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

João Carvalho, DMPA, Universidade Federal do RS, Brasil
Karabi Datta, Dep. M.Sc., Northern Illinois University, DeKalb, IL 60115, USA
Yoopyo Hong, Dep. M.Sc., Northern Illinois University, DeKalb, IL 60115, USA

***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 (i) solutions of ordinary Sylvester equations, in which case each one of the matrices is of much smaller order than that of the system matrix and furthermore, this small matrix can be chosen arbitrarily, (ii) orthogonal reduction of small order matrices. There are numerically stable algorithms for performing these tasks. The algorithm is rich in Level 3 Basic Linear Algebra Subprograms (BLAS-3) computations and 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.**

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 [6].

The problem of solving (1) arises in the construction of reduced order Observers [14] for the linear system

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

in the context of **State Estimation**.

It is well-known that the solvability of (1) is guaranteed if $\Omega(F) \bigcap \Omega(A) = \emptyset$. If F is indeed a stable matrix, 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 [14]:

$$\left[\begin{array}{c} X \\ C \end{array}\right] \hat{x}(t) = \left[\begin{array}{c} z(t) \\ y(t) \end{array}\right]. \tag{3}$$

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

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

which can have any initial condition $z(0) = z_0$.

The state estimation problem clearly requires that the solution matrix X of (1) has full rank. It is well-known [16] that necessary conditions for existence of a full rank solution X of (1) are that (A, C) is observable and (F, G) is controllable. 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.

The block algorithms are composed of Level-3 Blas (Basic Linear Algebra Subprograms) computations. Such computations are ideally suited for achieving high-speed in today's high performance computers [8]. 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, block algorithms in control are rare.

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 [17]. 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 by Carvalho and Datta [4]. Other block algorithms for this problem include [2], [5], [15].

There are two basic approaches for state estimation [6]: Eigenvalue Assignment approach and Sylvester-Observer equation approach. Since one way of finding feedback matrix for eigenvalue assignment is via Sylvester- observer equation [10], [16], [17], here we will pursue the Sylvester-observer equation approach.

In this paper, we present another block algorithm for solving the Sylvester-observer equation (1). This new algorithm seems to be more accurate than some of the block algorithms mentioned above and 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).

II. A 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 [5]) no reduction is imposed on the system matrix A. To be more specific, given matrices A, C and a stable self-conjugate set $\mathcal{S}$, we construct matrices X, F and R satisfying

$$XA - FX = R \tag{5}$$

First author partially supported by NSF grant ECS-0074411 and Brazilian CAPES grant BEX1624/98-9.

$$\Omega(F) = \mathcal{S} \tag{6}$$

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 [11] based strategy will reshape it so that at the end of the process X is a full-rank upper triangular matrix.

A. Development of the Algorithm

In this section, we propose our new block algorithm for solving equation (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 $C = R_c Q_c$ be the thin RQ factorization of C, where $Q_c \in \mathbb{R}^{r\times n}$ and $R_c \in \mathbb{R}^{r\times r}$. If we choose

$$R = \begin{bmatrix} N_1 \\ \dots \\ N_q \end{bmatrix} Q_c = NQ_c \tag{7}$$

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

$$G = \begin{bmatrix} G_1 \\ \dots \\ G_q \end{bmatrix}. \tag{8}$$

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

Second, we partition X and F conformally:

$$X = \begin{bmatrix} X_1 \\ \dots \\ X_q \end{bmatrix},\ F = \begin{bmatrix} F_{11} & & & \\ F_{21} & F_{22} & & \\ \vdots & \vdots & \ddots & \\ F_{q1} & F_{q,q-1} & & F_{qq} \end{bmatrix}. \tag{9}$$

Note that

$$\text{rank}\begin{bmatrix} G & FG & \dots & F^{n-r-1}G \end{bmatrix} = \text{rank} \begin{bmatrix} GR_c & FGR_c & \dots & F^{n-r-1}GR_c \end{bmatrix} = \text{rank}\begin{bmatrix} N & FN & \dots & F^{n-r-1}N \end{bmatrix}$$

where R_c is a full rank $r\times r$ matrix. This shows that if we chose (F, N) controllable, then automatically (F, G) is controllable. For example if we choose $N_1 = I_r, N_2 = \cdot\cdot = N_q = 0$, and F as in (9) with full-rank blocks $F_{i,i-1}$ and $F_{ij} = 0$ for $j < i-1$, then (F, N) is controllable.

