A New Block Algorithm for Full-Rank Solution of the Sylvester-Observer Equation

João Carvalho, Karabi Datta, and Yoopyo Hong

***Abstract*—A new block algorithm for computing a full rank solution of the Sylvester-observer equation arising in state estimation is proposed. The major computational kernels of this algorithm are: 1) solutions of standard Sylvester equations, in each case of which one of the matrices is of much smaller order than that of the system matrix and (furthermore, this small matrix can be chosen arbitrarily), and 2) orthogonal reduction of small order matrices. There are numerically stable algorithms for performing these tasks including the Krylov-subspace methods for solving large and sparse Sylvester equations. The proposed algorithm is also rich in Level 3 Basic Linear Algebra Subroutine (BLAS-3) computations and is thus suitable for high performance computing. Furthermore, the results on numerical experiments on some benchmark examples show that the algorithm has better accuracy than that of some of the existing block algorithms for this problem.**

***Index Terms*—Block algorithm Sylvester-observer equation, state estimation.**

I. Introduction

The matrix equation

$$XA - FX = GC \tag{1}$$

where the matrices $A \in \mathbb{R}^{n\times n}$, $C \in \mathbb{R}^{r\times n}$ are given and the matrices $X \in \mathbb{R}^{(n-r)\times n}$, $F \in \mathbb{R}^{(n-r)\times(n-r)}$ and $G \in \mathbb{R}^{(n-r)\times r}$ are to be found, is called the Sylvester-observer matrix equation [7].

The (1) is a variation of the well-known standard Sylvester equation $XA - TX = R$, in which A, T, and R are given and X is unknown. This is so called, because it arises in the construction of reduced-order observers [16] for the linear system

$$\begin{aligned} \dot{x}(t) &= \quad Ax(t) + Bu(t) \\ y(t) &= \qquad Cx(t) \end{aligned} \tag{2}$$

in the context of **state estimation**.

There are two basic approaches for state estimation [7]: Eigenvalue Assignment approach and the Sylvester-equation approach. Since one way of finding feedback matrix for eigenvalue assignment is via Sylvester equation [2], [12], [19], [20]; here we will pursue the Sylvester equation approach and, therefore, consider numerical solution of (1).

It is well known that the solvability of (1) is guaranteed if $\Omega(F)\bigcap\Omega(A) = \emptyset$, where $\Omega(M)$ denotes the spectrum of the matrix M. If F is indeed a stable matrix, then once a solution triple (X, F, G) of (1) is computed, an estimate $\hat{x}(t)$ to the state vector $x(t)$ can be computed by solving the following algebraic system of equations [16]:

$$\begin{bmatrix} X \\ C \end{bmatrix} \hat{x}(t) = \begin{bmatrix} z(t) \\ y(t) \end{bmatrix}. \tag{3}$$

Manuscript received September 16, 2002; revised April 14, 2003 and September 6, 2003. Recommended by Associate Editor J. Huang. The work of J. Carvalho was supported in part by the National Science Foundation under Grant ECS-0074411 and by the Brazilian CAPES under Grant BEX1624/98-9.

J. Carvalho is with the Departamento de Matemática Pura e Aplicada, Universidade Federal de Rio Grande do Sul, Brasil (e-mail: carvalho@mat.ufrgs.br).

K. Datta and Y. Hong are with the Department of Mathematical Sciences, Northern Illinois University, DeKalb, IL 60115 USA (e-mail: dattak@math.niu.edu; hong@math.niu.edu).

Digital Object Identifier 10.1109/TAC.2003.820150

Here, $z(t)$ is the state vector of the *observer* system

$$\dot{z}(t) = Fz(t) + Gy(t) + XBu(t),\ z(0) = z_0. \tag{4}$$

The state estimation problem clearly requires that the solution matrix X of (1) has full rank. Necessary conditions for existence of a full-rank solution X of (1) are that (A, C) is observable and (F, G) is controllable [18]. We will assume the observability of (A, C) and the matrices F and G will be constructed in such a way that the controllability of (F, G) will be satisfied.

A well-known method for solving the Sylvester-observer equation, based on the observer-Hessenberg decomposition of the observable pair (A, C) is due to Van Dooren [19]. The method is recursive in nature and computes the solution matrix X and the matrices F and G recursively, one row or column at a time.

Van Dooren's algorithm has been generalized to a block algorithm in [5]. The other block algorithms for this problem have been developed earlier in [3], [6], and [17].

