Contemporary Mathematics
Volume 47, 1985

# ON THE USE OF UNITARY STATE-SPACE TRANSFORMATIONS

Paul M. Van Dooren
Michel Verhaegen[1]

ABSTRACT. Canonical forms of state-space models of multivariable systems have often been proposed for solving certain analysis and design problems encountered in linear system theory. In this paper we show that for many problems one can as well make use of so-called condensed forms, which can be obtained under unitary state-space transformations.

It is shown that the use of these forms has lead recently to elegant and numerically reliable methods for solving several problems formulated in terms of state-space models. It is then also stressed that these forms are likely to yield a promising approach for other problems as well as e.g. problems in generalized state-space models.

1. INTRODUCTION. Several basic problems encountered in areas such as control theory, network theory, stochastic systems, are (or can be) formulated in terms of (generalized) state-space models and are often tackled then by using techniques borrowed from numerical linear algebra [1][2][3][4][5][6][7][8][9][10][52][53]. In this area, unitary transformations have become a major tool in the development of reliable numerical methods, this for two reasons :

(i) because of the numerical sensitivity of the problem at hand. The sensitivity of several problems in linear algebra can indeed be expressed in terms of certain norms, singular values or angles and each of these are in general invariant under unitary transformations. Unitary transformations therefore allow one to reformulate the problem in a new coordinate system (usually more amenable for solving the problem) and this without affecting its sensitivity.

Typical examples of this are the QR decomposition [11] and the SVD decomposition [12] , both used for constructing least squares solutions to linear systems : they reduce the least squares problem min $\|Ax-b\|_2$ to a solvable but smaller linear system, without altering its sensitivity (the latter is e.g. not the case for the methods using the modified system $A^*Ax-A^*b=0$) [13] .

(ii) because of the numerical stability of the algorithm used for solving the problem. Most decompositions involving unitary transformations can be obtained by a sequence of Givens transformations or Householder reflections [14], each

[1] Supported by the IWONL, Belgium

of which can be performed in a numerically stable manner. The concatenation of such transformations can also be performed in a backward stable manner. Numerical errors resulting from previous steps are indeed maintained in norm throughout subsequent steps because these transformations (and their inverse) have 2-norm equal to 1. This, of course, should be checked carefully for each decomposition under consideration (see e.g. [14][15]).

Although unitary transformations are slower than e.g. elementary transformations (generally by a factor of 2 for a comparable decomposition), the above two points are the main reasons why they became so popular. This is especially true for iterative algorithms where the distortion of the sensitivity and loss of accuracy can become significant when using non-unitary transformations. Typical examples of this are the QR and QZ algorithms [16][17], which in general are preferred over the comparable LR and LZ algorithms [15] [18] for solving eigenvalue and generalized eigenvalue problems, respectively. In such iterative algorithms it is also important (as well for unitary as for non-unitary methods) to make use of so-called condensed forms in the implementation of each step of the iteration. The reduction to such a condensed form is usually done only once and is then maintained throughout the iteration, thereby resulting in a significant saving of computing time. The best example of this is the Hessenberg form used in eigenvalue problems, which is constructed in the initial step of the algorithm and is then maintained in all the subsequent steps of the algorithm [14][15][16][17][18].

In this paper we show how to use such condensed forms of systems of matrices {A,B,C,D} arising in state-space models, in order to speed up certain algorithms using these systems. It turns out that these forms not only allow to make savings in computing time for existing algorithms, but they also are of valuable support in deriving new algorithms. This is shown hereafter by surveying a number of recently derived algorithms, each of which uses condensed forms in one way or another.

In the next section we give different types of condensed forms that have been proposed for systems of matrices {A,B,C,D} representing state-space models. The following sections are then devoted to different applications of these forms in specific analysis and design problems using state-space models. We then end with a section showing how to extend some of these ideas to some open problem including generalized state-space models.

## 2. CONDENSED STATE-SPACE MODELS.

As suggested in the previous section we restrict ourselves here to the use of unitary transformations for reasons of sensitivity and numerical stability. Similar results, though, also hold for elementary transformations in most of the applications mentioned in the sequel.

By condensed forms we thus mean here forms that contain as many zeros as possible and were obtained under unitary transformations. Since our system of matrices $\{A,B,C,D\}$ corresponds to a state space model (with $\lambda$ standing for the differential operator in the continuous time case and for the shift operator in the discrete time case) :

$$\begin{cases} \lambda x = Ax + Bu \\ y = Cx + Du \end{cases} \tag{1}$$

we restrict ourselves to unitary state-space transformations U (of the n-vector x), input transformations V (of the m-vector u) and output transformations W (of the p-vector y). The transformed system (1), using :

$$x_t = Ux \ ; \ u_t = Vu \ ; \ y_t = Wy \tag{2}$$

thus has the corresponding quadruple $\{A_t, B_t, C_t, D\}$ , where

$$A_t = UAU^* \ ; \ B_t = UBV^* \ ; \ C_t = WCU^* \tag{3}$$

with U,V and W unitary matrices. In practice, n is considerably larger than m and p and most of the zeros will therefore be obtained in $A_t$ by appropriately choosing U. The four main condensed forms encountered in the literature are :

(i) the <u>Hessenberg form</u>, where $A_t$ is upper Hessenberg.

(ii) the <u>Schur form</u>, where $A_t$ is in upper Schur form.

