Numerical aspects of system and control algorithms

P. Van Dooren

Journal A, Vol. 30, pp. 25-32, 1989.

Abstract

The area of multivariable linear systems is very rich in linear algebra problems and numerous algorithms have been derived in that area. Because of the increasing complexity of the problem being tackled, some of them have become challenging from a numerical point of view as well. The cooperation between researchers in numerical analysis and in linear system theory is increasing significantly the last few years. In this paper we give a survey of recent results of this joint research area.

1 Introduction

Since the very formulation of the numerical problems arising in linear system theory, numerical algorithms have been derived, leading in a later stage to software implementation as well (see e.g.[1] [2] [3]).

The numerical results obtained by some packages were not always satisfactory, especially when used on *tough* cases as e.g. large scale systems or ill-posed problems. The reasons for these failures could be of different types. Sometimes one would implement a straightforward method on computer but straightforward methods very often do not perform so well in the presence of rounding errors. In other cases appropriate methods were chosen but were implemented incorrectly. Finally, in some cases there was a total absence of reliable numerical software or even methods for the problems to be solved, which e.g. led to simplifying assumptions and thus less general algorithms.

It is only recently that these difficulties became apparent because the developed software was being used on tougher problems and the results did not meet the requirements anymore. The last few years numerical analysts have started to focus on the linear algebra problems arising in linear system theory (see e.g. [4]-[17]). Around the same period, researchers in the area of linear system theory turned their attention to advanced numerical software and tried to modify these methods or use their principles for the development of reliable software

dealing with their specific problems (see e.g. [18]-[22]). This cross fertilization has been largely promoted by the availability of software packages (e.g.[23]-[25]) and also by recent developments in numerical linear algebra (e.g. [26]-[28]). The increasing interest in this joint area also appears from special issues and recent conferences (e.g.[29]-[33]). In this paper, we try to give a survey of some recent results and trends in this joint research area. We put the emphasis on the numerical aspects of the problems/algorithms which is e.g. why we also spend some time in going over the numerical tools and techniques in this context (Section 2). In the next section we discuss a number of problems using the previously mentioned techniques. In Section 4 we try to point out some related problems where a collaboration between researchers of the two areas would be fruitfull or has already started.

2 Numerical background

In this section we review some concepts of basic importance in numerical linear algebra by discussing a typical matrix problem that is used in the sequel.

a) *Conditioning, stability and error analysis.*

Let us consider the singular value decomposition of an arbitrary $m \times n$ matrix A with coefficients in R or C [34]

$$A = U \cdot \Sigma \cdot V^* \tag{1}$$

Here U and V are, respectively, $m \times n$ and $n \times n$ unitary matrices and Σ is a $m \times n$ matrix of the form

$$\Sigma = \left[\begin{array}{c|c} \Sigma_r & 0 \\ \hline 0 & 0 \end{array}\right] ; \Sigma_r = \mathrm{diag}\{\sigma_1, \cdots, \sigma_r\} \tag{2}$$

with the *singular value* σ_i being positive and satisfying $\sigma_1 \geq \sigma_2 \cdots \geq \sigma_r > 0$. The computation of this decomposition is, of course, subject to rounding errors. denoting computed quantities by an overbar, we generally have for some *error matrix* E_A :

$$\overline{A} = A + E_A = \overline{U} \cdot \overline{\Sigma} \cdot \overline{V}^* \tag{3}$$

The computed decomposition thus corresponds exactly to a *perturbed* matrix $\overline{A}$. When using the SVD algorithm available in the literature [34], this perturbation can be bounded by [34][35] :

$$\| E_A \| \leq \pi \cdot \epsilon \cdot \| A \| \tag{4}$$

where ϵ is the machine precision and π some quantity depending on the dimensions m and n, but reasonably close to 1 (see also [16]). The error E_A induced

by this algorithm – called *backward error* because it is interpreted as an error on the data – has thus roughly the same norm as the *input error* E_i performed when e.g. reading the data A into the computer. When such a bound exists for the perturbations E_X induced by a numerical algorithm, it is called *backward stable* [35][36]. Notice that backward stability does not guarantee any bounds on the errors in the result $\overline{U}, \overline{\Sigma}$ and $\overline{V}$. This depends indeed on how perturbations on the data (namely $E_A = \overline{A} - A$) affect the resulting decomposition (namely $E_U = \overline{U} - U, E_\Sigma = \overline{\Sigma} - \Sigma$ and $E_V = \overline{V} - V$). This is commonly measured by the condition $\kappa[f(A)]$ of the computed object $X = f(A)$:

$$\kappa[f(A)] = \lim_{\delta \to 0} \sup_{d_2(A,\overline{A})=\delta} \left[\frac{d_1(X, \overline{X})}{\delta} \right] \; ; \; \overline{X} = f(\overline{A}) \tag{5}$$

where d_i (.,.) are distance functions in the appropriate spaces [37]. When $\kappa[f(A)]$ is infinite, the problem of determining X from A is *ill-posed* (against *well-posed*). when $\kappa[f(A)]$ is finite and *relatively large* or *relatively small*, the problem is said to be *badly conditioned* and *well conditioned*, respectively.
Notice that backward stability is a property of an algorithm while conditioning is associated with a problem and the specific data for that problem. The errors E_X in the result depend on both the stability of the algorithm used and the conditioning of the problem solved. A *good* algorithm therefore is supposed to be backward stable since the size of the errors E_X in the result is then mainly affected by the condition of the problem, not by the algorithm. An unstable algorithm, on the other hand, may yield a large error E_X even when the probllem is well conditioned.
Bounds of the type (4) are obtained by an error analysis of the algorithm used (see [35] for the state of the art). The condition of the problem is obtained by a sensitivity analysis (see [35][38][39] for some examples).

