eigenvalue problem," *SIAM J. Numer. Anal.,* vol. 10, no. 2, pp. 241–256, 1973.

[19] P. Van Dooren, "A generalized eigenvalue approach for solving Riccati equations," *SIAM Sci. Stat. Comp.,* vol. 2, no. 2, pp. 121–135, 1981.

[20] R.H. Bartels and G.W. Stewart, "Solution of the equation $AX + XB = C$," *Commun. ACM,* vol. 15, no. 9, pp. 820–826, 1972.

[21] A. Laub, "A Schur method for solving algebraic Riccati equations," *IEEE Trans. Automat. Contr.,* vol. 24, no. 6, pp. 913–921, 1979.

[22] T. Pappas, A. Laub, and N. Sandell, "On the numerical solution of the discrete time algebraic Riccati equation," *IEEE Trans. Automat. Contr.,* vol. 25, no. 4, pp. 631–641, 1980.

[23] P. Benner, V. Mehrmann, and H. Xu, "A numerically stable, structure preserving method for computing the eigenvalues of real Hamiltonian or symplectic pencils," *Numerische Mathematik,* vol. 78, no. 3, pp. 329–358, 1998.

[24] A. Bultheel, "Error analysis of incoming and outgoing schemes for the trigonometric moment problem," in *Pade Approximation and Its Applications*, M. de Bruin, and H. van Rossum, Eds. (Lecture Notes in Mathematics, vol. 888). Berlin, Germany: Springer, 1981, pp. 100–109.

[25] G. Cybenko, "The numerical stability of the Levinson–Durbin algorithm for Toeplitz systems of equations," *SIAM Sci. Stat. Comp.,* vol. 1, no. 3, pp. 303–319, 1980.

[26] P. Van Dooren, "Structured linear algebra problems in digital signal processing," in *Numerical Linear Algebra, Digital Signal Processing and Parallel Algorithms* (Nato Series F), G. Golub and P. van Dooren, Eds. NewYork: Springer Verlag, 1991, pp. 361–384.

[27] T. Kailath and A. Sayed, Ed., *Fast Reliable Algorithms for Matrices with Structure.* Philadelphia, PA: SIAM, 1999.

[28] A. Varga and P. Van Dooren, "Computational methods for periodic systems—An overview," in *Proc. IFAC PSYCO-2001,* Como, Italy, 2001, pp. 171–176.

[29] J. Sreedhar and P. Van Dooren, "Periodic descriptor systems: solvability and conditionability," *IEEE Trans. Automat. Contr.,* vol. 44, no. 2, pp. 310–313, 1999.

[30] A. Bojanczyk, G. Golub, and P. Van Dooren, "The periodic Schur decomposition. Algorithms and applications," in *Advanced Signal Processing Algorithms, Architectures, and Implementations III* (Proc. SPIE, vol. 1770), F.T. Luk, Ed. Bellingham, WA: 1992, pp. 31–42.

[31] A. Laub and J. Hench. "Numerical solution of the discrete-time periodic Riccati equation," *IEEE Trans. Automat. Contr.,* vol. 39, no. 6, pp. 1197—1210, 1994.

[32] J. Sreedhar and P. Van Dooren, "A Schur approach for solving some periodic matrix equations," in *Systems and Networks: Mathematical Theory and Applications,* U. Helmke, R. Mennicken, and J. Saurer, Eds. (Mathematical Research, vol. 77). Berlin, Germany: Akademie Verlag, 1994, pp. 339–362.

[33] S. Bittanti, P. Colaneri, and G. de Nicolao, "The difference periodic Riccati equation for the periodic prediction problem," *IEEE Trans. Automat. Contr.,* vol. 33, no. 8, pp. 706–712, 1988.

[34] A. Varga, "Periodic Lyapunov equations: some applications and new algorithms," *Int. J. Contr.,* vol. 67, no. 1, pp. 69–87, 1997.

[35] P. Van Dooren and P. Dewilde, "The eigenstructure of an arbitrary polynomial matrix. Computational aspects," *Linear Algebra Applicat.,* vol. 50, pp. 545–580, 1983.

[36] F. Tisseur and K. Meerbergen, "The quadratic eigenvalue problem," *SIAM Rev.,* vol. 43, no. 2, pp. 235–286, 2001.

[37] V. Kucera, *Discrete Linear Control. The Polynomial Equation Approach.* Chichester, U.K.: Wiley, 1979.

[38] Z. Vostry, "New algorithm for polynomial spectral factorization with quadratic convergence I." *Kybernetika*, vol. 11, no. 6, pp. 415–422, 1975.

Paul Van Dooren (vdooren@csam.ucl.ac.be) received the engineering degree in computer science and the doctoral degree in engineering, both from the Katholieke Universiteit te Leuven, Belgium. He was with Philips Research Laboratory Belgium for ten years and subsequently became a professor at the University of Illinois at Urbana-Champaign. He is currently a professor of mathematical engineering at the Université Catholique de Louvain. He received the Householder Award in 1981 and the Wilkinson Prize of Numerical Analysis and Scientific Computing in 1989. He is interested in numerical linear algebra with applications in systems and control theory and in digital signal processing. He can be contacted at Center for Systems Engineering and Applied Mechanics (CESAME) Université Catholique de Louvain, B-1348 Louvain-la-Neuve, Belgium.