Substituting (7) and (9) into (5) and equating corresponding blocks on the right and left hand sides of (5), we obtain

$$X_1 A - F_{11} X_1 = N_1 Q_c \tag{10}$$

$$X_i A - F_{ii} X_i = N_i Q_c + \sum_{j=1}^{i-1} F_{ij} X_j, \qquad i = 2,\dots,q. \tag{11}$$

Therefore, as long as the elements of the given set $\mathcal{S}$ can be successfully distributed in self-conjugate subsets $S_i \in \mathbb{C}^{n_i}$, $i = 1,\dots,q$, to be assigned as eigenvalues of the block matrices F_{ii}, $i = 1,\dots,q$, we are able to construct matrices X, F and G from their blocks computed recursively using (10) and (11).

We define

$$X^i = \begin{bmatrix} X_1 \\ \dots \\ X_i \end{bmatrix},\ G^i = \begin{bmatrix} G_1 \\ \dots \\ G_i \end{bmatrix} \tag{12}$$

$$F^i = \begin{bmatrix} F_{11} & \dots & F_{1,i-1} & 0 \\ \dots & \dots & \dots & 0 \\ F_{i-1,1} & & F_{i-1,i-1} & 0 \\ F_{i1} & \dots & \dots & F_{ii} \end{bmatrix}. \tag{13}$$

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

After each block X_i of the solution X has been computed, the matrix X^i defined above has the structure below:

$$X^i = \begin{bmatrix} * & * & * & * & * & * & * & * & * & * \\ & * & * & * & * & * & * & * & * & * \\ & & * & * & * & * & * & * & * & * \\ & & & & & & \dots & & \dots & \\ & & & & * & * & * & * & * & * \\ & & & & & * & * & * & * & * \\ * & * & * & * & * & * & * & * & * & * \\ & & & \dots & & & \dots & & & \\ * & * & * & * & * & * & * & * & * & * \end{bmatrix},$$

The matrix X^i is now made upper triangular form 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

$$X^i A - F^i X^i = G^i C \tag{14}$$

is updated to

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

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

$$X^i \leftarrow Q_i^T X^i\ ,\ F^i \leftarrow Q_i^T F^i Q_i\ ,\ G^i \leftarrow Q_i^T G^i. \tag{15}$$

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

The above 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: The 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 $C : R_c Q_c = C$ where $Q_c \in \mathbb{R}^{r\times n}$ and $R_c \in \mathbb{R}^{r\times r}$.

Step 3: For $i = 1, 2, \dots$ do Steps 4 to 10

Step 4: Set $S_i \in \mathbb{R}^{\ell}$ 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 Scour 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, such that (F, N) is controllable. Compute $G_i = N_i R_c^{-1}$.

Step 7: Solve the Sylvester equation using The Hessenberg-Schur Algorithm [11]:

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

for $X_i \in \mathbb{R}^{\ell \times n}$.

Step 8: Form X^i, G^i and F^i as in (12) and (13). If $i > 1$ then let n_i be the number of rows of X_i that are linearly independent of the rows of X^{i-1}. If $n_i < \ell$ then set $\ell = n_i$, choose another set S_i from $\mathcal{S}$ and do Steps 5 to 8 again.

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 [6]. Then compute the matrix updates

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

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, as long as we are able to exploit this structure in the solution of the subproblems in Step 7.
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 in the Hessenberg-Schur algorithm only the matrix A needs to be decomposed into Hessenberg form and *this is to be done once for all* the equations in step 6.
4) If matrix A is dense, an orthogonal similarity reduction $A \leftarrow P^T A P$, $C \leftarrow CP$, can be used so to bring Hessenberg structure to the matrix A. This allows Step 7 to be computed efficiently so that the whole algorithm requires $O(n^3)$ flops. 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 LAPACK [1], [8].

Flop-count:

- Reduction of A to Hessenberg form with implicit computation of P (if needed):
$$\frac{10n^3}{3} + 4n^2 r - 4r^3/3 \text{ flops}$$
- Step 2 (assuming explicit computation of Q_c [11]): $4n^2 r - 2nr^2 + 4r^3/3$ flops
- Step 8 : $2n^2 r(i-1) - r^3(i-1)^2$ flops
- Steps 9 and 10: $4\left[4nr^2 i - 2r^3 i^2 + 2nr^2 - 2r^3 i\right]$ flops
- Step 7 (using the Hessenberg-Schur method [12]): $10n^2 r + nr^2$ flops
- Recovery of X from X_h: $4n^3$ flops.