b) *Some typical difficulties.*

According to the above criteria a safe procedure seems to be to reformulate the problems arising in linear system theory as a concatenation of subproblems for which numerical stable software is available in the literature. Unfortunately one can not ensure that the stability of the subalgorithms carries over to the stability of the total algorithm. This requires a separate analysis which e.g. could rely on the sensitivity or conditioning of the subproblems. In the next section we show that delicate (i.e. badly conditioned) subproblems should be avoided whenever possible; a few examples are given where a possibly badly conditioned step is circumvented by carefully modifying or completing existing algorithms (see also [27][12][40]).

A second type of difficulty is the ill-posedness of some of the problems occurring in linear system theory [41]. Two approaches can be chosen here.
Either one develops an acceptable perturbation theory for such problems, making use of notions such as *restricted conditioning* which is the conditioning under

perturbations for which a certain property is holding (fixed rank etc. [42][12]). One then looks for restricting assumptions that make the problem well-posed. An other approach is to delay any such *restricting choices* to the end and eventually leave it up to the user what choice to make by looking at the results. The algorithm then provides quantitative measures that help him make this choice (see e.g. [18][20][19]). By this approach one may avoid artificial restrictions of the other approach that sometimes do not respect the practical significance of the problem.

A third possible *pitfall* is that many users almost always prefer fast algorithms to slower ones, while it is also often the case that the slower ones are more reliable (see e.g. [5][78]).

3 Linear system theoretic problems

In this section we survey a number of linear algebra problems arising in linear system theory. The survey is by no means complete since this interdisciplinary area is quite extended and developing fast. Uppercase is used for matrices, lowercase for scalars.

Many of the scalar alorithms we discuss do not extend trivially to the matrix case. If they do, we only mention the matrix case. We only discuss the numerical aspects here; for the system theoretical background, we refer to the literature.

a. Identification ([43][44] and references therein)

a1. *Realization* ([5] and references therein).

Let *R(z)* be a $m \times n$ transfer function of a discrete time causal system, and let its impulse response be given by

$$R(z) \;=\; \sum_{i=0}^{\infty} H_i \; z^{-i} \tag{6}$$

The realization problem is to find the transfer function $R(z)$ in e.g. polynomial description $D(z)\; N^{-1}(z)$ or state space description $R(z) = J + H(zI - F)^{-1}G$, when the impulse response $\{H_i\}$ is given. In a later stage one might also be interested in the poles and zeros of $R(z)$, etc.

In the scalar case this poblem is in fact the Padé approximation problem, for which fast methods exist (see [5][10] for a survey). In [5] it is shown that these methods are unstable and that they all boil down to factorizing the Hankel matrix :

$$H = \begin{bmatrix} H_1 & H_2 & \cdots & H_n \\ H_2 & & & \vdots \\ \vdots & & & \vdots \\ H_n & \cdots & \cdots & H_{2n-1} \end{bmatrix} \tag{7}$$

where n is an upper bound for the degree of $R(z)$. In [5] a stable but slower algorithm is given for finding a polynomial and a state-space description of $R(z)$. A typical state space method based on the Singular Value Decomposition of H was given in [45] and shown to be stable in [21]. As shown in [20][21] the realization $\{\tilde{R}, \tilde{G}, \tilde{H}, \tilde{J}\}$ computed is *balanced*, which has the nice property of being less sensitive to rounding errors in H_i than the models used in [5]. Here a typical example can be given of the first difficulty discussed in Section 2b. For computing the poles of $R(z)$ one could compute its Padé approximation and then compute the zeros of $N(z)$. The construction of this intermediate result can be very sensitive to rounding errors in H_i, which endangers the computation of the zeros of $N(z)$. As reported in [20][21] the method using the eigenvalues of F is less subject to rounding errors.
Recently, the connection of this problem was made with an approximation problem over the unit circle [46]. This link might lead to a sensitivity analysis of the realization problem (see [47][15] and references therein) and also gives new insights in the connection with other problems. It is important to remark here also that the algorithmic approach based on the SVD of H easily extends to the matrix case. This has led to a renewed interest in the matrix Padé approximation problem and the partial realization problem [73].