In this paper, we present another block algorithm. A distinguishing feature of this new algorithm, compared to other above existing block algorithm is that it is guaranteed to give a full-rank solution X with a triangular structure. This structure can be exploited in computing the first $(n-r)$ components of the vector $\hat{x}(t)$ during the process of solving the linear algebraic system (3). The algorithm also seems to be more accurate then some of the other block algorithms.

The block algorithms are composed of Level 3 Basic Linear Algebra Subroutine (BLAS-3) computations. Such computations are ideally suited for achieving high-speed in today's high performance computers [10]. Indeed many traditional numerical linear algebra algorithms for matrix computations have been re-designed or new algorithms have been created for this purpose and a high-quality mathematical software package, called LAPACK [1] have been developed based on those block algorithms. Unfortunately, such algorithms in control are rare.

II. New Block Algorithm

We propose to solve (1) by imposing some structure on the right-hand side of the equation. This means that (like in the SVD-based method [6]) no reduction is imposed on the system matrix. To be more specific, given matrices A, C and a stable self-conjugate set $\mathcal{S}$, we construct matrices X, F and R satisfying

$$XA - FX = R \quad \Omega(F) = \mathcal{S} \tag{5}$$

and such that we are able to solve $GC = R$ for $G \in \mathbb{R}^{(n-r)\times r}$ later. As the solution X is being computed, a Householder-QR [13] based strategy will be applied so that at the end of the process, X will be a full-rank upper triangular matrix.

A. Development of the Algorithm

In this section, we propose our new block algorithm for solving (1). First, we investigate the solution of $GC = R$ for G. A solution exists only if the rows of the matrix R belong to the row space of the matrix C. Assume that the matrix C has full rank r and let $C = R_cQ_c$ be the thin RQ factorization of C [13], where $Q_c \in \mathbb{R}^{r\times n}$ and $R_c \in \mathbb{R}^{r\times r}$. If we choose

$$R = \begin{bmatrix} N_1 \\ \dots \\ N_q \end{bmatrix} Q_c = NQ_c \tag{6}$$

where $N_i \in \mathbb{R}^{n_i \times r}, i = 1, \ldots, q$, and $n_1 + n_2 + \cdots + n_q = n - r = s$, then we can find a solution $G \in \mathbb{R}^{(n-r)\times r}$ of $GC = R$ where $G_i R_c = N_i$, $i = 1, \ldots, q$ and

$$G = \begin{bmatrix} G_1 \\ \ldots \\ G_q \end{bmatrix}. \tag{7}$$

In particular, the choice $N_1 = I_r$ ensures that $rank(R) = rank(C) = r$.

Second, we partition X and F conformally

$$X = \begin{bmatrix} X_1 \\ \ldots \\ X_q \end{bmatrix} \quad F = \begin{bmatrix} F_{11} & 0 & \ldots & 0 \\ F_{21} & F_{22} & \ldots & \vdots \\ \vdots & \vdots & \ddots & \vdots \\ F_{q1} & F_{q,q-1} & & F_{qq} \end{bmatrix}. \tag{8}$$

Note that

$$\begin{aligned} &\mathrm{rank}\,[G \quad FG \cdots F^{n-r-1}G] \\ =&\mathrm{rank}\,[GR_c \quad FGR_c \cdot\cdot F^{n-r-1}GR_c] \\ &\text{(where } R_c \text{ is a full rank } r \times r \text{ matrix)} \\ =&\mathrm{rank}\,[N\ FN \cdot\cdot F^{n-r-1}N]. \end{aligned}$$

This shows that if we chose (F, N) as a controllable pair, then automatically (F, G) is controllable. To ensure the controllability of the pair (F, N), we choose $N_1 = I_r, N_2 = \cdot\cdot = N_q = 0$, and F as in (8) with full-rank blocks $F_{i,i-1}$ and $F_{ij} = 0$ for $j < i - 1$. Then, (F, N) is controllable.

Substituting (6) and (8) into (5) and equating corresponding blocks on the right and left-hand sides of (5), we obtain

$$X_1 A - F_{11} X_1 = N_1 Q_c \tag{9}$$

$$X_i A - F_{ii} X_i = N_i Q_c + \sum_{j=1}^{i-1} F_{ij} X_j, \quad i = 2, \ldots, q. \tag{10}$$

Therefore, as long as the elements of the given set $\mathcal{S}$ can be successfully distributed in self-conjugate subsets $S_i \in \mathbb{C}^{n_i}, i = 1, \ldots, q$, which are to be assigned as eigenvalues of the block matrices $F_{ii}, i = 1, \ldots, q$, we are able to construct matrices X, F and G from their blocks by computing them recursively using (9) and (10).