Therefore, this algorithm requires approximately

$$\frac{77n^3}{3} + \frac{29n^2 r}{2}$$

floating-point operations. The count for Step 7 assumes the worst-case scenario where the eigenvalues of F_{ii} are all non-real.

III. An Illustrative Numerical Example

To illustrate the implementation of the proposed algorithm, consider

$$A = \begin{bmatrix} .995 & 2.041 & -3.162 & 3.112 & -2.69 & .126 & 2.576 \\ 2.694 & 0.815 & 2.552 & 1.953 & 1.438 & -2.547 & 1.255 \\ 1.953 & -1.010 & .117 & 1.144 & 2.694 & 3.035 & 1.739 \\ -2.231 & -1.635 & 3.101 & 1.437 & -.956 & -1.430 & 2.340 \\ 1.462 & .829 & .076 & -3.292 & -.852 & -2.465 & -1.228 \\ 3.431 & -2.182 & -1.959 & 2.366 & 3.037 & .544 & 3.268 \\ -.722 & -.419 & 1.307 & -.590 & 2.300 & .798 & -1.580 \end{bmatrix}$$

$$C = \begin{bmatrix} 0.204 & 5.542 & 5.057 & 4.685 & 4.370 & 6.415 & 1.757 \\ 4.785 & 4.506 & 2.679 & 5.564 & 0.060 & 4.374 & 5.140 \end{bmatrix}$$

$$S = \left\{ \begin{array}{ccccc} -1. & -1.-1.\,\mathbf{i} & -1.+1.\,\mathbf{i} & -2.-1.\,\mathbf{i} & -2.+1.\,\mathbf{i} \end{array} \right\}$$

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

Step 2: the RQ factorization of C gives

$$R_c = \begin{bmatrix} -7.625162 & -9.136243 \\ & -11.264567 \end{bmatrix}$$

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

Step 3: $i = 1$.

Step 4: $S_1 = \left\{ \begin{array}{cc} -1.00 + 1.00i & -1.00 - 1.00i \end{array} \right\}$

Step 5: We set F_{11} to be

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

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

Step 6: The free assignment is done via

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

by simplicity.

Step 7: Solving $X_1 A - F_{11} X_1 = N_1 Q_c$ gives

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

as solution of this Sylvester equation.

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

Step 9: The orthogonal matrix that reduces X_1 is

$$Q_1 = \begin{bmatrix} -.318454 & -.947938 \\ -.947938 & .318454 \end{bmatrix}$$

and after the reduction

$$X_1 = \begin{bmatrix} .420 & -.107 & .271 & .171 & .263 & -.144 & .471 \\ 0 & -.014 & -.198 & -.136 & -.217 & -.079 & .421 \end{bmatrix}$$

$$F_{11} = \begin{bmatrix} -1 & 1 \\ -0.000000 & -1 \end{bmatrix}, \; G_1 = \begin{bmatrix} .0418 & .0503 \\ .1243 & -.1291 \end{bmatrix}$$

Step 3: $i = 2$.

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

Step 5: We set F_{22} to be

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

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

Step 6: The free assignment is done via

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

by simplicity.

Step 7: Solving $X_2A - F_{22}X_1 = N_2Q_c + F_{21}X_1$ gives X_2 such that

$$\begin{bmatrix} X_1 \\ X_2 \end{bmatrix} = \begin{bmatrix} .420 & -.107 & .271 & .171 & .263 & -.144 & .471 \\ .0 & -.014 & -.198 & -.1359 & -.217 & -.079 & .421 \\ .072 & -.024 & .089 & .068 & .115 & -.032 & .030 \\ .213 & -.138 & .051 & .129 & .217 & -.044 & .235 \end{bmatrix}$$

.