a2. *Linear prediction*([48] and references therein)
The problem here is to model a given time signal $\{y_k\}$ with z transform $Y(z)$ as the output of a linear time invariant system

$$Y(z) = \frac{g}{a(z)}.U(z) \tag{8}$$

where $a(z) = 1 + a_1\, z^{-1} + \cdots a_p\, z^{-p}$ and $U(z)$ is the z transform of a periodic pulse train or of white noise, depending on the application [48]. The so-called autocorrelation method converts this problem to one of factorizing or inverting the positive definite Toeplitz matrix

$$T = \begin{bmatrix} r(0) & \cdots & r(p) \\ \vdots & \ddots & \vdots \\ r(p) & \cdots & r(0) \end{bmatrix} ; r(i) = \sum_{n=-\infty} y_n y_{n+i} \tag{9}$$

where $r(i)$ is the autocorrelation function of the signal $\{y_k\}$. The Choleski factorization of this matrix requires $O(n^3)$ operations, while the Levinson-Durbin algorithm only requires $O(n^2)$ operations by efficiently exploiting the Toeplitz structure of T. For quite some time the stability and conditioning issues of this approach have been a point of controversary (see [8]) until recently a numerical treatment of the problem has shown that the algorithm is reasonably stable, but that the problem is quite often badly conditioned [8][49]. We ought to notice that the problem (8) is in fact an approximation problem, whose solution is given by

the above Toeplitz factorization. The error analysis of [8][49] only refers to the errors in the factorization, not to the approximation error. We point out that the *lattice algorithm* of Itakura & Saito [79] – which computes the same Choleski factor by directly working on the data $\{y_i\}$ arranged in a Toeplitz matrix – is also stable in the same sense [80]. Moreover, it avoids the construction of the Toeplitz matrix T which could lead to problems in extremely badly conditioned cases (see [80] for more details).

a3. *Spectral factorization* (see [43][50] and references therein).

An alternative to (8) is to allow for a more general type of transfer function between $Y(z)$ and $U(z)$:

$$Y(z) = g \cdot \frac{b(z)}{a(z)} \cdot U(z) \tag{10}$$

This now can be converted to a spectral factorization problem [43][50] of the rational transfer function

$$\Phi(z) = R_p(z) + R_p(z^{-1}) = g^2 \, \frac{b(z^{-1})b(z)}{a(z^{-1})a(z)} \tag{11}$$

where $R_p(z)$ is a *p-th* order approximation of the impulse response $R(z)$:

$$R(z) \approx \frac{r(0)}{2} + r(1)\, z^{-1} + r(2)\, z^{-2} + \cdots \tag{12}$$

Again here the original problem is one of approximation for which (11)(12) is a proposed solution. A polynomial approach would yield $a(z)$ during the realization step (12) and $b(z)$ during the spectral factorization step, which can be converted to a polynomial spectral factorization problem (see e.g. [51]) :

$$\Phi_N(z) = g^2 \, b(z^{-1}) \, b(z); \tag{13}$$

where

$$\Phi_N(z) = \Phi_p z^{-p} + \cdots + \Phi_1 z^{-1} + \Phi_0 + \Phi_1 z + \cdots + \Phi_p z^p \tag{14}$$

is derived from $\Phi(z)$.

Bauer has shown that $b(z)$ can be obtained from the rows of the Choleski factor B of the infinite (positive definite) Toeplitz matrix

$$T_\Phi = \begin{bmatrix} \Phi_0 & \Phi_1 & \cdots & \Phi_p & 0 \\ \Phi_1 & \Phi_0 & \ddots & & \ddots \\ \vdots & \ddots & \ddots & & \\ \Phi_p & & & & \\ & \ddots & & & \\ 0 & & & & \end{bmatrix} = B^T B;\ B = \begin{bmatrix} b_0^{(1)} & b_1^{(1)} & \cdots & b_p^{(1)} & & \\ & b_0^{(2)} & b_1^{(2)} & \cdots & b_p^{(2)} & 0 \\ & & \ddots & \ddots & & \ddots \\ & 0 & & & & \end{bmatrix} \tag{15}$$

Unfortunately, the convergence of $b^{(i)}(z)$ to $b(z)$ and the conditioning of the solution $b(z)$ is rather bad when some of the roots of $b(z)$ get close to the unit circle. The same problems occur with the improved Bauer algorithm [52] and with Vostry's Newton type scheme [53] :

$$b^{(i+1)}(z) = \frac{1}{2}[b^{(i)}(z) + x^{(i)}(z)] \tag{16}$$

where

$$b^{(i)}(z^{-1})x^{(i)} + b^{(i)}(z).x^{(i)}(z^{-1}) = 2\Phi_N(z) \tag{17}$$

The solution of (16)(17) requires $O(p^2)$ operations per iteration step and the convergence of $b^{(i)}(z)$ to $b(z)$ is quadratic when no roots of $b(z)$ are close to the unit circle. The stability of Vostry's method is good since it is a method of iterative refinement using the original data $\phi_N(z)$ for computing the correction [35]. A third method - suffering from similar problems of convergence - is the one described by Henrici using the FFT algorithm [54], but it also allows for an implicit stability criterion.