Let us define

$$X^i = \begin{bmatrix} X_1 \\ \ldots \\ X_i \end{bmatrix} \quad G^i = \begin{bmatrix} G_1 \\ \ldots \\ G_i \end{bmatrix} \tag{11}$$

$$F^i = \begin{bmatrix} F_{11} & \ldots & F_{1,i-1} & 0 \\ \ldots & \ldots & \ldots & 0 \\ F_{i-1,1} & & F_{i-1,i-1} & 0 \\ F_{i1} & \ldots & \ldots & F_{ii} \end{bmatrix}. \tag{12}$$

Next, we now update each X^i using QR factorization, so that the matrix X has an uppertriangle structure.

After each block X_i of the solution X has been computed, the matrix X^i defined previously will have the following structure:

$$X^i = \begin{bmatrix} * & * & * & * & * & * & * \\ & * & * & * & * & * & * \\ & & \cdot & \cdot & \cdot & \cdot & \cdot \\ & & & & * & * & * \\ * & * & * & * & * & * & * \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ * & * & * & * & * & * & * \end{bmatrix}.$$

The matrix X^i is now made upper triangular by premultiplying X^i with an appropriate orthogonal matrix Q_i (for example, Q_i can be product of suitable Householder matrices).

Symbolically, we write: $X^i \leftarrow Q_i^T X^i$ where X^i is updated to the matrix $Q_i^T X^i$ and the updated matrix $Q_i^T X^i$ is overwritten by X^i.

The matrix equation

$$X^i A - F^i X^i = G^i C \tag{13}$$

is then updated to

$$Q_i^T X^i A - Q_i^T \cdot F^i Q_i \cdot Q_i^T X^i = Q_i^T G^i C$$

meaning that it is possible to update the solution matrices, at every step of the orthogonal reduction, simply by computing

$$X^i \leftarrow Q_i^T X^i,\ F^i \leftarrow Q_i^T F^i Q_i,\ G^i \leftarrow Q_i^T G^i. \tag{14}$$

B. Proposed Block Algorithm for Solving $XA - FX = GC$

The aforementioned discussion leads to the following algorithm.

Input: Matrices $A \in R^{n\times n}$ and $C \in \mathbb{R}^{r\times n}$ of the system **(2)** and a self-conjugate set $\mathcal{S} \in \mathbb{C}^{n-r}$.
Output: Block matrices X, F and G, such that $\Omega(F) = \mathcal{S}$ and $XA - FX = GC$.
Assumption: System **(2)** is observable, C has full rank and $\Omega(A) \bigcap \mathcal{S} = \emptyset$.
Step 1: Set $s = n - r$, $\ell = r$ and $N_1 = I_{r\times r}$, $G_1 = R_c^{-1}$ and $n_1 = r$.
Step 2: Compute the thin RQ factorization of C **[13]**: $R_c Q_c = C$ where $Q_c \in \mathbb{R}^{r\times n}$ and $R_c \in \mathbb{R}^{r\times r}$.
Step 3: For $i = 1, 2, \ldots$ do Steps 4 to 10
 Step 4: Set $S_i \in \mathbb{R}^{\ell}$ to be a self-conjugate subset of the part of $\mathcal{S}$ that was not used yet.
 Step 5: Set $F_{ii} \in \mathbb{R}^{\ell\times\ell}$ to be any matrix in upper real Schur form satisfying $\Omega(F_{ii}) = S_i$.
 Step 6: Free parameter setup. If $i > 1$ set $N_i \in \mathbb{R}^{\ell\times n_i}$ and $F_{ij} \in \mathbb{R}^{\ell\times n_j}$, $j = 1, \ldots, i-1$ to be arbitrary matrices, so that (F, N) is controllable. Compute $G_i = N_i R_c^{-1}$.
 Step 7: Solve the Sylvester equation:

$$X_i A - F_{ii} X_i = N_i Q_c + \sum_{j=1}^{i-1} F_{ij} X_j$$

for $X_i \in \mathbb{R}^{\ell\times n}$.
 Step 8: Form X^i, G^i and F^i as in **(11)** and **(12)**. If $i > 1$, then set n_i as the number of rows of X_i that are linearly independent of the rows of X^{i-1}. If $n_i < \ell$, then

set $\ell = n_i$, and choose another set S_i from $\mathcal{S}$ and repeat Steps 5–8.

