NUMERICAL ALGORITHMS AND SOFTWARE TOOLS FOR ANALYSIS AND MODELLING OF DESCRIPTOR SYSTEMS

Andras Varga

Department of Mechanical Engineering
Ruhr—University Bochum
Institute for Automatic Control
P.O. Box 102148, D—4630 Bochum, Germany.

Abstract. The paper presents a state–of–the art overview of algorithms and software for analysis and modelling of linear descriptor systems. The presentation focuses on algorithms for which reliable software implementations are already available. Most of mentioned software tools belong to a small Fortran subroutines library, called DESCRIPT, developed by the author. This library contains robust implementation of numerically reliable algorithms covering the basic computational problems in analysis and modelling of descriptor systems. DESCRIPT is the first systematized collection of software tools dedicated to descriptor systems. The analysis and modelling facilities for descriptor systems available in the interactive package CASAM are based exclusively on the subroutines from DESCRIPT.

Keywords. Descriptor systems; Numerical methods; Systems analysis; Numerical software.

INTRODUCTION

In the last few years, the development of reliable numerical algorithms for analysis and modelling of descriptor systems was an active area of research. The need for such algorithms arises primarily from an increased theoretical and practical interest for this more general system description. However, many standard state–space systems problems lead naturally to descriptor systems formulations and therefore can be reliably solved by only using descriptor systems computational techniques. A special application area of descriptor systems is in numerical manipulation of rational and polynomial matrices. Operations with such matrices like sum, product, rational inverse, greatest common divisor, etc. can be easily expressed in terms of equivalent descriptor systems operations.

Surprisingly, very few software implementations of algorithms for descriptor systems are reported in the literature. Several subroutines dealing with linear matrix pencils are available in the SLICOT library [1] and thus are suitable for solving some descriptor systems analysis problems. The new releases of the BIMAS [2], [4] and BIMASC [3], [4] libraries also includes software intended for descriptor systems.

In this paper we shall frequently refer to a small Fortran library, called DESCRIPT, containing exclusively subroutines dedicated to solve computational problems arising in the analysis and modelling of descriptor systems. This library originates from the routines available in BIMAS and BIMASC libraries and completed with new subroutines implementing newer algorithms. DESCRIPT is a stand alone library containing robust implementations of many sophisticated algorithms which are for the first time available as a systematized collection of software tools exclusively devoted to descriptor systems. The routines from DESCRIPT are also available, with proper interfaces, in the last release of the RASP Control Library [5]. These routines also underlaid the new analysis and modelling facilities provided for descriptor systems in the interactive package CASAM [6].

The purpose of this paper is to give an up to date overview of existing numerically reliable algorithms for solving computational problems encountered in the analysis and modelling of descriptor systems. Due to space limitations, the presentation is restricted only to algorithms for which reliable software implementations are already available. The discussion focuses around the software tools available in the DESCRIPT library, but also the software available in SLICOT will be mentioned. A short description of the interactive package CASAM is also included.

SIMILARITY TRANSFORMATIONS

Consider the linear descriptor system

$$\begin{aligned} E\lambda x(t) &= Ax(t) + Bu(t) \\ y(t) &= Cx(t) + Du(t) \end{aligned} \tag{1}$$

where E, A are n×n matrices, B is an n×m matrix, C is an p×n matrix and D is an p×m matrix, and where λ is either the differential operator d/dt or the advance operator z, depending on the type of the system. Generally, the matrix E can be singular, but we shall assume in what follows that the pencil $\lambda E-A$ is regular, that is, $\det(\lambda E-A)\not\equiv 0$. The system (1) will be alternatively referred to as the quadruple $\{\lambda E-A,B,C,D\}$ or as the triple $\{\lambda E-A,B,C\}$ if $D = 0$. Its transfer–function matrix (TFM) is the p×m rational matrix

$$G(\lambda) = C(\lambda I-A)^{-1}B + D \ . \tag{2}$$

Two systems $\{\lambda\tilde{E}-\tilde{A},\tilde{B},\tilde{C}\}$ and $\{\lambda E-A,B,C\}$ whose matrices are related by

$$\lambda\tilde{E}-\tilde{A} = Q(\lambda E-A)Z,\ \tilde{B} = QB,\ \tilde{C} = CZ, \tag{3}$$

where Q and Z are square invertible matrices, are called *similar* and the transformation (3) is called a *similarity transformation*. Note that similar systems with same D have the same TFM.