A completely different and conceptually very simple approach is to compute the roots of $z^p \cdot \phi_N(z)$ and perform the factorization by grouping the roots inside the unit circle. This method is quite fast – $O(n^2)$ operations independently of the location of the zeros – and backward stable when carefully implementing the deflations [55]. A similar approach in the state space domain has been developed recently and is based on the computation of a stable deflating subspace of the pencil [40]

$$z \begin{bmatrix} 0 & 0 & 0 \\ 0 & F' & 0 \\ 0 & G' & 0 \end{bmatrix} - \begin{bmatrix} F & 0 & G \\ 0 & I & H' \\ H & 0 & J + J' \end{bmatrix} \tag{18}$$

where $R_p(z) = H(zI_p - F)^{-1}G + J$. Backward stable software for solving this generalized eigenvalue problem can also be found in [40]. The method also applies to the matrix case and avoids the construction of $\phi_N(z)$. Especially in the matrix case, this approach is to be preferred over the matrix generalization of (15) and (16-17) – such as described in [81] – since those then imply computational schemes that preclude any guarantee of numerical stability.

b. *Analysis*

The results here hold for both discrete time (where λ will stand for the shift operator z) and the continuous time case (where λ will stand for the differential operator D). The transfer function $R(\lambda)$ is given by a polynomial representation $V(\lambda)T^{-1}(\lambda)U(\lambda) + W(\lambda)$ or by a state space model $C(\lambda I - A)^{-1} B + D$.

b1. *Polynomial case.* (see [51] and references therein)

One is interested in a number of structural properties of the transfer function $R(\lambda)$ such as poles, zeros, decoupling zeros etc. In the scalar case – i.e. $\{T(\lambda), U(\lambda), V(\lambda), W(\lambda)\}$ are then scalar polynomials – all this can be found with a GCD extraction routine and a rootfinder, for which reliable methods exist ([51][55][28][54] and references therein). In the matrix case the problem becomes much more complex and the basic method for GCD extraction - the Euclidean algorithm - becomes unstable (see [12]). Moreover more structural elements (null spaces etc.) are added to the picture, which makes the polynomial approach less attractive than the state space approach [12][56][57].

b2. *State-space approach* (see [41][12] and references therein)

The structural properties of interest are poles and zeros of $R(\lambda)$, decoupling zeros, controllable and unobservable subspace, supremal (A, B)-invariant and controllability subspaces, factorizability of $R(\lambda)$, left and right null spaces of $R(\lambda)$.

These concepts play a fundamental role in several design problems and have received considerable attention over the last few years [9][12][13][16][18][19][20][30][56]. In [12][82] it is shown that all the concepts mentioned above can be considered as generalized eigenstructure problems and that they can be computed via the Kronecker canonical form of the pencils :

$$\begin{bmatrix} \lambda I - A \end{bmatrix} \qquad \begin{bmatrix} \lambda I - A \mid B \end{bmatrix}$$
$$\left[\begin{array}{c} \lambda I - A \\ \hline -C \end{array}\right] \quad ; \quad \left[\begin{array}{c|c} \lambda I - A & B \\ \hline -C & D \end{array}\right] \tag{19}$$

or from other pencils derived therefrom. Backward stable software was also derived to compute the Kronecker structure of an arbitrary pencil [27][83].
A remaining problem here is that several of the structural elements listed above may be ill-posed in some cases and that one has to develop the notion of restricted conditioning in these cases (see [12]). Sensitivity results in this area are still sparse but are slowly emerging [27][84].
A completely different approach is to reformulate the problem as an approximation or optimization problem for which *quantitative measures* are derived, leaving the final choice to the user. Results in this vein are obtained for controllability, observability [20][13], (almost) (A, B)-invariant and controllability subspaces [18][84].

c. *Design*

Some of the problem treated here could as well be included in Section 3b but were moved to this paragraph because of their design implication.

c1. *Lyapunov and Riccati equations.*

The Lyapunov equation in the continuous and discrete time case :

$$\begin{aligned} AP_c + P_cA' &= -BB' & \qquad A'P_o + P_oA &= -C'C \\ FP_cF' - P_c &= -GG' & \qquad F'P_oF - P_o &= -H'H \end{aligned} \tag{20}$$

frequently occur in analysis/design problems of linear system theory. The solution of the matrices P_c and P_o of (20) can be obtained via the Schur decomposition of the matrices A and F [4][7].

The stability of the method has been analyzed in [7]. Although only a *weak stability* is obtained, this is satisfactory in most cases. When A and F are stable, the solution of the above equations are also equal to the grammians P_c (T) and P_o (T) for $T = \infty$:

$$\begin{aligned} P_c(T) &= \int_0^T e^{At}BB'\, e^{A't}\, dt \quad ; \quad P_o(T) = \int_0^T e^{A't}C'Ce^{At}dt \\ P_c(T) &= \sum_{k=0}^T A^kBB'\, A'^k \quad ; \quad P_o(T) = \sum_{k=0}^T A'^kC'CA^k \end{aligned} \tag{21}$$

These can be used via some additional transformations (see [20][21][58]) to compute so-called *balanced* realizations $\{\tilde{A}, \tilde{B}, \tilde{C}\}$ and $\{\tilde{F}, \tilde{G}, \tilde{H}\}$ in the continuous time and discrete time case, respectively. For these realizations both P_o and P_c are equal and diagonal. These realizations have some nice sensitiity properties with respect to poles, zeros, truncation errors in digital filter implementations etc. [20][21][59]. They are therefore recommended whenever the choice of realization is left to the user.