Step 9: Find, implicitly, an orthogonal matrix Q_i that reduces X^i to upper triangular form via left multiplication by Q_i^T, using, say householder matrices ,**[13]**. Then compute the matrix updates

$$X^i \leftarrow Q_i^T X^i \quad G^i \leftarrow Q_i^T G^i \quad F^i \leftarrow Q_i^T F^i Q_i$$

Step 10: If $n_1 + \ldots + n_i = s$, then let $q = i$ and exit loop.

Step 11: Form the matrices $X = X^q$, $F = F^q$ and $G = G^q$.

Remarks:

1) Some compatibility between the structure of the vector $\mathcal{S}$ and the parameters $n_i, i = 1, \ldots, q$ is required so that Step 4 is always possible to be accomplished.
2) The algorithm does not require reduction of the system matrices A and C. This feature is specially attractive when A is large and sparse. There now exist Krylov-subspace based methods for Sylvester equations, suitable for large and sparse computations [8]. If A is small and dense, the standard Hessenberg–Schur method [14] can be used.
3) In Step 6, it is possible to exploit the freedom of assigning F_{ij} to facilitate the solution of the Sylvester equation in Step 7. In particular, the diagonal blocks F_{ii} can be chosen in real-Schur forms, so that if the Hessenberg-Schur algorithm is used, then only the matrix A needs to be decomposed into Hessenberg form and *this is to be done once for all* the equations in step 7.
4) If matrix A is dense, an orthogonal similarity reduction $A \leftarrow P^T AP$, $C \leftarrow CP$, can be used so to bring Hessenberg structure to the matrix A. This will allow Step 7 to be computed efficiently. If (X_h, F, G) is the solution of this reduced problem, then $X = X_h P^T$ is the solution of the original problem.
5) The algorithm is rich in Level 3-BLAS computations and thus is suitable for high-performance computing using the software LAPACK [1], [10], which is especially designed for such a purpose.
6) The total flop-count of the algorithm is approximately
$$21\frac{2}{3}n^3 + 14 \cdot 5n^2 r$$
flops. This amount is smaller than that of Van Dooren's method [19]: $2rn^3 + 7n^3$, if r, the number of columns of the matrix C, is grater than 7.

III. An Illustrative Numerical Example

To illustrate the implementation of the proposed algorithm, we take matrices A and C, and the set S given in (I).

Step 1: $l = 2, N_1 = I_2$.

Step 2: The RQ factorization of C gives the matrices R_c and Q_c given in (II) at the bottom of the page.

Step 3: $i = 1$.

Step 4:
$S_1 = \{\, -1.0000 + 1.0000i \quad -1.0000 - 1.0000i \,\}$

Step 5: F_{11} to be

$$F_{11} = \begin{bmatrix} -1 & -1 \\ 1 & -1 \end{bmatrix}.$$

Clearly $\Omega(F_{11}) = S_1$.

Step 6: Set

$$N_1 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

for simplicity.

Step 7: Solve $X_1 A - F_{11} X_1 = N_1 Q_c$, using MATLAB Command *lyap*.

$$X_1 = \begin{bmatrix} -.134 & .280 & .067 & -.055 & .103 & -.444 & .235 \\ -.398 & -.104 & .438 & -.124 & .314 & -.027 & .219 \end{bmatrix}$$

Step 8: $n_1 = 2$, $l = \min\{2, 7-2-2\} = 2$.

Step 9: After the reduction with an orthogonal matrix, we have the matrix X_1, as shown in (III) at the bottom of the page.

Step 3: $i = 2$.