The similarity transformations are the basic preprocessing tools for most of the analysis and modelling problems addressed later. From numerical point of view, it is important that the transformation matrices Q and Z to be well–conditioned (ideally orthogonal) matrices. Many useful reductions of systems matrices to special condensed forms can be done by using exclusively orthogonal transformations. The following transformations using an orthogonal pair (Q,Z) are implemented in DESCRIPT:

1. Reductions of E to upper–triangular form by using the QR or RQ decompositions of E [7].

2. Reduction of E to a diagonal form by using the singular value decomposition of E [7].

3. Reduction of the pair (A,E) to the generalized Hessenberg form (GHF) or the generalized real Schur form (GRSF) by using the QZ–algorithm [8]. Software for computing the complete QZ–decomposition of a pair (A,E) is available also in SLICOT.

4. Reordering of the generalized eigenvalues of a pair (A,E) in GRSF such that

$$Q(\lambda E-A)Z = \begin{bmatrix} \lambda E_{11}-A_{11} & \lambda E_{12}-A_{12} \\ 0 & \lambda E_{22}-A_{22} \end{bmatrix}, \tag{4}$$

where $\sigma(A_{11},E_{11}) \cap \sigma(A_{22},E_{22}) = \phi$ and $\sigma(A_{11},E_{11}) \subset \Gamma$, a given symmetric region of the complex plane ($\sigma(A,E)$ denotes the generalized eigenvalues of the pair (A,E)). The reordering algorithm was proposed in [9] and corresponding software is also available in SLICOT.

5. Finite–infinite spectrum separation of the pair (A,E) in the form (4), where the pair (A_{11},E_{11}) has only finite generalized eigenvalues (E_{11} is invertible) and the pair (A_{22},E_{22}) has only infinite generalized eigenvalues (E_{22} nilpotent, A_{22} invertible). The implemented algorithm in DESCRIPT is presented in [10] and is based on a more general procedure proposed in [11]. A more efficient method (of complexity $0(n^3)$) to compute this separation was proposed in [12] and software for this method is available in SLICOT.

6. Reduction of the system $\{\lambda E-A,B,C\}$ to the *orthogonal controllability form* (OCF)

$$Q(\lambda E-A)Z = \begin{bmatrix} \lambda E_c-A_c & * \\ 0 & \lambda E_{nc}-A_{nc} \end{bmatrix} \begin{matrix} r \\ n-r \end{matrix}, \tag{5a}$$
$$\qquad\qquad\qquad\qquad r \qquad\quad n-r$$

$$QB = \begin{bmatrix} B_c \\ 0 \end{bmatrix} \begin{matrix} r \\ n-r \end{matrix}, \quad CZ = [\ C_c\ C_{nc}\ ], \tag{5b}$$
$$\qquad\qquad\qquad\qquad\qquad\qquad r \quad n-r$$

where the system $\{\lambda E_c-A_c,B_c,C_c\}$ is controllable. The implemented algorithm was proposed in [13] and has complexity $0(n^3)$. Earlier proposed algorithms (as for instance those in [14] or [15]) have $0(n^4)$ computational complexity.

A further reduction of pencil (4) to a block–diagonal form is only possible by using a non–orthogonal similarity transformation. The off–diagonal term in (4) can be annihilated by pre– and post–multiplying of (4) with $\begin{bmatrix} I & -L \\ 0 & I \end{bmatrix}$ and $\begin{bmatrix} I & R \\ 0 & I \end{bmatrix}$, respectively, where L and R satisfy the generalized Sylvester equations

$$\begin{aligned} A_{11}R-LA_{22} &= -A_{12} \\ E_{11}R-LE_{22} &= -E_{12}\ . \end{aligned} \tag{6}$$

A numerically reliable algorithm to solve (6) was proposed in [16] and corresponding software is available in SLICOT. The reduction of the pair (A,E) to a block–diagonal form has applications in computing additive decompositions of rational matrices with respect to specified regions [17]. In particular, by using the block–diagonal form resulting from a finite–infinite spectrum separation, it is possible to separate the proper and the polynomial parts of a given TFM.

ANALYSIS OF DESCRIPTOR SYSTEMS

Typical computational analysis problems consist of determining by using reliable numerical methods systems properties as for example stability, controllability, observability, etc. or evaluating system specific parameters as for instance poles and zeros, frequency responses etc. Many of the condensed forms obtainable by the above presented similarity transformations are useful in performing such computations.