When A and F are not stable one can still use the *finite range* grammians (21) for $T < \infty$ for the purpose of balancing [20]. A reliable method for computing integrals and sums of the type (21) can be found in [11][2]. A sensitivity analysis of the matrix exponential can be found in [38]. It is also shown in [20] that the controllable subspace and the unobservable subspace are the image and the kernel of $P_c(T)$ and $P_o(T)$, respectively. From this sensitivity properties of these spaces under perturbations of $P_c(T)$ and $P_o(T)$ are derived. A comparison of this approach and the one discussed under b2 is given in [12].

The algebraic Riccati equation in the continuous and discrete time cases :

$$\begin{aligned} Q + AP + PA' - PBR^{-1}B'P = 0 \\ Q + F'PF - P - F'PG(R + G'PG)^{-1}G'PF = 0 \end{aligned} \tag{22}$$

appears in several design/analysis problems such as optimal control, optimal filtering, spectral factorization etc. (see [14][22][40][60][61][63] and references therein). Several algorithms have been proposed via various approaches [3][14][40][60][62][63].

The (generalized) eigenvalue approach relates the solution of (22) to the computation of specific subspaces connected to the (generalized) eigenvalue problems :

$$\lambda \begin{bmatrix} I & 0 \\ 0 & I \end{bmatrix} - \begin{bmatrix} A & -BR^{-1}B' \\ -Q & -A' \end{bmatrix} \; ; \; \lambda \begin{bmatrix} I & GR^{-1}G' \\ 0 & F' \end{bmatrix} - \begin{bmatrix} F & 0 \\ -Q & I \end{bmatrix} \tag{23}$$

$$\lambda \begin{bmatrix} I & 0 & 0 \\ 0 & I & 0 \\ 0 & 0 & 0 \end{bmatrix} - \begin{bmatrix} A & 0 & B \\ -Q & -A' & 0 \\ 0 & B' & R \end{bmatrix} \; ; \; \lambda \begin{bmatrix} I & 0 & 0 \\ 0 & F' & 0 \\ 0 & G' & 0 \end{bmatrix} - \begin{bmatrix} F & 0 & -G \\ -Q & I & 0 \\ 0 & 0 & R \end{bmatrix}$$

for which stable software has been developed (see [40][60][61] and references therein). Stability and conditioning of these approaches are e.g. discussed in [40][60][39]. An analysis of other methods has been considered in [63][85][86].

c2. *Feedback control.*

Linear state feedback can be considered as an inverse eigenvalue problem [33] : given a pair (A, B), one looks for a matrix F such that the eigenvalues of the matrix $A+BF$ lie on specified locations or in specified regions. Several approaches have been proposed but the emphasis is shifting towards numerically reliable methods which at the same time might tell something about the sensitivity of the problem [64][87][88][89]. Special cases are e.g. deadbeat control (where $A + BF$ is supposed to be nilpotent) [22] and observer design [90].

c3. *Cascade factorization.*

In filter design the problem of cascade factorization plays an important role for purposes of sensitivity (see e.g. [66][69]). In [65] a general solution is given to this problem via state space techniques. The problem is shown to be equivalent to the computation of chains of invariant subspaces of two matrices, whose eigenvalues are the poles and zeros of the rational matrix to be factorized. The sensitivity of the approach can be found in [65][39]. Stable software for the problem is being developed [91].

c4. *Frequency response methods.*

The last few years there is a revived interest in frequency-response methods for design/analysis of linear systems (see e.g. [67] and the Special Issue [30]). Special attention is paid to the computational issues such as reliability, speed and sensitivity [30][68]. The singular value decomposition and its particular robustness properties are frequently used here (see [30] for recent developments). There is also a growing interest for reliable numerical methods in H_∞-control [92].

4 Alternative directions

In this concluding section we briefly go over a number of research topics in linear system theory where numerical linear algebra is infiltrating and numerical analysis in a broader sense if of great importance.

a. *Ordinary differential equations.*

The development of numerical methods for ODE's has been of great concern in circuits and system theory for purposes of computer aided design (see [6][70][29] and references therein). They are being used also for some analysis/design problems of control theory such as computing admissible initial conditions of descriptor systems, optimal control etc. [9][31]. A growing interest is observed [31].

b. *Sparse and structured matrices.*

Very often descriptions of linear systems are very sparse (e.g. [9][71]) or lead to solving structured matrix problems (e.g. Hankel, Toeplitz as seen earlier) and algorithms should be adapted to exploit this structure if at least possible without harming too much the stability of the algorithm. The numerical literature is quite rich in methods that are especially devised for dealing with such matrices [25][72] and some methods start to be applied in linear system theory [9][28][31].

c. *Fixed point arithmetic.*

In many engineering situations (e.g. digital signal processing) the user is forced to use fixed point arithmetic in order to meet some imposed restriction.
Some numerical algorithms have been tested on stability for fixed point arithmetic (see e.g. [8]) but the classical methods for fixed point error analysis [35] do not perform so well when very small bit ranges are considered (see [66] and references therein).

d. *Approximation.*

The approximation problem occurs on different places (such as filter design, speech processing); the area is quite extended in numerical analysis [74].

e. *Optimization.*

In computer aided design e.g. specific optimization problems occur which have not been treated yet in the numerical literature (see Special Issue [29] and [93]).

f. *Parallel algorithms.*

With the fast development of VLSI technology, the interest in parallel processing is also increasing. Fast parallel algorithms are being developed but the interest in reliable numerical software remains present (e.g. [28][86]).

g. *Software packages.*

Many numerical software packages [23][24][25][75] have been developed and are widely distributed. Special purpose packages for system theoretic applications are also being developed [3][76][77]. There is a growing need for unification and cooperation in that area [31][91].