Step 4: $S_2 = \{-2.00 - 1.00i - 2.00 + 1.00i\}$.

$$A = \begin{bmatrix} 0.995 & 2.041 & -3.162 & 3.112 & -2.689 & 0.126 & 2.576 \\ 2.694 & 0.815 & 2.552 & 1.953 & 1.438 & -2.547 & 1.255 \\ 1.953 & -1.010 & 0.117 & 1.144 & 2.694 & 3.035 & 1.739 \\ -2.231 & -1.635 & 3.101 & 1.437 & -0.956 & -1.430 & 2.340 \\ 1.462 & 0.829 & 0.076 & -3.292 & -0.852 & -2.465 & -1.228 \\ 3.431 & -2.182 & -1.959 & 2.366 & 3.037 & 0.544 & 3.268 \\ -0.722 & -0.419 & 1.307 & -0.590 & 2.300 & 0.798 & -1.580 \end{bmatrix} \qquad \text{(I)}$$

$$C = \begin{bmatrix} .20 & 5.54 & 5.06 & 4.69 & 4.37 & 6.42 & 1.76 \\ 4.79 & 4.51 & 2.68 & 5.56 & .06 & 4.37 & 5.14 \end{bmatrix}$$

$$S = \{-1. -1. -1.\mathbf{i} -1. +1.\mathbf{i} -2. -1.\mathbf{i} -2. +1.\mathbf{i}\}.$$

$$R_c = \begin{bmatrix} -7.625\,162 & -9.136\,243 \\ & -11.264\,567 \end{bmatrix}$$

$$Q_c = \begin{bmatrix} 0.482 & -0.474 & -0.018 & -0.282 & 0.117 & 0.637 & 0.209 \\ -0.425 & 0.458 & 0.314 & -0.561 & 0.063 & 0.308 & 0.313 \end{bmatrix}. \qquad \text{(II)}$$

Step 5: Set F_{22} to be

$$F_{22} = \begin{bmatrix} -2 & -1 \\ 1 & -2 \end{bmatrix}$$

Then, clearly $\Omega(F_{22}) = S_2$.

Step 6: The free assignment is done via

$$N_2 = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}, \quad F_{21} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

by simplicity.

Step 7: Solving $X_2A - F_{22}X_1 = N_2Q_c + F_{21}X_1$, we obtain the matrix given by (IV) at the bottom of the page.

Step 8: $n_2 = 2$, $l = \min\{2, 7-2+(2+2)\} = 1$.

Step 9: After the reduction, we have the matrix given in (V) at the bottom of the page.

Step 3: $i = 3$.

Step 4: $S_3 = \{-1.0000\}$.

Step 5: We set $F_{33} = -1.0000$.

Step 6: Set $N_3 = [0 \quad 0]$, $F_{32} = [1 \quad 0]$.

Step 7: Solving $X_3A - F_{33}X_3 = N_3Q_c + F_{32}X_2$, we obtain X_3 given in (VI) at the bottom of the page.

Step 8: $n_3 = 1$, $l = \min\{2, 7-2+(2+2+1)\} = 0$.

Step 9: After the reduction, we have the final solution X given in (VII) at the bottom of the page.

Step 10: Since $n_1+n_2+n_3 = 7-2$ we set $p = 3$ and exit the loop.

Step 11: The algorithm finishes with matrices $X \in \mathbb{R}^{5\times 7}$, $F \in \mathbb{R}^{5\times 5}$ and $G \in \mathbb{R}^{5\times 2}$.

It can be shown that $\|XA - FX - GC\|_F = 2.4037 \times 10^{-15}$. □

A. Remark

In order to solve the state estimation problem, system (3) is reduced to upper triangular form by premultiplication by an orthogonal matrix Q_4^T, given by (VIII) at the bottom of the next page. Because of the upper triangular structure of the matrix X, the matrix Q_4 is obtained as the product of six appropriate Householder matrices, (which is not shown here). Therefore, the linear system (3) is reduced to

$$U\hat{x}(t) = Q_4^T \begin{bmatrix} z(t) \\ y(t) \end{bmatrix}$$

where the matrix U is given by (IX) at the top of the page 2228.

$$X_1 = \begin{bmatrix} .420 & -.107 & .271 & .171 & .263 & -.144 & .471 \\ & -.014 & -.198 & -.136 & -.217 & -.079 & .421 \end{bmatrix} \tag{III}$$

$$\begin{bmatrix} X_1 \\ X_2 \end{bmatrix} = \begin{bmatrix} .420 & -.107 & .271 & .171 & .263 & -.144 & .471 \\ & -.014 & -.198 & -.1359 & -.217 & -.079 & .421 \\ .072 & -.024 & .089 & .068 & .115 & -.032 & .030 \\ .213 & -.138 & .051 & .129 & .217 & -.044 & .235 \end{bmatrix}. \tag{IV}$$