The stability of a descriptor system can be assessed on the basis of poles, computed as the generalized eigenvalues of the pair (A,E) by using the QZ–algorithm. The controllability can be determined on the basis of the OCF (5). If the system is uncontrollable ($n\neq r$), the stabilizability of the system can be assessed by computing the uncontrollable poles of the system (the generalized eigenvalues of the pair (A_{nc},E_{nc})). The impulsive or impulse–free behavior of a descriptor system can be easily determined from the finite–infinite spectrum separation. The frequency response

$$G(j\omega) = C(j\omega I - A)^{-1}B + D$$

can be efficiently evaluated for various frequency values after a preliminary reduction of the pair (A,E) to the GHF. Many other analysis problems (see for example [18]) can be addressed by using or extending the previous algorithms.

Of particular interest for the analysis of descriptor systems is the computation of zeros, defined as the *Smith zeros* of the (n+p)×(n+m) system matrix

$$S(\lambda) = \begin{bmatrix} \lambda E - A & B \\ -C & D \end{bmatrix} . \qquad (7)$$

A numerically reliable method to compute the system zeros can be viewed as a powerful analysis tool because properties of the system (1) as for instance stability, controllability, stabilizability, observability or detectability can be easily assessed by computing the zeros of particular system matrices (for $p = 0$ and/or $m = 0$).

A general algorithm for computing both finite and infinite zeros is based on the reduction of pencil (7) to the Kronecker's canonical form, for which, an efficient and numerically stable algorithm is described in [12] and corresponding software is available in SLICOT. For computing only the finite zeros, a somewhat more efficient numerically stable algorithm which exploits the structure of the pencil (7) was proposed in [19]. An implementation of this algorithm is provided in DESCRIPT. An alternative algorithm proposed in [20], is restricted to minimal systems and involves a delicate matrix inversion step.

MODELLING OF DESCRIPTOR SYSTEMS

The computational problems addressed in this section concerns with conversions between different generalized systems representations. Besides the descriptor representation (1) and the TFM representation (2), another description which arises in practice is the polynomial matrix representation

$$\begin{aligned} T(\lambda)\xi(t) &= U(\lambda)u(t) \\ y(t) &= V(\lambda)\xi(t)+W(\lambda)u(t) \end{aligned} , \qquad (8)$$

where ξ is the so–called internal state and $T(\lambda)$, $U(\lambda)$, $V(\lambda)$, $W(\lambda)$ are q×q, q×m, p×q and p×m polynomial matrices, respectively.

Given a TFM or polynomial matrix representation, it easy to construct equivalent descriptor representations by using straightforward formulas as those given in [14] or [13]. The resulting descriptor models are generally non–minimal. The computation of minimal order (irreducible) descriptor representations from non–minimal ones can be performed by successively removing the uncontrollable and unobservable parts of the systems. These eliminations can be performed in a numerically stable way by using the algorithm which computes the OCF (5) of a descriptor system [13]. The evaluation of the TFM (2) corresponding to the descriptor system (1) can be also reliably performed by using the poles–zeros algorithm of [10]. For all these computations, robust software implementations are available in DESCRIPT. Numerically reliable algorithms for computing minimal realizations and for evaluating TFMs have been also proposed in [21] and [22], respectively.

INTERACTIVE SOFTWARE TOOLS

Facilities for interactive analysis and modelling of descriptor systems are available in the last release of the interactive package CASAM (Computer Aided Systems Analysis and Modelling) [6]. All main computational routines from DESCRIPT are provided with interactive interfaces and can be easily accessed by the means of a simple command language. Each command corresponds to a stand alone executable program. A flexible data organization allows an easy communication between programs. The IBM PC/AT version of CASAM can be used to solve problems with 35–40 state variable, all computations being done in double precision. A full interface with MATLAB, at the data level, is also provided. CASAM is apparently the only existing interactive package offering computational facilities for analysis and modelling of descriptor systems.

CONCLUSIONS

We presented a succinct up to date overview of numerically reliable algorithms and associated software tools for the analysis and modelling of descriptor systems. Many of mentioned algorithms have however a wider application area. They are frequently used, for instance, as preliminary preprocessing steps in controllers and observers synthesis algorithms. Moreover, many of descriptor systems techniques have applications in solving problems arising in the standard state–space theory. The Fortran library DESCRIPT and the interactive package CASAM are apparently the only existing systematized collections of software tools for handling descriptor systems.