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

Step 9: The orthogonal matrix that reduces $\begin{bmatrix} X_1 \\ X_2 \end{bmatrix}$ is

$$Q_2 = \begin{bmatrix} -.881700 & .444581 & .112851 & .110531 \\ .0 & -.179335 & .951465 & -.250105 \\ -.151624 & .000728 & -.251158 & -.955996 \\ -.446782 & -.877603 & -.137469 & .106308 \end{bmatrix}$$

and after the reductions

$$\begin{bmatrix} X_1 \\ X_2 \end{bmatrix} = \begin{bmatrix} -.476 & .160 & -.275 & -.218 & -.347 & .151 & -.525 \\ .0 & .076 & .111 & -.012 & -.035 & -.011 & -.072 \\ .0 & .0 & -.187 & -.145 & -.235 & -.078 & .414 \\ .0 & .0 & .0 & .001 & -.003 & .030 & -.056 \end{bmatrix}$$

$$\begin{bmatrix} F_{11} & F_{12} \\ F_{21} & F_{22} \end{bmatrix} = \begin{bmatrix} -1.089 & -.355 & -1.289 & .661 \\ -.417 & -1.612 & -.292 & 1.061 \\ .869 & -.871 & -1.241 & -.194 \\ .082 & -1.098 & -.257 & -2.057 \end{bmatrix}$$

$$\begin{bmatrix} G_1 \\ G_2 \end{bmatrix} = \begin{bmatrix} -.0368 & -.0443 \\ -.0037 & .0455 \\ .1230 & -.1172 \\ -.0265 & .0378 \end{bmatrix}$$

Step 3: $i = 3$.

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

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

Step 6: The free assignment is done via $N_3 = \begin{bmatrix} 0 & 0 \end{bmatrix}$, $F_{32} = \begin{bmatrix} 1 & 0 \end{bmatrix}$.

Step 7: Solving $X_3A - F_{33}X_3 = N_3Q_c + F_{32}X_2$ gives X_3 such that

$$\begin{bmatrix} X_1 \\ X_2 \\ X_3 \end{bmatrix} = \begin{bmatrix} -.476 & .160 & -.275 & -.218 & -.347 & .151 & -.525 \\ .0 & .076 & .111 & -.012 & -.035 & -.011 & -.072 \\ .0 & .0 & -.187 & -.145 & -.235 & -.078 & .414 \\ .0 & .0 & .0 & .001 & -.003 & .030 & -.056 \\ .0.238 & -.156 & .069 & .165 & .270 & -0.009 & .0.256 \end{bmatrix}$$

.

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

Step 9: The orthogonal matrix that reduces $\begin{bmatrix} X_1 \\ X_2 \\ X_3 \end{bmatrix}$ is

$$Q_3 = \begin{bmatrix} -.894709 & .296784 & .050001 & -.325600 & .053847 \\ .0 & -.747318 & .099536 & -.648165 & .107192 \\ .0 & .0 & -.988717 & -.147791 & .024441 \\ .0 & .0 & .0 & -.163162 & -.986599 \\ .446649 & .594504 & .100160 & -.652229 & .107864 \end{bmatrix}$$

and after the reductions

$$\begin{bmatrix} X_1 \\ X_2 \\ X_3 \end{bmatrix} = \begin{bmatrix} .532 & -.213 & .277 & .269 & .431 & -.140 & .584 \\ .0 & -.102 & -.124 & .043 & .084 & .047 & .051 \\ .0 & .0 & .188 & .148 & .239 & .082 & -.417 \\ .0 & .0 & .0 & -.007 & -.006 & -.030 & -.001 \\ .0 & .0 & .0 & .0 & .004 & -.026 & .057 \end{bmatrix}$$

$$\begin{bmatrix} F_{11} & F_{12} & F_{13} \\ F_{21} & F_{22} & F_{23} \\ F_{31} & F_{32} & F_{33} \end{bmatrix} = \begin{bmatrix} -1.0712 & -.2134 & -1.5465 & -.3715 & .6611 \\ -.2554 & -1.1787 & -.3756 & -.2876 & .6524 \\ .8103 & -.8537 & -1.2097 & -.3026 & -.2851 \\ -.1410 & -.5662 & .0558 & -1.4735 & .7897 \\ .0974 & -.7623 & -.1601 & -.8292 & -2.0669 \end{bmatrix}$$

$$\begin{bmatrix} G_1 \\ G_2 \\ G_3 \end{bmatrix} = \begin{bmatrix} .0329 & .0397 \\ -.0081 & -.0472 \\ -.1238 & .1182 \\ .0005 & -.0039 \\ .0267 & -.0377 \end{bmatrix}$$

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 $X \in \mathbb{R}^{5\times7}$, $F \in \mathbb{R}^{5\times5}$ and $G \in \mathbb{R}^{5\times2}$ obtained from their blocks shown above.