(iii) the <u>controller-Hessenberg form</u>, where the compound matrix $\left[B_t \,\middle|\, A_t\right]$ is upper trapezoidal

(iv) the <u>observer-Hessenberg form</u>, where the compound matrix $\left[\begin{array}{c} A_t \\ \hline C_t \end{array}\right]$ is upper trapezoidal.

These forms are illustrated below for m=3, n=7, p=2 :

$$\left[\begin{array}{c|c} B_h & A_h \\ \hline D & C_h \end{array}\right] = \left[\begin{array}{ccc|ccccccc} x & x & x & x & x & x & x & x & x & x \\ x & x & x & x & x & x & x & x & x & x \\ x & x & x & 0 & x & x & x & x & x & x \\ x & x & x & 0 & 0 & x & x & x & x & x \\ x & x & x & 0 & 0 & 0 & x & x & x & x \\ x & x & x & 0 & 0 & 0 & 0 & x & x & x \\ x & x & x & 0 & 0 & 0 & 0 & 0 & x & x \\ \hline x & x & x & x & x & x & x & x & x & x \\ x & x & x & x & x & x & x & x & x & x \end{array}\right] \tag{4}$$

$$\left[\begin{array}{c|c} B_s & A_s \\ \hline D & C_s \end{array}\right] = \left[\begin{array}{ccc|ccccccc} x&x&x&x&x&x&x&x&x&x \\ x&x&x&0&x&x&x&x&x&x \\ x&x&x&0&0&x&x&x&x&x \\ x&x&x&0&0&0&x&x&x&x \\ x&x&x&0&0&0&0&x&x&x \\ x&x&x&0&0&0&0&0&x&x \\ x&x&x&0&0&0&0&0&0&x \\ \hline x&x&x&x&x&x&x&x&x&x \\ x&x&x&x&x&x&x&x&x&x \end{array}\right] \tag{5}$$

$$\left[\begin{array}{c|c} B_c & A_c \\ \hline D & C_c \end{array}\right] = \left[\begin{array}{ccc|ccccccc} x&x&x&x&x&x&x&x&x&x \\ 0&x&x&x&x&x&x&x&x&x \\ 0&0&x&x&x&x&x&x&x&x \\ 0&0&0&x&x&x&x&x&x&x \\ 0&0&0&0&x&x&x&x&x&x \\ 0&0&0&0&0&x&x&x&x&x \\ 0&0&0&0&0&0&x&x&x&x \\ \hline x&x&x&x&x&x&x&x&x&x \\ x&x&x&x&x&x&x&x&x&x \end{array}\right] \tag{6}$$

$$\left[\begin{array}{c|c} B_o & A_o \\ \hline D & C_o \end{array}\right] = \left[\begin{array}{ccc|ccccccc} x&x&x&x&x&x&x&x&x&x \\ x&x&x&x&x&x&x&x&x&x \\ x&x&x&x&x&x&x&x&x&x \\ x&x&x&0&x&x&x&x&x&x \\ x&x&x&0&0&x&x&x&x&x \\ x&x&x&0&0&0&x&x&x&x \\ x&x&x&0&0&0&0&x&x&x \\ \hline x&x&x&0&0&0&0&0&x&x \\ x&x&x&0&0&0&0&0&0&x \end{array}\right] \tag{7}$$

where the x's denote arbitrary elements. Notice that the first form has (n-2)(n-1)/2 zeros, while the other three forms have as many zeros as each other, namely n(n-1)/2. Some slightly modified forms exist as well. In order to obtain the Schur form (5), one needs in general complex transformation matrices even when A is real. A so-called real Schur form for real A matrices will indeed have 2×2 bumps on the diagonal corresponding to the pairs of complex eigenvalues [16]. Its number of zeros is thus between the forms (4) and (5). Variants of the forms (6) and (7) are the so-called staircase forms [19][20][7][21][22][23][10][24] . These variants all require rank tests, which is reflected by the fact that some of the x elements are known to be non zero. These elements are e.g. denoted by X's in the form due to Minimis & Paige [10]:

$$\left[\begin{array}{c|c} B_o & A_o \\ \hline D & C_o \end{array}\right] = \left[\begin{array}{ccc|ccccccc} 0 & X & x & x & x & x & x & x & x & x \\ 0 & 0 & X & x & x & x & x & x & x & x \\ 0 & 0 & 0 & X & x & x & x & x & x & x \\ 0 & 0 & 0 & 0 & X & x & x & x & x & x \\ 0 & 0 & 0 & 0 & 0 & 0 & X & x & x & x \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & X & x & x \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & X & x \\ \hline x & x & x & x & x & x & x & x & x & x \\ x & x & x & x & x & x & x & x & x & x \end{array}\right] \qquad (8)$$

Here we always have more than $n(n-1)/2$ and less than $n(n+2m-1)/2$ zeros, and less than $n(n+p-1)/2$ for the equivalent staircase form of (7). Finally, one can construct lower condensed forms corresponding to all the above upper forms, by merely working on the dual quadruple $\{A^*, C^*, B^*, D^*\}$, where $M^*$ denotes the (conjugate) transpose of a (complex) matrix M. Notice that only the staircase forms require more than a state-space transformation U : an input transformation V is needed to obtain the form (8) and an output transformation W is needed for the staircase form corresponding to (7).

The algorithms yielding these different forms all have a comparable complexity : the number of flops required for the forms (4)(5)(6)(7)and (8) roughly are $n^2(3n+m+p)$, $n^2(5kn+p+n)$, $n^2(3n+m+p)$, $n^2(3n+m+p)$ and $n^2(3n+6m+p)$, respectively. Here, k is the average number of QR steps used in the algorithm yielding the Schur form (5), and is usually between 1 and 2 [14]. These operation counts include the construction of the corresponding state-space transformation U (and, if needed, V or W) yielding the corresponding condensed form.