$$\begin{bmatrix} X_1 \\ X_2 \end{bmatrix} = \begin{bmatrix} -.476 & .160 & -.275 & -.218 & -.347 & .151 & -.525 \\ 0 & .076 & 0.111 & -0.012 & -.035 & -.011 & -.072 \\ 0 & 0 & -.187 & -.145 & -.235 & -.078 & .414 \\ 0 & 0 & & .001 & -.003 & .030 & -.056 \end{bmatrix} [\quad]. \tag{V}$$

$$\begin{bmatrix} X_1 \\ X_2 \\ X_3 \end{bmatrix} = \begin{bmatrix} -.476 & .160 & -.275 & -.218 & -.347 & .151 & -.525 \\ 0 & .076 & .111 & -.012 & -.035 & -.011 & -.072 \\ 0 & 0 & -.187 & -.145 & -.235 & -.078 & .414 \\ 0 & 0 & 0 & .001 & -.003 & 0.030 & -.056 \\ .238 & -.156 & .069 & .165 & .270 & -.009 & .256 \end{bmatrix}. \tag{VI}$$

$$\begin{bmatrix} X_1 \\ X_2 \\ X_3 \end{bmatrix} = \begin{bmatrix} .532 & -.213 & .277 & .269 & .431 & -.140 & .584 \\ 0 & -.102 & -.124 & 0.43 & .084 & .047 & .051 \\ 0 & 0 & .188 & .148 & 0.239 & .082 & -.417 \\ 0 & 0 & 0 & -.007 & -.006 & -.030 & -.001 \\ 0 & 0 & 0 & 0 & .004 & -.026 & .057 \end{bmatrix} \tag{VII}$$

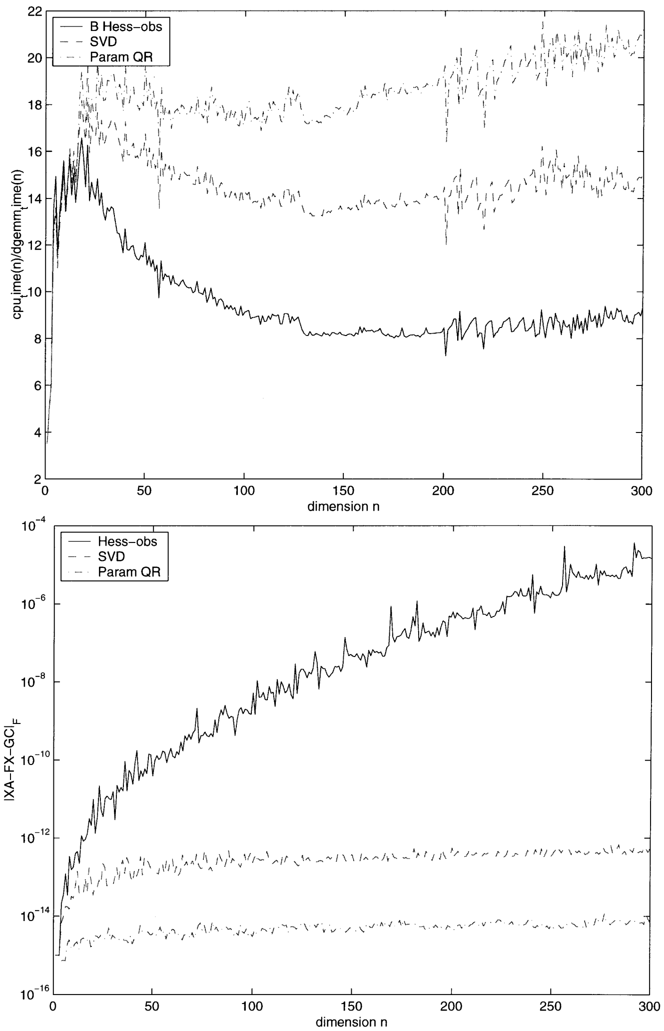

Fig. 1. Here n is the size of the system matrix $A = Pentoep(n)$, regarded as pentadiagonal. The dash-dotted line corresponds to the proposed algorithm, the dashed line to the SVD-based algorithm and the solid line to the Hessenberg reduction algorithm.

