Automatica 47 (2011) 244–248

Contents lists available at ScienceDirect

Automatica

journal homepage: www.elsevier.com/locate/automatica

Technical communique

An efficient algorithm for the solution of a coupled Sylvester equation appearing in descriptor systems[☆]

Amir Shahzad[a], Bryn Ll. Jones[b], Eric C. Kerrigan[a,c,*], George A. Constantinides[a]

[a] Department of Electrical and Electronic Engineering, Imperial College London, London SW7 2AZ, UK

[b] The Scottish Association for Marine Science, Scottish Marine Institute, Oban, Argyll PA37 1QA, UK

[c] Department of Aeronautics, Imperial College London, London SW7 2AZ, UK

ARTICLE INFO

Article history:
Available online 4 December 2010

Keywords:
Descriptor systems
Matrix equations
Numerical algorithms
Coupled Sylvester equation
Efficient algorithms

ABSTRACT

Descriptor systems consisting of a large number of differential–algebraic equations (DAEs) usually arise from the discretization of partial differential–algebraic equations. This paper presents an efficient algorithm for solving the coupled Sylvester equation that arises in converting a system of linear DAEs to ordinary differential equations. A significant computational advantage is obtained by exploiting the structure of the involved matrices. The proposed algorithm removes the need to solve a standard Sylvester equation or to invert a matrix. The improved performance of this new method over existing techniques is demonstrated by comparing the number of floating-point operations and via numerical examples.

1. Introduction

Descriptor systems, also known as singular systems, implicit systems or generalized state-space systems, emerge in many engineering applications (Dai, 1989; Kunkel & Mehrmann, 2006). For example, in fluid mechanical systems, a descriptor system is produced by the discretization of Navier–Stokes equations (Jones, Kerrigan, & Morrison, 2009). Descriptor systems typically consist of coupled differential and algebraic equations. As a consequence, the control of descriptor systems is less well-understood than that for conventional state-space systems. However, it is often possible, via a sequence of transformations (Gerdin, 2006), to completely decouple the differential and algebraic parts of a descriptor system, thus enabling the application of standard state-space control theory to this class of system.

There are three major steps in the transformation: a generalized Schur decomposition, also known as a Weierstrass–Schur form; solving a coupled Sylvester equation, also called a generalized Sylvester equation; construction of appropriately-defined transformation matrices (Gerdin, Schön, Glad, Gustafsson, & Ljung, 2007). The first step is extensively well-studied in the field of numerical linear algebra (Golub & Van Loan, 1996; Kågström & Wiberg, 2000). Various existing methods for transforming a matrix into a Jordan–Schur form and a matrix pencil into a Weierstrass–Schur form are compared by Kågström and Wiberg (2000). Furthermore, these methods are extended to extracting the partial information corresponding to dominant eigenvalues from large scale matrices and matrix pencils. The solution and perturbation analysis of a coupled Sylvester equation is presented in Kågström (1994) and Kågström and Westin (1989). In Kågström and Westin (1989) the Schur method (Bartels & Stewart, 1972) and the Hessenberg–Schur method (Golub, Nash, & Van Loan, 1979), which are used in solving a standard Sylvester equation, are extended for a coupled Sylvester equation. In Jones et al. (2009) the coupled Sylvester equation is transformed into a standard Sylvester equation and then standard techniques for solving a Sylvester equation are used.

This paper focuses on the efficient solution of the coupled Sylvester equation. The computational advantage over existing methods is obtained by exploiting the special structure of the matrices involved in the transformation of a DAE. The main contribution of this paper is to present a new algorithm for the solution of the above-mentioned coupled Sylvester equation, which is not only computationally more efficient than existing techniques, but also possesses the following important characteristics:

- no need to take an inverse of a matrix,
- no matrix by matrix multiplication, and
- no need to solve a standard Sylvester equation.

☆ This paper was not presented at any IFAC meeting. This paper was recommended for publication in revised form by Associate Editor Mikael Johansson under the direction of Editor André L. Tits.

* Corresponding author at: Department of Electrical and Electronic Engineering, Imperial College London, London SW7 2AZ, UK. Tel.: +44 0 2075945139x6343; fax: +44 0 2075946282.