References

[1] J.L. Melsa, Computer programs for computational assistance in the study of linear control theory, Mc Graw Hill, New York, 1970.

[2] E. S. Armstrong, ORACLS-A system for linear quadratic Gaussian control law design, NASA Techn. Paper 1106, NASA Langley Res. Center, Hampton, Virginia, 1978.

[3] G. J. Bierman, M. W. Nead, A parameter estimation subroutine package, JPL Publication 77-26, NASA Jet Propulsion Lab., Pasadena, California, 1978.

[4] R. M. Bartels, G. W. Stewart, Solution of the matrix equation AX+XB=C, Comm. ACM, Vol. 15, pp. 820-826, 1972.

[5] L. S. de Jong, Numerical aspects of the recursive realization algorithm, SIAM Contr. & Opt., Vol. 16, pp. 646-659, 1978.

[6] C. W. Gear, Simultaneous numerical solution of differential-algebraic euations, IEEE Trans. Circ. Th., Vol. CT-18, pp. 89-94, 1971.

[7] G. H. Golub, S. Nash, C. Van Loan, A Hessenberg-Schur method for the problem AX+XB=C, IEEE Trans. Aut. Contr., Vol. AC-24, pp. 909-913, 1979.

[8] G. Cybenko, The numerical stability of the Levinson-Durbin algorithm for Toeplitz systems of euations, SIAM Sci. & Stat. Comp., Vol. 1, 1980.

[9] R. F. Sincovec, A. Erisman, E. Yip, M. Epton, Analysis of descriptor systems using numerical algorithms, IEEE Trans. Aut. Contr., Vol. AC-26, pp. 139-147, 1981.

[10] A. Bultheel, Recursive algorithms for the matrix Padé problem, Math. of Comp., Vol.35, pp. 875-892, 1980.

[11] C. Van Loan, Computing Integrals involving the matrix exponential, IEEE Trans. Aut. Contr., Vol. AC-23, pp. 395-404, June 1978.

[12] P. Van Dooren, The generalized eigenstructure problem in linear system theory, IEEE Trans. Aut. Contr., Vol. AC-26, pp. 111-129, Feb. 1981.

[13] C. C. Paige, Properties of numerical algorithms related to computing controllability, IEEE Trans. Aut. Contr., Vol. AC-26, pp. 130-138, Feb. 1981.

[14] A. Barraud, Investigations autour de la fonction signe d'une matrice. Application à l'équation de Riccati, RAIRO. Automat. Syst. Anal. Contr., Vol. 13, pp. 335-368, 1979.

[15] M. Gutknecht, J. Smith, L. Trefethen, The Carathéodory-Fejer method for recursive digital filter design, IEEE Trans. Acoust. Speech & Signal Proc., Vol. ASSP-31, pp. 1417-1426, 1983.

[16] V. Klema, A. Laub, The singular value decomposition : its computation and some applications, IEEE Trans. Aut. Contr., Vol. AC-25, Apr. 1980.

[17] J. Wilkinson, Linear differential equations and Kronecker's canonical form, in Recent Advances in Numerical Analysis, pp. 231-265, C. de Boor, G. Golub Eds., New York, Academic, 1978.

[18] B. Moore, A. Laub, Computation of supremal (A,B)-invariant and controllability subspaces, IEEE Trans. Aut. Contr., Vol. AC-23, pp. 783-792, 1978.

[19] A. Laub, B. Moore, Calculation of transmission zeros using QZ techniques, Automatica, Vol. 14, pp. 557-566, 1978.

[20] B. Moore, Principal component analysis in linear systems, controllability, observability and model reduction, IEEE Trans. Aut. Contr., Vol. AC-26, pp. 17-31, Feb. 1981.

[21] S. Kung, A new identification and model reduction algorithm via singular value decompositions, Proc. 12-th Asilomar Conf. Circ., Syst. Comp., Nov. 1978.

[22] P. Van Dooren, Deadbeat control : a special inverse eigenvalue problem, BIT, Vol. 24, pp. 681-699, Dec. 1984.

[23] B. Garbow et al., Matrix eigensystem routines-EISPACK guide extension, New York : Springer, 1977.

[24] J. Dongarra et al., LINPACK user's guide, Philadelphia : SIAM, 1979.

[25] O. Arushanian et al., The TOEPLITZ Package : User's guide, Techn. Rept. Argonne Nat. Lab., 1979.

[26] C. Moler, G. Stewart, An algorithm for the generalized matrix eigenvalue problem, SIAM J. Num. Anal., Vol. 10, pp. 241-256, 1973.

[27] P. Van Dooren, The computation of Kronecker's canonical form of a singular pencil, Linear Algebra & Appl., Vol. 27, pp. 103-141, 1979.

[28] R. Brent, F. Gustavson, D. Yun, Fast solution of Toeplitz systems of euations and computation of Padè approximants, Journal of Algorithms, Vol. 1, pp. 259-295, 1980.