## 3. EXPLOITING CONDENSED FORMS.

The forms presented above gain specific interest when using them as part of another algorithm. Their zeros are efficiently exploited in the following situations.

(i) updating Schur forms. Finding a (real) Schur form of a full n×n matrix takes $O(n^3)$ operations. Yet, when starting from one Schur form, it takes only $O(kn)$ operations to obtain another one with a different order of eigenvalues on diagonal [13][26] , where k is the number of permutations of two adjacent diagonal elements needed to go from one form to the other.

(ii) performing an implicit QR step. This is normally part of the QR algorithm yielding the Schur form in an iterative manner. Yet, when the eigenvalues are known, it can be used as an efficient tool to obtain a Schur form with prescribed ordering of eigenvalues on the diagonal [30] . Its complexity is $O(n^2)$ for a single QR step, this for moving any given eigenvalue (or pair of complex conjugate eigenvalues) to the bottom corner of a Schur form.

(iii) further elimination to obtain a QR decomposition. The amount of computations needed to obtain a QR decomposition of an $m \times n$ matrix with only k subdiagonals is roughly $kn^2$ (when $m > n$). This decomposition, used for inverting square matrices or solving least squares problems involving arbitrary matrices, may thus be performed economically for such matrices with only few subdiagonals.

These three basic tools of numerical linear algebra now appear to be useful in several problems occurring in linear system theory and happen to be economical when the given system quadruple $\{A,B,C,D\}$ is in one of the condensed forms given above. A few applications are given in the next sections.

4. LINEAR STATE FEEDBACK. Several methods have been proposed for solving the problem of pole placement via linear state feedback, i.e. specifying the spectrum $\{\lambda_1,\ldots, \lambda_n\}$ of $(A+BF)$ where A and B are two matrices of a given state-space model and F is to be chosen. We here describe a few methods which make use of condensed forms.

A first method, due to Varga [9] , uses the system $\{A_s,B_s,C_s,D\}$ in Schur form as in (5). By performing a feedback $F_1$ with only non zero elements in the last column, one obtains a feedback system $(A_s+B_sF_1)$ which is still in (upper triangular) Schur form and whose bottom eigenvalue can be assigned arbitrarily (e.g. to $\lambda_1$) by an appropriate choice of the non zero elements in $F_1$. One of course can not use the other columns of F to place to another eigenvalues without destroying the triangular form of $(A_s+B_sF)$. But, instead, one can use a reordering technique as described in 3.(i) to move the assigned eigenvalue $\lambda_1$ to the (1,1) position of the Schur form by a unitary state-space transformation $U_1$. In this new coordinate system one can now use a second feedback $F_2$, again with only non zero elements in the last column, to assign the bottom eigenvalue to $\lambda_2$. This assigned eigenvalue is then moved in turn to the (2,2) position of the Schur form by an updating unitary transformation $U_2$. This process is repeated untill all n eigenvalues $\lambda_i$ are assigned. This method works as well for single input as multi-input systems and its complexity is $O(n^3)$.

Another method, due to Miminis & Paige [10] , starts from the staircase form (8). For a single input system, $A_c$ is then in Hessenberg form and $B_c$ has only one non zero element. It is shown in [10] that, using an implicit QR-step on $A_c$ with shift $\lambda_1$ one then comes to the configuration :

$$[U_1B_c \mid U_1(A_c+B_cF_1)U_1^*] = \left[\begin{array}{c|c|cccccc} x & \lambda_1 & x & x & x & x & x & x \\ \hdashline X & 0 & x & x & x & x & x & x \\ 0 & 0 & X & x & x & x & x & x \\ 0 & 0 & 0 & X & x & x & x & x \\ 0 & 0 & 0 & 0 & X & x & x & x \\ 0 & 0 & 0 & 0 & 0 & X & x & x \\ 0 & 0 & 0 & 0 & 0 & 0 & X & x \end{array}\right] \qquad (9)$$

when appropriately choosing $F_1$. This now displays the eigenvalue $\lambda_1$ which is already placed and a complementary subsystem which is again in staircase form, but now of dimension reduced by 1. The same idea is then recursively applied until all the eigenvalues $\lambda_i$ are assigned. The complexity is here again $O(n^3)$ since each QR step is of $O(n^2)$ as explained in 3.(ii). The extension of these ideas to multi input systems is described in [10].

A third method, due to Kautsky, Nichols & Van Dooren [27], is specifically designed for multi input systems, and uses the degrees of freedom that are still available in the choice of F, to minimize the sensitivity of the assigned eigenvalues $\lambda_i$. In a sense it thus tries to find a solution that is as robust as possible. This then becomes an optimization problem which would become very expensive in computing time if one would not make extensive use of condensed forms. The eigenvector $x_i$ corresponding to each assigned eigenvalue $\lambda_i$ must indeed belong to the null space $S_i$ of the matrices $\left[A_{21} \mid A_{22}-\lambda_i I\right]$, where $A_{21}$ and $A_{22}$ are submatrices of the staircase form (8) :

$$\left[\begin{array}{c|c|c} B_1 & A_{11} & A_{12} \\ \hline 0 & A_{21} & A_{22} \end{array}\right] = \left[\begin{array}{ccc|cc|ccccc} 0 & X & x & x & x & x & x & x & x & x \\ 0 & 0 & X & x & x & x & x & x & x & x \\ \hline 0 & 0 & 0 & X & x & x & x & x & x & x \\ 0 & 0 & 0 & 0 & X & x & x & x & x & x \\ 0 & 0 & 0 & 0 & 0 & 0 & X & x & x & x \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & X & x & x \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & X & x \end{array}\right] \qquad (10)$$