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

$$eig(F) = \left\{ \begin{array}{c} -1.00000000000000 + 1.00000000000000\mathbf{i}, \\ -1.00000000000000 - 1.00000000000000\mathbf{i}, \\ -1.00000000000000, \\ -2.00000000000000 + 1.00000000000000\mathbf{i}, \\ -2.00000000000000 - 1.00000000000000\mathbf{i} \end{array} \right\}$$

□

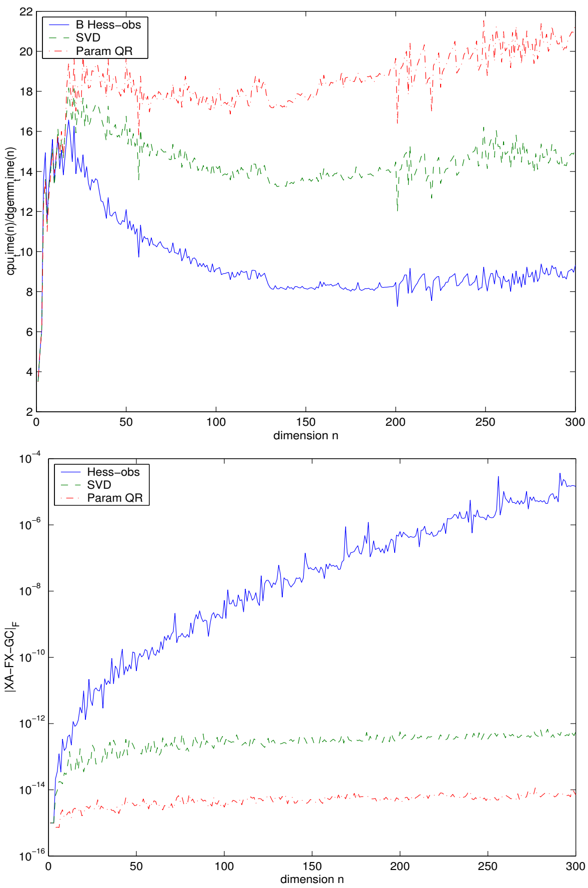

Fig. 1. Here n is the size of the system matrix $A = Pentoep(n)$, regarded as pentadiagonal. The dash-dotted line corresponds to the proposed algorithm, the dashed line to the SVD-based algorithm and the solid line to the Hessenberg reduction algorithm.

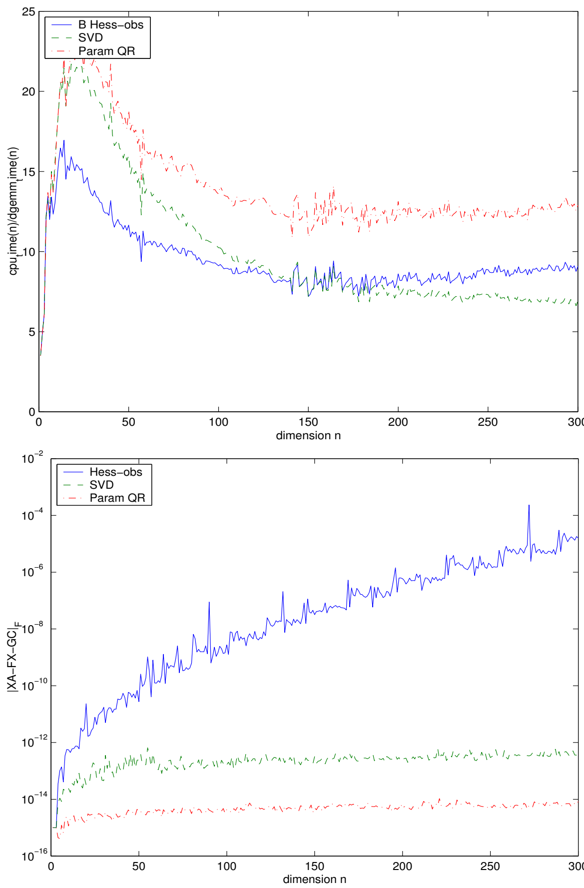

Fig. 2. Here n is the size of the system matrix $A = Pentoep(n)$, regarded as toeplitz. The dash-dotted line corresponds to the proposed algorithm, the dashed line to the SVD-based algorithm and the solid line to the Hessenberg reduction algorithm.