[29] IEEE Trans. Circ. & Syst., Special Issue on Computational Methods, Vol. CAS-26, Sept. 1979.

[30] IEEE Trans. Aut. Contr., Special Issue on Linear Multivariable Control Systems, Vol. AC-26, Feb. 1981.

[31] Workshop on Numerical Methods in Automatic Control, Lund Inst. Techn., Lund, Sweden, Sept. 1980.

[32] Workshop on Control Design, GE Res. Center & Rensselaer Pol. Inst., New York, May 1981.

[33] SIAM Conf. on Linear Algebra in Signals, Systems and Control, Boston, August 1986.

[34] J. Wilkinson, C. Reinsch, Handbook for automatic computation II : Linear algebra, New York : Springer, 1971.

[35] J. Wilkinson, The algebraic eigenvalue problem, Oxford : Clarendon, 1965.

[36] L. S. de Jong, Towards a formal definition of numerical stability, Numer. Math., Vol. 28, pp. 211-220, 1977.

[37] J. R. Rice, A theory of condition, SIAM Numer. Anal., Vol. 3, pp. 287-310, 1966.

[38] C. Van Loan, The sensitivity of the matrix exponential, SIAM Numer. Anal., Vol. 14, pp. 971-981, 1977.

[39] G. Stewart, Error and perturbation bounds for subspaces associated with certain eigenvalue problems, SIAM Rev., Vol. 15, pp. 727-764, 1973.

[40] P. Van Dooren, A generalized eigenvalue approach for solving Riccati equations, SIAM Sci. & Stat. Comp., Vol. 2, pp. 121-135, 1981.

[41] W. M. Wonham, Linear multivariable theory. A geometric approach, 2nd ed., New York : Springer, 1979.

[42] W. Kahan, Conserving confluence curbs ill-condition, Dept. Comp. Sci., Univ. California, Berkeley, Intern. Rep., 1972.

[43] A. Willsky, Digital signal processing and control and estimation theory. Points of tangency, areas of intersection and parallel direction, Cambridge : MIT Press, 1979.

[44] K. Aström, P. Eykhoff, System identification. A survey, Automatica, Vol. 7, pp. 123-162, 1971.

[45] H. Zeiger, A. Mc Ewen, Approximate linear realizations of given dimension via Ho's algorithm, IEEE Trans. Aut. Contr., Vol. AC-19, pp. 153, 1974.

[46] V. Adamjan, D. Arov, M. Krein, Analytic properties of Schmidt pairs for a Hankel operator and the generalized Schur-Takagi problem, Mat. USSR Sbornik, Vol. 15, pp. 31-73, 1971.

[47] Y. Genin, S. Kung, A two variable approach to the model reduction problem with Hankel criterion, IEEE Circ. & Syst., Vol. CAS-28, pp. 912-924, Sept. 1981.

[48] J. Makhoul, Linear prediction : a tutorial review, Proc. IEEE, Vol. 63, pp. 561-580, 1975.

[49] A. Bultheel, Error analysis of incoming and outgoing schemes for the trigonometric moment problem, in Lecture Note on Mathematics, Van Rossum & de Bruin Eds., Berlin Springer, 1980.

[50] B. D. O. Anderson, J. B. Moore, Optimal filtering, New Jersey : Prentice Hall, 1979.

[51] V. Kucera, Discrete linear control. The polynomial equation approach, Chichester : Wiley & Sons, 1979.

[52] J. Rissanen, Algorithm for triangular decomposition of block Hankel and Toeplitz matrices with application to factoring positive matrix polynomials, Math. Comp., Vol. 27, pp. 147-154. 1973.

[53] Z. Vostry, New algorithm for polynomial spectral factorization with quadratic convergence I. Kybernetika, Vol. 11, pp. 415-422, 1975.

[54] P. Henrici, Fast Fourier methods in computational complex analysis, SIAM Rev., Vol. 21, pp. 481-527, 1979.

[55] M. Jenkins, J. Traub, A three-stage algorithm for real polynomials using uadratic iteration, SIAM Numer. Anal., Vol. 7, pp. 545-566, 1970.

[56] P. Van Dooren, P. Dewilde, The eigenstructure of an arbitrary polynomial matrix: computational aspects, Linear Algebra & Applications, Vol. 50, pp. 545-579, 1983.

[57] L. Silverman, P. Van Dooren, A system theoretic approach for GCD extraction, IEEE Trans. Aut. Contr., Vol. AC-26, pp. 1273-1276, 1981.

[58] A. Laub, On computing balancing transformations, Proc. 1980 Jacc, pp. FA8-E, 1980.

[59] C. Mullis, R. Roberts, synthesis of minimum roundoff noise fixed point digital filters, IEEE Trans. Circ. & Syst., Vol. CAS-23, pp. 551-562, 1976.

[60] A. Laub, A Schur method for solving algebraic Riccati equations. IEEE Trans. Aut. Contr., Vol. AC-24, pp. 913-921, 1979.

[61] T. Pappas, A Laub, N. Sandell, On the numerical solution of the discrete time algebraic Riccati equation, IEEE Trans. Aut. Contr., Vol. AC-25, pp. 631-641, 1980.

[62] T. Kailath, a view of three decades of linear filtering theory, IEEE Trans. Inf. Th., Vol. IT-20, pp. 146-181, 1974.