Acknowledgement. This work was carried out with the assistance of the Alexander von Humboldt Foundation, Bonn, F.R.G.

REFERENCES

[1] A. van den Boom, A.Brown, A.Geurts, S.Hammarling, R.Kool, M.Vanbegin, P. Van Dooren and S. Van Huffel, SLICOT, a subroutine library in control and systems theory. *Prepr. 5th IFAC CADCS '91 Symp.*, Swansea, UK, p.63—68, 1991.

[2] A.Varga and V.Sima, BIMAS — A basic mathematical package for computer aided systems analysis and design, *Proc. 9th IFAC Congress,* Budapest, Pergamon Press, 1985.

[3] A.Varga and A.Davidoviciu, BIMASC — A package of Fortran subprograms for analysis, modelling, design and simulation of control systems, *Prepr. 3rd IFAC Symp. on CAD in Control and Engineering Systems,* Copenhagen, Pergamon Press, 1985.

[4] D.K.Frederick, C.J.Herget, R.Kool, M.Rimvall, *The Extended List of Control Software,* WGS Edition, 1992.

[5] G.Grübel and H.D.Joos, RASP and RSYST — Two complementary program libraries for concurrent control engineering, *Prepr. 5th IFAC CADCS '91 Symp.,* Swansea, UK, p.101—106, 1991.

[6] A.Varga, CASAM — An interactive package for computer aided systems analysis and modelling, in *"Systems Analysis and Simulation"*, Eds. A.Sydow, M.Thoma and R.Vichnevetsky, vol.II, p.120—123, Akademie—Verlag Berlin, 1985.

[7] G.H.Golub and C.F. Van Loan, *Matrix Computations,* The John Hopkins University Press, Baltimore, 1983.

[8] C.B.Moler and G.W.Stewart, An algorithm for generalized matrix eigenvalue problems, *SIAM J. Numer. Anal.,* **10**, p.241—256, 1973.

[9] P. Van Dooren, Algorithm 590. DSUBSP and EXCHQZ: Fortran subroutines for computing deflating subspaces with specified spectrum, *ACM Trans. Math. Software,* **8**, p.376—382, 1982.

[10] A.Varga, Computation of transfer function matrices of generalized state—space models, *Int. J. Control,* **50**, p.2543—2561, 1990.

[11] P. Van Dooren, Computing the eigenvalues of a polynomial matrix, *Proc. IBM–NFWO Symp.,* Brussels, 1979.

[12] Th. Beelen and P. Van Dooren, An improved algorithm for computation of Kronecker's canonical form of a singular pencil, *Lin. Alg. Appl.,* **105**, p.9—65, 1988.

[13] A.Varga, Computation of irreducible generalized state—space realizations, *Kybernetika,* **26**, p.89—106, 1990.

[14] P. Van Dooren, The generalized eiegenstructure problem in linear system theory, *IEEE Trans. Autom. Contr.,* **AC—26,** p.111—129, 1981.

[15] K.—W.E. Chu, A controllability condensed form and a state feedback pole assignment algorithm for descriptor systems, *IEEE Trans. Autom. Contr.,* **AC—33,** p.366—370, 1988.

[16] B.Kagström and L.Westin, Generalized Schur methods with condition estimators for solving the generalized Sylvester equation, *IEEE Trans. Autom. Contr.,* **AC—34**, p.745—751, 1989.

[17] B.Kagström and P. Van Dooren, Additive decomposition of a transfer function with respect to a specified region, *Proc. MTNS Symp.,* Brussels, 1989.

[18] F.L.Lewis, A tutorial on the geometric analysis of linear time—invariant implicit systems, *Automatica,* **28**, p.119—137, 1992.

[19] A.Varga, Computation of zeros of generalized state—space systems, *Prepr. 5th IFAC CADCS '91 Symp.,* Swansea, UK, p.164—167, 1991.

[22] P.Misra, G.Gu and P. Van Dooren, Characterization and computation of transmission zeros of singular multivariable systems, *Proc. 29th IEEE CDC,* Honolulu, Hawaii, 1990.

[21] P.Misra, Hessenberg—triangular reduction and transfer function matrices of singular systems, *IEEE Trans. Circ. and Syst.,* **CAS—36,** p.907—912, 1989.

[22] P.Misra and R.V.Patel, Computation of minimal order realizations of generalized state space systems, *Circ., Syst. and Sign. Process.,* **8**, p.49—70, 1989.