These null spaces $S_i$ are obtained by computing the RQ decomposition of the matrices $\left[A_{21} \mid A_{22}-\lambda_i I\right]$ which can be done cheaply because these matrices are already nearly upper triangular. After that, the core of the iterative algorithm consists of updating the choices of the eigenvectors $x_i$ from these spaces $S_i$ in order to minimize a certain sensitivity function. Here again updating QR decompositions of nearly triangular matrices, makes it possible to obtain an algorithm yielding a satisfactory solution within $O(n^3 m)$ operations.

Finally, a fourth method, due to Van Dooren [28], solves the deadbeat control problem for multi input systems. Here the minimum norm feedback matrix F is computed that assigns all the eigenvalues $\lambda_i$ at zero and, moreover, minimizes the lengths of the corresponding Jordan chains. The algorithm makes extensive use of the staircase form (8) of the system, in order to build recursively the minimum norm solution F to this problem. Due to the use of this condensed form, the complexity of this algorithm is $O(n^3)$.

5. KALMAN FILTERING. In this problem [29] one considers the stochastic signal $\{y(k)\}$ generated by the (time invariant) system :

$$\begin{cases} x(k+1) = A.x(k) + B.u(k) \\ y(k) \quad = C.x(k) + v(k) \end{cases} \tag{11}$$

where u(k) and v(k) are independent white noise sequences with (fixed) covariances $C_{uu}$ and $C_{vv}$, respectively. One of the most suited methods to solve this problem is the so-called covariance square root filter, whereby one has to perform, recursively, QR factorizations of the compound matrices :

$$Q \begin{bmatrix} C_u & 0 \\ S_k.C^* & S_k.A^* \\ 0 & C_v.B^* \end{bmatrix} = \begin{bmatrix} * & * \\ 0 & S_{k+1} \\ 0 & 0 \end{bmatrix} \tag{12}$$

Here, $C_u$, $C_v$ and $S_k$ are upper triangular Choleski factors of the positive definite matrices :

$$C_{uu}=C_u^*.C_u \ ; \ C_{vv} = C_v^*C_v \ ; \ P_k=S_k^*S_k \tag{13}$$

whereby $P_k$ is the covariance of the optimal state estimator at time k. Given $P_0$, the above recursion thus allows to compute all these covariances and from this the Kalman filter is then easily constructed. When now choosing the dual state-space system $\{A^*,C^*,B^*,D^*\}$ in controller Hessenberg form (6), a substantial part of the matrix to be triangularized is already zero [30]. For m=3, n=7, p=2 the decomposition (12) indeed looks like :

$$Q. \left[\begin{array}{cc|ccccccc} x & x & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & x & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \hline x & x & x & x & x & x & x & x & x \\ 0 & x & x & x & x & x & x & x & x \\ 0 & 0 & x & x & x & x & x & x & x \\ 0 & 0 & 0 & x & x & x & x & x & x \\ 0 & 0 & 0 & 0 & x & x & x & x & x \\ 0 & 0 & 0 & 0 & 0 & x & x & x & x \\ 0 & 0 & 0 & 0 & 0 & 0 & x & x & x \\ \hline 0 & 0 & x & x & x & x & x & x & x \\ 0 & 0 & x & x & x & x & x & x & x \\ 0 & 0 & x & x & x & x & x & x & x \end{array}\right] = \left[\begin{array}{cc|ccccccc} x & x & x & x & x & x & x & x & x \\ 0 & x & x & x & x & x & x & x & x \\ \hline 0 & 0 & x & x & x & x & x & x & x \\ 0 & 0 & 0 & x & x & x & x & x & x \\ 0 & 0 & 0 & 0 & x & x & x & x & x \\ 0 & 0 & 0 & 0 & 0 & x & x & x & x \\ 0 & 0 & 0 & 0 & 0 & 0 & x & x & x \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & x & x \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & x \\ \hline 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{array}\right] \tag{14}$$

The number of elements that have to be zeroed is now clearly O(np+nm), instead of $O(n^2)$ when not using condensed forms. Since this decomposition is the main part of the Kalman filter recursion, considerable savings can be obtained when

p and m are much smaller than n (which is usually the case). Similar savings are also obtained when using the Schur form in both the square root covariance filter and square root information filter [30] .
We also note here that the optimal control problem over a finite time interval k is dual to the above problem [2] and can thus also be solved in a similar fashion.