E-mail addresses: a.shahzad07@imperial.ac.uk (A. Shahzad), bryn.jones@sams.ac.uk (B.L. Jones), e.kerrigan@imperial.ac.uk (E.C. Kerrigan), g.constantinides@imperial.ac.uk (G.A. Constantinides).

0005-1098/$ – see front matter © 2010 Elsevier Ltd. All rights reserved.
doi:10.1016/j.automatica.2010.10.038

2. Problem formulation

Consider a linear differential–algebraic equation (DAE) of the form

$$E\dot{x}(t) = Fx(t) + Gu(t), \quad x(t_0) = x_0, \tag{1}$$

where $E, F \in \mathbb{C}^{n\times n}$, $G \in \mathbb{C}^{n\times m}$, $x(t)$ is the state vector and $u(t)$ is the input. Solving (1) for $x(t)$ with given initial condition x_0 and $u(\cdot)$ when E is non-singular is straightforward. In this paper, we assume that E is singular. Therefore, (1) cannot be solved by a standard linear ordinary differential equation (ODE) solver. To overcome this problem, we have adopted the procedure of Gerdin (2006), which transforms (1) into the following set of linear ODEs and a set of algebraic equations:

$$\dot{z}_1(t) = Az_1(t) + B_1u(t), \tag{2a}$$

$$z_2(t) = -\sum_{i=0}^{k-1} N^iB_2\frac{\mathrm{d}^iu(t)}{\mathrm{d}t^i}, \tag{2b}$$

where $A \in \mathbb{C}^{p\times p}$, $B_1 \in \mathbb{C}^{p\times m}$, $B_2 \in \mathbb{C}^{q\times m}$, $N \in \mathbb{C}^{q\times q}$ is a nilpotent matrix of index k i.e. $N^k = 0$ and $n = p + q$. Note that $z_1(t)$ and $z_2(t)$ are decoupled. Let us call (2) the *standard form* of (1). The theoretical background and the procedure to compute the matrices involved in this standard form are described next.

Definition 1 (*Golub & Van Loan, 1996*)**.** Let $E, F \in \mathbb{C}^{n\times n}$ be two matrices. A *matrix pencil* is a set of all matrices of the form $F - \lambda E$ with $\lambda \in \mathbb{C}$. The eigenvalues of this matrix pencil are defined by $\lambda(F, E) := \{s \in \mathbb{C} : \det(F - sE) = 0\}$.

Definition 2 (*Kunkel & Mehrmann, 2006*)**.** A matrix pencil $F - \lambda E$ is called *regular* if there exists an $s \in \mathbb{C}$ such that $\det(F - sE) \neq 0$, or equivalently, $\lambda(F, E) \neq \mathbb{C}$.

The regularity of a matrix pencil $F - \lambda E$ is equivalent to the existence and uniqueness of the solution for system (1) (Dai, 1989).

Lemma 1 (*Gerdin, 2006*, *Lemma 2.1*)**.** *Consider the system* (1)*. If the matrix pencil* $F - \lambda E$ *is regular, then there exist non-singular matrices* P_1 *and* Q_1 *such that*

$$P_1EQ_1 = \begin{bmatrix} E_1 & E_2 \\ 0 & E_3 \end{bmatrix} \quad \textit{and} \quad P_1FQ_1 = \begin{bmatrix} F_1 & F_2 \\ 0 & F_3 \end{bmatrix}, \tag{3}$$

where $E_1 \in \mathbb{C}^{p\times p}$ *is non-singular,* $E_3 \in \mathbb{C}^{q\times q}$ *is upper triangular with all diagonal elements zero,* $F_3 \in \mathbb{C}^{q\times q}$ *is non-singular and upper triangular,* $E_2, F_2 \in \mathbb{C}^{p\times q}$*, and* $F_1 \in \mathbb{C}^{p\times p}$*.*

The generalized Schur decomposition (3) and the subsequent reordering of the diagonal elements of E_1 can be done using MATLAB's `qz` and `ordqz` functions, respectively. These functions call LAPACK routines `zgges` and `ztgsen` for complex matrices.

Remark 1. The decomposition of the matrix pencil $F - \lambda E$ by MATLAB's `qz` function produces upper triangular matrices. Therefore, E_1 and F_1 would be upper triangular.