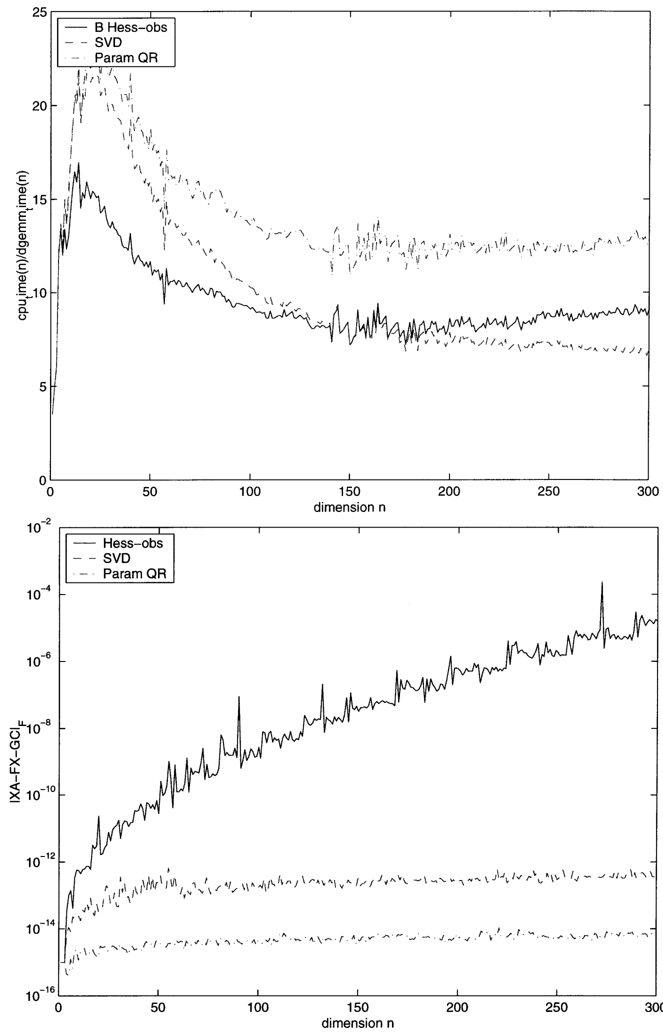

Fig. 2. Here n is the size of the system matrix $A = Pentoep(n)$, regarded as toeplitz. The dash-dotted line corresponds to the proposed algorithm, the dashed line to the SVD-based algorithm and the solid line to the Hessenberg reduction algorithm.

IV. Comparison of Efficiency and Accuracy With Existing Block Algorithms on Benchmark Examples

Figs. 1 and 2 show a comparison, in terms of accuracy and speed, of the proposed algorithm with the recent SVD-based [6] and the observer-Hessenberg reduction based [5] algorithms. The left-hand side graph of each figure compares the CPU-time and the right-hand side compares the accuracy. n is the size of the matrix A. The comparison is made on benchmark testing with the family *Pentoep* and with the family of Riemann matrices [15]. Speed is measured in terms of normalized CPU-time, that is, the required CPU-time is divided by the CPU-time of a call to the LAPACK [1]routine **dgemm** for multiplying two arbitrary matrices. Accuracy is measured by computing the Frobenius norm $\|XA - FX - GC\|_F$. Computations were performed in **Matlab 6** in **Pentium II** 400 MHz environment. The results of our experiment show that the proposed algorithm can achieve a better accuracy with a comparable speed for the problems tested.

$$Q_4 = \begin{bmatrix} -.11 & -.135 & -.942 & -.092 & -.271 & .011 & .001 \\ .0 & -.019 & .05 & .886 & -.4593 & -.013 & -.036 \\ .0 & .0 & -.29 & .45 & .844 & -.04 & -.010 \\ .0 & .0 & .0 & -.047 & -.014 & -.641 & -.77 \\ .0 & .0 & .0 & .0 & -.0285 & -.766 & .64 \\ -.042 & .990 & -.1248 & .005 & -.0449 & .001 & -.001 \\ -.99 & -.027 & .110 & .010 & .032 & -.001 & -.0 \end{bmatrix}. \tag{VIII}$$

$$U = \begin{bmatrix} -4.82 & -4.69 & -2.90 & -5.75 & -.29 & -4.6 & -5.24 \\ -.0 & 5.4 & 4.9 & 4.45 & 4.27 & 6.25 & 1.52 \\ .0 & .0 & -.66 & -.27 & -1.01 & -.209 & -.081 \\ .0 & .0 & -.0 & .16 & .163 & .167 & -.137 \\ .0 & .0 & .0 & -.0 & -.148 & -.061 & -.449 \\ .0 & .0 & .0 & -.0 & -.0 & .036 & -.026 \\ .0 & .0 & .0 & -.0 & -.0 & .0 & .039 \end{bmatrix}. \tag{IX}$$

V. Conclusion

A new block algorithm for solving the Sylvester-observer equation is proposed. The algorithm does not require the reduction of the system matrix A and is then ideally suitable for large and sparse computations by using the recently developed Krylov-subspace based methods. This algorithm is well-suited for implementation on high-performance computing using *LAPACK* and it seems to be accurate compared with similar ones; however, numerical stability properties have not been studied yet.