6. LYAPUNOV AND RICCATI EQUATIONS. For the solution of the matrix Lyapunov equations :

$$A^*P + PA = -C^*C \tag{15a}$$

$$P - A^*PA = C^*C \tag{15b}$$

efficient $O(n^3)$ methods have been described in [31][32][33]. They all use the Schur form of the A matrix in order to solve for P [31][32] or its upper triangular Choleski factor [33] in a fast recursive manner. Solving the linear systems of equations (15) using the sparse Kronecker product notation would indeed yield a slower method, even when using sparse matrix techniques. The solution of these equations can e.g. be used for stabilization via feedback or also in a recursive method for solving the Riccati equation [34]:

$$C^*C + A^*P + PA - PGG^*P = 0 \tag{16a}$$

with

$$GG^* = BR^{-1}B^* \tag{16b}$$

In the latter case, it is more interesting to use the form (6) since one has to solve recursively equations of the type [34]:

$$A_k^* . P_{k+1} + P_{k+1} . A_k = -[C^*C + P_k.G.G^*.P_k] \tag{17a}$$

where

$$A_k = A - G.G^*.P_k \tag{17b}$$

If $[B \mid A]$, and therefore also $[G \mid A]$, is in upper trapezoidal form as in (6), then all $A_k$ have the same pattern of zeros as A. When m=1, this then yields automatically $A_k$ in Hessenberg form. This form, needed in the Schur reduction of $A_k$, then does not have to be computed anymore. A reliable method of solving the above equation (16), is also to compute the stable subspace (i.e. the invariant subspace corresponding to the eigenvalues with negative real part) of the Hamiltonian matrix :

$$H = \begin{bmatrix} A & -BR^{-1}B^* \\ -C^*C & -A^* \end{bmatrix} \tag{18}$$

which can be done with the QR algorithm [6][26]. The preliminary reduction to Hessenberg form which is thereby needed, is obtained at no cost for a system $\{A,B,C,D\}$ in the form (7) when p=1, since it only requires a permutation of rows and columns of $A^*$. This idea can also be retrieved in the work of Byers [35], who uses this form as a starting point for a QR algorithm which also preserves the Hamiltonian structure (18) throughout the iterative process [36]. Such eigenvalue methods as [6][8][35] are in general to be preferred over other iterative methods such as the Kalman recursion given in (12) or the Newton method given in (17) because of the quadratic convergence of the QR-type algorithms.

7. OBSERVERS. For the construction of a Luenberger observer [37] for a plant (1), one has to solve the Sylvester type equation :

$$TA - FT = DC \tag{19}$$

where A and C are known and T,F and D are to be chosen. For a reduced order observer, the restrictions on the solution are that F should be $(n-p)\times(n-p)$ and stable, and that the compound matrix $\left[\frac{T}{C}\right]$ should be invertible [38] . An early method proposed to solve this problem [37], is to choose F and C appropriately and then to solve for T in (19). For this, one can use (as recommended in [39]) a method where both A and F are put in condensed forms via unitary similarity transformations (this amounts to state-space transformations in the plant and in the observer) :

$$T_{vu}.A_u - F_v.T_{vu} = D_v.C_u \tag{20a}$$

with

$$T_{vu} = V.T.U^* \;;\; A_u = UAU^* \;; \qquad F_v = V.F.V^* \;;\; D_v = V.D \;;\; C_u = C.U \tag{20b}$$

Here $A_u$ is chosen to be upper Hessenberg and $F_v$ lower Schur. This is illustrated below for n=6, p=2 ;

$$T_{vu}.\begin{bmatrix} x&x&x&x&x&x\\ x&x&x&x&x&x\\ 0&x&x&x&x&x\\ 0&0&x&x&x&x\\ 0&0&0&x&x&x\\ 0&0&0&0&x&x \end{bmatrix} - \begin{bmatrix} \lambda_1&0&0&0\\ x&\lambda_2&0&0\\ x&x&\lambda_3&0\\ x&x&x&\lambda_4 \end{bmatrix}.T_{vu} = \begin{bmatrix} x&x&x&x&x&x\\ x&x&x&x&x&x\\ x&x&x&x&x&x\\ x&x&x&x&x&x \end{bmatrix} \tag{21}$$

From this form one now easily solves for $T_{vu}$ column by column (starting from the top one), provided that the eigenvalues $\lambda_i$ of F are disjoint from those of A [39] . Another approach, inspired from the above one, is to use the observer-Hessenberg form for the plant (i.e. for A,C) and still the lower Schur form for the observer (i.e. for F). For n=6, p=2, this e.g. gives :

$$T_{vu}\cdot\begin{bmatrix} x & x & x & x & x & x \\ x & x & x & x & x & x \\ x & x & x & x & x & x \\ 0 & x & x & x & x & x \\ 0 & 0 & x & x & x & x \\ 0 & 0 & 0 & x & x & x \end{bmatrix} - \begin{bmatrix} \lambda_1 & 0 & 0 & 0 \\ x & \lambda_2 & 0 & 0 \\ x & x & \lambda_3 & 0 \\ x & x & x & \lambda_4 \end{bmatrix} . T_{vu} = D_v \cdot \begin{bmatrix} 0 & 0 & 0 & 0 & x & x \\ 0 & 0 & 0 & 0 & 0 & x \end{bmatrix} \tag{22}$$

where now the right hand side is written in factored form. In [38] it is shown that this form can now be used to solve for the columns of $T_{vu}$, $F_v$ and $D_v$ simultaneously, provided the eigenvalues $\lambda_i$ of F where chosen first. The advantage of such an approach is that more degrees of freedom are left in the equation (22) which then allows us to look for a minimum norm solution, having better robustness properties than other solutions [38]. As in the previous methods, one takes full advantage of the condensed forms to wind up with an $O(n^3)$ algorithm for solving this equation.