Comparison of Efficiency and Accuracy with existing block algorithms:

Figures 1 and 2 show a comparison, in terms of accuracy and speed, of the proposed algorithm with the recent SVD-based [5] and the observer-Hessenberg reduction based [4] algorithms. The comparison is made on benchmark testing with the family *Pentoep* of pentadiagonal toeplitz matrices [13]. 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 routine **dgemm** for multiplying two arbitrary matrices. Accuracy is measured by computing the Frobenius norm $\|XA - FX - GC\|_F$. The benchmarks were done in **Matlab 6** in **Pentium II** 400 MHz environment; they show that the proposed algorithm can achieve a better accuracy with a comparable speed in structured problems.

IV. 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 as long as the solution of small sized standard Sylvester equations do not required this reduction. This algorithm is well-suited for high-performance implementation using *LAPACK* and it seems to be accurate compared with similar ones; numerical stability properties have not been studied yet.

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, Second Edition.* SIAM, Philadelphia, PA, 1995.

[2] C. Bischof, B. Datta and A. Purkayastha, *A parallel algorithm for the Sylvester observer equation*, SIAM J. Sci. Comp., **17**, 686-698, 1996.

[3] D. Calvetti, B. Lewis and L. Reichel,*On the solution of large Sylvester-observer equation*, Num. Lin. Alg. Appl. **8** nr. 6-7, 435-452, 2001.

[4] J. Carvalho and B. Datta, *A new block algorithm for the Sylvester- observer equation arising in state-estimation.* Proc. IEEE International Conference on Decision and Control, Orlando, pp 3398-3403, 2001.
observer

[5] B. Datta and D. Sarkissian, *Block algorithms for state estimation and functional observers*, Proc. IEEE International Symposium on Computer-Aided Control System Design, Anchorage, pp. 19-23, 2000.

[6] B. Datta, *Numerical Methods for Linear Control Systems Design and Analysis*. Academic Press, to appear in 2002.

[7] K. Datta, *The matrix equation $AX - XB = R$ and its applications*. Lin. Alg. Appl, **109**, 91-105, 1988.

[8] J. Dongarra, I. Duff, D. Sorensen and H. Van der Vorst. *Numerical Linear Algebra for High-performance Computers*. SIAM Press, 1998.

[9] J. Doyle and G. Stein, *Robustness with Observers*. IEEE Trans. Aut. Contr., **AC-26**, pp.4-16, 1981.

[10] G.R. Duan, *Solutions of the equations* $AV + BW = VF$ *and their application to eigenstructure assignment in linear systems*, IEEETAC, **38**, no. 2, pp. 276-280, 1993.

[11] G. Golub, C. Van Loan, *Matrix Computations*, 3rd. ed., Johns Hopkins Univ. Press, 1996.

[12] G.Golub, S.Nash and C.Van Loan,*A Hessenberg-Schur method for the problem* $AX+XB = C$, IEEE Trans. Autom. Contr. **AC-24**, pp. 909-913, 1979.

[13] N.J. Highan, *The Test Matrix Toolbox for Matlab*. Numer. Anal. Rep. 276, Manchester Centre for Comput. Mathematics, Univ. of Manchester, UK, 1995.

[14] D. Luenberger, *Observing the state of a linear system*. IEEE Trans. Mil. Electr., **8**, 74-80, 1964.

[15] B. Shafai and S.P. Bhattacharyya, An algorithm for Pole assignment in high order multivariable systems, IEEETAC, vol. 33, no. 9, pp. 870-876, 1988.

[16] E.DeSouza , S.P. Bhattacharyya, *Controllability, observability and the solution of* $AX - XB = C$, Lin. Alg. Appl. **39**, 167-188, 1981.

[17] P. Van Dooren, *Reduced order observers: A new algorithm and proof*. Syst.Cont.Lett. **4**, 243-251, 1984.

[18] Y. Zhang and J. Wang, *Global exponential stability of recurent neural networks for synthesizing linear feedback control system via pde assignment*, IEEE TNN, vol. 13, pp. 633-644, 2002.