[63] R. Byers, Hamiltonian and symplectic algorithms for the algebraic Riccati equation, Ph. D. Thesis, Cornell University, Ithaca, N.Y., 1983.

[64] G. Klein, B. Moore, Eigenvalue-generalized eigenvector assignment with state feeback, IEEE Aut. Contr., Vol. AC-22, pp. 140-141, 1977.

[65] P. Van Dooren, P. Dewilde, Minimal cascade factorization of real and complex rational transfer matrices, IEEE Trans. Circ. & Syst., Vol. CAS-28, pp. 390-400, 1981.

[66] L. Rabiner, B. Gold, Theory and application of digital signal processing, New Jersey : Prentice-Hall, 1975.

[67] H. Rosenbrock, Computer-aided control system design, London : Academic, 1974.

[68] A. Laub, Robust stability of linear systems : some computational considerations, in Information linkage between applied mathematics and industry II, Schoenstadt et al. Eds., New York : Academic, 1980.

[69] E. Deprettere, P. Dewilde, Orthogonal cascade realization of real multiport digital filters, Circ. Th. & Appl., Vol. 8, pp. 245-272, 1980.

[70] R. Brayton, F. Gustavson, G. Hachtel, A new efficient algorithm for solving differential alg. systems using implicit backward differentiation formulas, Proceedings IEEE, Vol. 60, pp. 98-108, 1972.

[71] G. Hachtel, R. Brayton, F. Gustavson, The sparse tableau approach to network analysis and design, IEEE Trans. Circ. Th., Vol. CT-18, pp. 101-113, 1971.

[72] D. Rose, R. Willoughby, Sparse matrices and their applications, New York : Plenum, 1973.

[73] A. Antoulas, New results on the algebraic theory of linear systems: the solution of the cover problems, Linear Algebra & Applications, Vol. 50, 1-45, 1983.

[74] E. Cheney, Introduction to approximation theory, New York, Mc Graw Hill, 1966.

[75] D. Coleman et al., A system of iteratively reweighted least squares computation, ACM TOMS, Vol. 6, pp. 327-336, 1980.

[76] L. Nagel, SPICE2 : a computer program to simulate semiconductor circuits, MEMO ERL-M520, Electr. Res. Lab., U.C. Berkeley, Californie, May 1975.

[77] SLICE, A Subroutine LIbrary for Control Engineering, Numerical Algorithms Group, Oxford, 1987.

[78] J. Bunch, Stability of methods for solving Toeplitz systems of equations, SIAM Sci. & Stat. Comp., Vol. 6, pp. 349-364, 1985.

[79] F. Itakura, S. Saito, Digital filtering techniques for speech analysis and synthesis, Conference Record, 7th Int. Congress on Acoustics, Paper 25, C 1, 1971.

[80] G. Cybenko, The numerical stability of the lattice algorithm for lesat squares linear prediction problems, BIT, Vol. 24, pp. 441-455, Dec. 1984.

[81] J. Jezek, V. Kucera, Efficient algorithm for spectral factorization, Proceedings 9th IFAC Congress, Budapest, July 1984.

[82] P. Van Dooren, Reducing subspaces : computational aspects and applications in linear system theory, Lecture Notes on Control and Information Sciences, Vol. 44, pp. 935-953, Springer, Dec. 1982.

[83] Th. Beelen, P. Van Dooren, An improved algorithm for the computation of Kronecker's canonical form of a singular pencil, Linear Algebra & Applications, 1988.

[84] J. Demmel, B. Kagström, Computing stable eigendecompositions of matrix pencils $A-\lambda B$, Linear Algebra & Applications, Vol.88/89, pp. 139-186, 1987.

[85] J. Gardiner, A. Laub, A generalization of the matrix-sign-function solution to the algebraic Riccati equations, Int. Journ. Control, Vol. 44, pp. 823-832, 1986.

[86] J.-P. Charlier, P. Van Dooren, A systolic algorithm for Riccati and Lyapunov equations, Mathematics of Control, Signals and Systems, Vol. 3, 1989.

[87] G. Miminis, C. Paige, An algorithm for pole assignment of time invariant multi input systems, Proc. 21st IEEE Conf. Dec. & Contr., pp. 62-67, 1982.

[88] A. Varga, A Schur method for pole assignment, IEEE Trans. Aut. Contr., Vol. AC-26, pp. 517-519, 1981.

[89] J. Kautsky, N. Nichols, P. Van Dooren, Robust pole assignment in linear state feedback, Int. Journ. Control, Vol. 41, pp. 1129-1156, 1985.

[90] P. Van Dooren, Reduced order observers : a new algorithm and proof, Systems & Contol Letters, Vol. 4, pp. 243-251, 1984.

[91] WGS Newsletter and Reports, Secr. : R. Kool, Techn. University of Eindhoven, Faculteit Wiskunde & Informatica, Postbus 513, NL-5600 MB Eindhoven, The Netherlands.

[92] H. Kwakernaak, Robust control, Journal A, Vol. 29, pp. 17-27, 1988.

[93] K. Machielsen, Numerical solution of state constrained optimal control problems, Journal A, Vol. 29, pp. 47-53, 1988.