8. POLES, ZEROS AND TRANSFER FUNCTIONS. For the computation of poles and zeros of a system (1), one is recommended to use eigenvalue and generalized eigenvalue techniques [5][7][40]. The use of the condensed forms (6) and (7) is then particularly indicated when m=p=1. In both cases A is then in Hessenberg form, which is the first step towards the Schur decomposition and thus the poles of the transfer function. For the zeros of the transfer function $h(\lambda)$, one considers the pencil [40]:

$$\left[\begin{array}{c|c} A - \lambda I_n & B \\ \hline C & D \end{array}\right] = T - \lambda S. \tag{23}$$

For n=6, m=p=1, and using the condensed form (7), this looks like :

$$T-\lambda S = \left[\begin{array}{cccccc|c} x & x & x & x & x & x & x \\ x & x & x & x & x & x & x \\ 0 & x & x & x & x & x & x \\ 0 & 0 & x & x & x & x & x \\ 0 & 0 & 0 & x & x & x & x \\ 0 & 0 & 0 & 0 & x & x & x \\ \hline 0 & 0 & 0 & 0 & 0 & x & x \end{array}\right] - \lambda \left[\begin{array}{cccccc|c} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ \hline 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{array}\right] \tag{24}$$

This pencil is now in generalized Hessenberg form, which is the first step towards the generalized Schur decomposition of (24) and thus the computation of zeros of the transfer function. Moreover, as is shown in [41][42], the form (24) is such that one readily extracts the output decoupling zeros and zeros at infinity by merely checking whether some of the elements of T are zero. A similar result also holds for input decoupling zeros when using the condensed form (6) instead of (7). This then leads to smaller dimensional (generalized) eigenvalue problems for the poles (zeros) of $h(\lambda)$ in terms of a matrix (pencil) that is still in (generalized) Hessenberg form [41]. Together with the gain g, the poles $\{\beta_i\}$ and zeros $\{\alpha_j\}$ are sufficient for reconstructing the transfer function :

$$h(\lambda) = n(\lambda)/d(\lambda) = g \prod_j(\lambda-\alpha_j)/\prod_i(\lambda-\beta_i) \tag{25}$$

and the above approach is shown to be one of the most reliable tools to compute this information [40][41][42] . This can also be used for computing the entries

$$h_{ij}(\lambda) = n_{ij}(\lambda)/d_{ij}(\lambda) \tag{26}$$

of the transfer matrix $H(\lambda)$ of a state-space system $\{A,B,C,D\}$ with several inputs and outputs. Realizations for the single entries $h_{ij}(\lambda)$ are indeed given by $\{A,B_i,C_j,D_{ij}\}$ , where $B_i$ is the i-th column of B, $C_j$ the j-th row of C and $D_{ij}$ the (i,j) element of D. These systems are of course not necessarily minimal but, as discussed above, all the information needed to reconstruct the polynomials $n_{ij}(\lambda)$ and $d_{ij}(\lambda)$ is easily obtained from the controller - or observer-Hessenberg form of these single input-single output systems. Deriving these p×m condensed forms can now be done at relatively low cost since the updating state-space transformation $U_j$ for reducing e.g. $\{A,B_i,C_j,D_{ij}\}$ to observer-Hessenberg form is the same for i=1,...,m. All p×m condensed forms are thus obtained in $O(pn^3)$ operations.

9. FREQUENCY RESPONSE AND SYSTEM RESPONSE. In [43] it is shown that using the Hessenberg form (4), considerably reduces the amount of work needed to compute the frequency responses :

$$H_c(j\,\omega_k) = C(j\,\omega_k-A)^{-1}\,B+D \tag{27a}$$

$$H_d(e^{j\omega_k}) = C(e^{j\omega_k}-A)^{-1}\,B+D \tag{27b}$$

of a continuous-time and discrete-time system, respectively, when this is needed for several values k=1,...,N. The inverses of the shifted A matrices in (27) normally require indeed $O(n^3)$ operations. When, on the other hand, A is Hessenberg, so are the shifted A matrices and each inverse only requires $O(n^2)$ operations. Let indeed $Q_k.R_k$ be the QR factorization of these Hessenberg matrices (requiring $2n^2$ operations) then

$$H_k = (C\,R_k^{-1})(Q_k^*B) + D\ , \tag{28}$$

which takes $pn^2/2$ operations for computing $CR_k^{-1}$, 4mn for $Q_k^*B$ ( since $Q_k$ is a sequence of n Givens rotations) and pmn for finally obtaining $H_k$. When p is small, the QR factorization thus takes most of the work, which can e.g. be avoided by using the Schur form (5) instead. In [43] , how to deal more efficiently with real matrices {A,B,C,D} is also addressed. If N is very high (say more than 1000), this might still be too expensive and using the polynomials $n_{ij}(\lambda)$ and $d_{ij}(\lambda)$, as defined in section 8, will yield a O(n) method (which can be slightly less reliable, though).

For the computation of system responses (i.e. step response, impulse response, transient response, etc.) one has to evaluate expressions of the type :

$$y_k = \sum_{i=1}^{k} C.A^{k-i}.B.u(i-1)+C.A^k.X_0 \tag{29a}$$

$$y(T) = \int_0^T C.e^{A(T-t)}.B.u(t).dt+C.e^{AT}.X(0) \tag{29b}$$

For (29a) one has to build up the consecutive powers of A, which is cheaper for A in Schur form and, to a lesser extent, for A in Hessenberg form. For computing the exponential of a matrix the method of Pade approximation is particularly indicated [4][44]. Its implementation also requires the accumulation of products of matrices which can have up to half of their elements equal to zero when a condensed form is chosen for {A,B,C,D}. For the convolution of (29b) a numerical integration method for ODE's can be used, which again is more economical when e.g. the condensed form (6) is chosen [45]. Finally, the same holds also for the computation of integrals involving exponentials [4] , which arise e.g. in the construction of Grammians. Other papers recommending the use of Schur and Hessenberg forms can also be found in the ODE literature [46][47].