Acknowledgment

The authors would like to thank the reviewers for their comments and helpful suggestions.

References

[1] E. Anderson, Z. Bai, C. Bischof, S. Blackford, J. Demmel, J. Dongarra, J. DuCroz, A. Greenbaum, S. Hammerling, A. Mckenney, and D. Sorensen, *Lapack User's Guide*, 2nd ed. Philadelphia, PA: SIAM, 1995.

[2] S. P. Bhattacharyya and E. deSouza, "Pole assignment via Sylvester's equating," in *Syst. Control Letter*, vol. 1, 1982, pp. 261–283.

[3] C. Bischof, B. N. Datta, and A. Purkayastha, "A parallel algorithm for the Sylvester observer equation," *SIAM J. Sci. Comput.*, vol. 17, pp. 686–698, 1996.

[4] D. Calvetti, B. Lewis, and L. Reichel, "On the solution of large Sylvester-observer equation," *Numer. Lin. Alg. Applicat.*, vol. 8, pp. 435–452, 2001.

[5] J. Carvalho and B. N. Datta, "A new block algorithm for the Sylvester-observer equation arising in state-estimation," *Proc. IEEE Int. Conf. Decision Control*, pp. 3398–3403, 2001.

[6] B. N. Datta and D. Sarkissian, "Block algorithms for state estimation and functional observers," *Proc. IEEE Int. Symp. Computer-Aided Control System Design*, pp. 19–23, 2000.

[7] B. N. Datta, Numerical Methods for Linear Control Systems Design and Analysis. New York: Elsevier, 2003, to be published.

[8] ——, "Krylov subspace methods for large-scale matrix problems in control," *Future Generation Comput. Syst.*, vol. 19, pp. 1253–1263, 2003.

[9] K. Datta, "The matrix equation $AX - XB = R$ and its applications," *Lin. Alg. Applicat.*, vol. 109, pp. 91–105, 1988.

[10] J. Dongarra, I. Duff, D. Sorensen, and H. Van der Vorst, *Numerical Linear Algebra for High-Performance Computers*. Philadelphia, PA: SIAM, 1998.

[11] J. Doyle and G. Stein, "Robustness with observers," in *IEEE Trans. Automat. Contr.*, vol. AC-26, Jan. 1981, pp. 4–16.

[12] G. R. Duan, "Solutions of the equations $AV + BW = VF$ and their application to eigenstructure assignment in linear systems," *IEEE Trans. Automat. Contr.*, vol. 38, pp. 276–280, Feb. 1993.

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

[14] G. Golub, S. Nash, and C. Van Loan, "A hessenberg-schur method for the problem $AX + XB = C$," in *IEEE Trans. Automat. Contr.*, vol. AC-24, Nov. 1979, pp. 909–913.

[15] N. J. Highan, "The test matrix toolbox for matlab," Manchester Centre for Comput. Math., Univ. Manchester, Manchester, U.K., Numer. Anal. Rep. 276, 1995.

[16] D. Luenberger, "Observing the state of a linear system," *IEEE Trans. Mil. Electr.*, vol. 8, pp. 74–80, Jan. 1964.

[17] B. Shafai and S. P. Bhattacharyya, "An algorithm for pole assignment in high order multivariable systems," *IEEE Trans. Automat. Contr.*, vol. 33, pp. 870–876, Sept. 1988.

[18] E. DeSouza and S. P. Bhattacharyya, "Controllability, observability and the solution of $AX - XB = C$," *Lin. Alg. Applicat.*, vol. 39, pp. 167–188, 1981.

[19] P. Van Dooren, "Reduced order observers: A new algorithm and proof," *Syst. Control Lett.*, vol. 4, pp. 243–251, 1984.

[20] Y. Zhang and J. Wang, "Global exponential stability of recurrent neural networks for synthesizing linear feedback control system via pole assignment," *IEEE Trans. Neural Networks*, vol. 13, pp. 633–644, Oct. 2002.