There are three main steps in computing the standard state-space form of (1), which are listed below:

(1) Compute the generalized Schur decomposition of the matrix pencil $F - \lambda E$ as

$$P_1(F - \lambda E)Q_1 = \begin{bmatrix} F_1 & F_2 \\ 0 & F_3 \end{bmatrix} - \lambda \begin{bmatrix} E_1 & E_2 \\ 0 & E_3 \end{bmatrix}. \tag{4}$$

(2) Solve the following coupled Sylvester equation for L and R:

$$E_1R + LE_3 = -E_2, \tag{5a}$$

$$F_1R + LF_3 = -F_2, \tag{5b}$$

where $L, R \in \mathbb{C}^{p\times q}$.

(3) According to Lemma 1, if the matrix pencil $F - \lambda E$ in (1) is regular, there exist nonsingular matrices P and Q such that the transformation:

$$PEQQ^{-1}\dot{x}(t) = PFQQ^{-1}x(t) + PGu(t) \tag{6}$$

gives the system in standard form (2), where

$$P := \begin{bmatrix} E_1^{-1} & 0 \\ 0 & F_3^{-1} \end{bmatrix} \begin{bmatrix} I & L \\ 0 & I \end{bmatrix} P_1, \tag{7a}$$

$$Q := Q_1 \begin{bmatrix} I & R \\ 0 & I \end{bmatrix}, \qquad N := F_3^{-1}E_3, \qquad A := E_1^{-1}F_1, \tag{7b}$$

$$\begin{bmatrix} B_1 \\ B_2 \end{bmatrix} := PG, \qquad x(t) = Q \begin{bmatrix} z_1(t) \\ z_2(t) \end{bmatrix}. \tag{7c}$$

3. Solution of the coupled Sylvester equation

In this section, we propose a new and efficient algorithm for the solution of the coupled Sylvester equation (5) for R and L. This is done by exploiting the structure of the given matrices. From (5), we get

$$E_1R + L \begin{bmatrix} 0 & e^3_{12} & e^3_{13} & \cdots & e^3_{1,q} \\ 0 & 0 & e^3_{22} & \cdots & e^3_{2,q} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & e^3_{q-1,q} \\ 0 & 0 & 0 & \cdots & 0 \end{bmatrix} = -E_2, \tag{8a}$$

$$F_1R + L \begin{bmatrix} f^3_{11} & f^3_{12} & \cdots & f^3_{1,q} \\ 0 & f^3_{22} & \cdots & f^3_{2,q} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & f^3_{q,q} \end{bmatrix} = -F_2, \tag{8b}$$

where e^k_{ij} denotes the (i, j)th element of matrix E_k. The ith column of R, L, E_2, and F_2 is denoted by r_i, l_i, e^2_i, and f^2_i respectively. By comparing the first column of both sides of (8a), we get $E_1r_1 = -e^2_1$. Since E_1 is upper triangular, the above equation can be solved for r_1 using backward substitution. By comparing the first column of both sides of (8b), we get $l_1 = -\frac{1}{f^3_{11}}\left(f^2_1 + F_1r_1\right)$ if $f^3_{11} \neq 0$. Since F_3 is non-singular and upper triangular, $f^3_{ii} \neq 0$ for each i. Similarly, by comparing the ith column of (8a), we get $E_1r_i = -e^2_i - \sum_{k=1}^{i-1} e^3_{ki}l_k$ and by comparing the ith column of (8b) we get $l_i = -\frac{1}{f^3_{ii}}\left(f^2_i + F_1r_i + \sum_{k=1}^{i-1} f^3_{ki}l_k\right)$ if $f^3_{ii} \neq 0$. The complete algorithm is described in Algorithm 1. Algorithm 1 is well-defined for $q \geq 1$, which is proven next.

Proposition 1. *Consider* Lemma 1*. If* $\operatorname{rank}(E) = n - r$ *for* $r \geq 1$*, then* $q \geq 1$*.*

Proof. Let $\tilde{E} := P_1EQ_1$, where P_1 and Q_1 are non-singular matrices defined in Lemma 1, hence $\operatorname{rank}(P_1) = \operatorname{rank}(Q_1) = n$.