## 10. POSSIBLE EXTENSIONS.

It has been shown in the previous sections that condensed forms are a useful tool in a lot of problems formulated in state-space. These forms either lead to novel algorithms for certain problems (see sections 4 and 7), or allow one to speed up existing algorithms (see sections 5,6,8 and 9). The latter is especially experienced in algorithms of an iterative type, i.e. where certain matrix operations have to be performed many times. The prototype example in linear algebra is of course the QR algorithm working on Hessenberg matrices. In this paper typical examples are given by the problem of Kalman filtering and frequency response evaluation. Similar improvements may also be expected in e.g. optimization problems since they often need iterative methods to solve them. One example of this here is the robust pole placement problem described in section 4.

Another area where extensions can be expected is that of generalized state space models :

$$\begin{cases} \lambda Ex = Ax + Bu \\ y = Cx + Du \end{cases} \qquad (30)$$

involving a quintuple of matrices $\{A,B,C,D,E\}$ . Under the allowable transformations :

$$A_t = TAU^* \ ; \ B_t = TBV^* \ ; \ C_t = WCU^* \ ; \ E_t = TAU^* \qquad (31)$$

one obtains condensed forms similar to those of section 2 :

(i) the <u>generalized Hessenberg form</u>, where $A_t$ is upper Hessenberg and $E_t$ upper triangular.

(ii) the <u>generalized Schur form</u>, where $A_t$ is quasi upper triangular and $E_t$ is upper triangular.

(iii) the <u>controller-Hessenberg form</u>, where $\left[B_t \mid A_t\right]$ is upper trapezoidal and $E_t$ upper triangular.

(iv) the <u>observer-Hessenberg form</u>, where $\left[\begin{array}{c} A_t \\ \hline C_t \end{array}\right]$ is upper trapezoidal and $E_t$ upper triangular.

These forms are thus identical to the above ones except for $E_t$ which is upper triangular in each case, and was the identity matrix before. Variants of these forms and number of operations to obtain them are comparable to what was stated for state space models. Problems where these forms have been used or should be usable are : pole placement [10],[27] , optimal filtering [48], Sylvester type equations [39][47] , pole and zero evaluation [7], optimal control [49] [50],deadbeat control [51] and so on.

11. CONCLUSION. The survey of numerical methods given in this paper has put the emphasis heavily on the use of unitary transformations because of their great importance in preserving the sensitivity of the considered problems and in allowing for the derivation of numerically stable algorithms. It is shown that so-called condensed forms constitute an intermediate step in many of the reliable methods presently available in the area of linear system theory. It is also stressed that the use of these forms can be extended to other problems and will probably lead also there to good algorithms,again because of the use of unitary transformations.

## BIBLIOGRAPHY

1. G. Bierman, "Sequential square root filtering and smoothing of discrete linear systems", Automatica, Vol. 10, pp. 147-158, 1974.

2. L. Silverman, "Discrete Riccati equations : alternative algorithms, asymptotic properties and system theoretic interpretations", Control and Dynamic Systems, New York : Academic, 1976.

3. C. Paige, M. Saunders, "Least squares estimation of discrete linear dynamic systems using orthogonal transformations", SIAM J. Num. Anal., Vol. 14, pp. 180-193, 1977.

4. C. Van Loan, "Computing integrals involving the matrix exponential", IEEE Trans. Aut. Control, Vol. AC-23, pp. 395-404, 1978.

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

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

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

8. P. Van Dooren, "A generalized eigenvalue approach for solving Riccati equations", SIAM J. Scisc.,Vol. 2, pp. 121-135, 1981.

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

10. G. Miminis, C. Paige, "An algorithm for pole assignment of time invariant multi input linear system", Proceedings 21st IEEE Conf. Dec. & Contr., pp. 62-67, 1982.

11. G. Golub, "Numerical methods for solving linear least squares problems", Numer. Math., Vol. 7, pp. 206-216, 1965.

12. G. Golub, W. Kahan, "Calculating the singular values and pseudo-inverse of a matrix", SIAM J. Num. Anal., Vol. 2, pp. 205-224, 1965.

13. G. Golub, J. Wilkinson, "Ill-conditioned eigensystems and the computation of the Jordan canonical form", SIAM Rev., Vol. 15, pp. 727-764, 1973.

14. J. Wilkinson, C. Reinsch, "Handbook for automatic computation. II : Linear algebra", New York : Springer, 1971.

15. J. Wilkinson, "The algebraic eigenvalue problem", Oxford : Clarendon, 1965.

16. G. Francis, "The QR transformation. A unitary analogue to the LR transformation - I & II", Computer J., Vol. 4, pp. 265-271 & 332-345, 1961 & 1962.

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

18. L. Kaufman, "The LZ-algorithm to solve the generalized eigenvalue problem", SIAM J. Num. Anal., Vol. 11, pp. 997-1024, 1974.

19. P. Van Dooren, A. Emami-Naeini, L. Silverman, "Stable extraction of the Kronecker structure of pencils", Proceedings 17th IEEE Conf. Dec. & Contr., pp. 521-524, 1979.

20. D. Boley, "Computing the controllability-observability decomposition of linear time-invariant systems. A numerical approach", Ph. D. Thesis, Dept. Comp. Sci., Stanford Univ., 1981.

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

22. A. Varga, "A numerically stable algorithm for standard controllability form determination", Electronic Letters, Vol. 17, pp. 74-75, 1981.

23. R. Patel, "Computation of minimal order state-space realizations and observability indices using orthogonal transformations", Int. J. Contr., Vol. 33, pp. 227-246, 1981.

24. M. Konstantinov, P. Petkov, N. Christov, "Synthesis of linear systems with desired equivalent form", Journ.CACM, Vol. 6, pp. 27-35, 1980.

25. M. Wonham, "Linear multivariable control. A geometric approach", New York : Springer, 1974.

26. G. Stewart, "Algorithm 506 : HQR3 and EXCHNG, Fortran subroutines for calculating and ordering the eigenvalues of a real upper Hessenberg matrix", ACM TOMS, Vol. 2, pp. 275-280, 1976.

27. J. Kautsky, N. Nichols, P. Van Dooren, "Robust pole assignment in linear state feedback", Int. J. Contr., to appear.

28. P. Van Dooren, "Deadbeat control : a special inverse eigenvalue problem", BIT, to appear.

29. B. Anderson, J. Moore, "Optimal filtering", Englewood Cliffs : Prentice Hall, 1979.

30. M. Verhaegen, P. Van Dooren, "An efficient implementation of square root filtering : error analysis, complexity and simulation on flight-path reconstruction", Proceedings 6th Int. Conf. Anal. & Opt., Lecture Notes in Control & Information Sciences, Vol. 63, pp. 250-267, 1984.

31. R. Bartels, G. Stewart, "Solution of the matrix equation AX+XB=C", Comm. ACM, Vol. 15, pp. 820-826, 1972.

32. A. Barraud, "A numerical algorithm to solve $A^TXA-X=Q$", IEEE Trans. Aut. Contr., Vol. AC-22, pp. 883-885, 1977.

33. S. Hammarling, "Numerical solution of the stable non-negative definite Lyapunov equation", IMA J. Num. Anal., Vol. 2, pp. 303-323, 1982.

34. S. Hammarling, "Newton's method for solving the algebraic Riccati equation", NPL Report DITC 12/82, National Physical Laboratory, 1983.

35. R. Byers, "Hamiltonian and symplectic algorithms for the algebraic Riccati equation", Ph. D. Thesis, Dept. Appl. Math., Cornell Univ., 1983.

36. C. Paige, C. Van Loan, "A Schur decomposition for Hamiltonian matrices", Lin. Alg. & Appl., Vol. 41, pp. 11-32, 1981.

37. D. Luenberger, "Observing the state of a linear system", IEEE Trans. Mil. Electr., Vol. MIL-8, pp. 74-80, 1964.

38. P. Van Dooren, "Reduced order observers : a new algorithm and proof", System & Control Letters, Vol. 4, pp. 243-251, 1984.

39. G. Golub, S. Nasch, C. Van Loan, "A Hessenberg-Schur method for the problem AX+XB=C", IEEE Trans. Aut. Contr., Vol. AC-24, pp. 909-913, 1979.

40. A. Emami-Naeini, P. Van Dooren, "Computation of zeros of linear multivariable systems", Automatica, Vol. 18, pp. 415-430, 1982.

41. A. Emami-Naeini, P. Van Dooren, "On the computation of transmissions zeros and transfer functions", Proceedings 21st IEEE Conf. Dec. & Contr., pp. 51-55, 1982.

42. A. Varga, V. Sima, "Numerically stable algorithm for transfer function matrix evaluation", Int. J. Contr., Vol. 33, pp. 1123-1133, 1981.

43. A. Laub, "Efficient multivariable frequency response computations", IEEE Trans. Aut. Contr., Vol. AC-26, pp. 407-408, 1981.

44. C. Moler, C. Van Loan, "Nineteen dubious ways to compute the exponential of a matrix", SIAM Rev., Vol. 20, pp. 801-836, 1978.

45. A. Varga, V. Sima, C. Varga, "On numerical simulation of linear continuous control systems", Proceedings Simulation '83 Symposium, Praha, 1983.

46. W. Enright, "Improving the efficiency of matrix operations in the numerical solution of stiff ordinary differential equations", ACM Trans. Math. Software, Vol. 4, pp. 127-136, 1978.

47. M. Epton, "Methods for the solution of AXD-BXC=E and its application in the numerical solution of implicit ordinary differential equations", BIT, Vol. 20, pp. 341-345, 1980.

48. C. Paige, "Covariance matrix representation in linear filtering", this volume.

49. A. Laub, "Schur techniques in invariant embedding for solving two point boundary value problem", Proceedings 21st IEEE Conf. Dec. & Contr., pp. 56-61, 1982.

50. M. Singer, S. Hammarling, "A canonical form for the algebraic Riccati equation", Proceedings MTNS-83 International Symposium, Lecture Notes in Control and Information Sciences, Vol. 58, pp. 389-405, 1983.

51. R. Eising, "Polynomial matrices and feedback", IEEE Trans. Aut. Contr., to appear.

52. R. Patel, P. Misra, "Numerical algorithms for eigenvalue assignment by state feedback", Proceedings of the IEEE, Vol. 72, Dec. 1984.

53 R. Patel, "Computation of matrix fraction descriptions of linear time-invariant systems", IEEE Trans. Aut. Contr., Vol. AC-26, pp. 148-161, Febr. 1981.

Paul Van Dooren
Philips Research Laboratory Brussels
Av. Van Becelaere, 2
B-1170 Brussels
BELGIUM

Michel Verhaegen
ESAT
Katholieke Universiteit Leuven
Kardinaal Mercierlaan 94
B-3030 Heverlee
BELGIUM