Krylov Subspace Methods for Large-Scale Matrix Problems in Control

Biswa Nath Datta

Department of Mathematical Sciences
Northern Illinois University
DeKalb, IL 60115
E-mail: *dattab@math.niu.edu*
URL: *math.niu.edu/∼dattab*

Abstract

This paper presents a brief but state-of-the-art survey of some of existing Krylov subspace methods for large-scale matrix problems in control. Based on the discussions and observations, some research problems are suggested.

1 Introduction

The design and analysis of linear control systems in state-space forms:

$$\left.\begin{aligned}\dot{x}(t) &= Ax(t) + Bu(t)\\ y(t) &= Cx(t)\end{aligned}\right\} \quad \textbf{(Continuous-time case)} \tag{1.0.1}$$

and

$$\left.\begin{aligned}x_{k+1} &= Ax_k + Bu_k\\ y_k &= Cx_k\end{aligned}\right\} \quad \textbf{(Discrete-time case)} \tag{1.0.2}$$

give rise to a wide variety of important linear algebra problems. These include **controllability, observability, and stability problems, Lyapunov, Sylvester, Sylvester-Observer and Riccati equations, eigenvalue and eigenstructure assignment and model reduction.**

In (1.0.1) and (1.0.2), the matrices $A \in \mathbb{R}^{n\times n}$, $B \in R^{n\times m}$ $(m \leq n)$ and $C \in \mathbb{R}^{r\times n}(r \leq n)$ are, respectively, called the **state, control and output matrices.** These systems are sometimes conveniently written in compact form as (A, B, C). If $m = 1$, then the system is called **single-input system** and similarly if $r = 1$, the system is called **single-output system**. If the system is both single-input and single-output, then it is called a single-input and single-output system and is denoted by **SISO**. The multi-input and multi-output **(MIMO)** systems are similarly defined. An SISO system is denoted by (A,b,c).

There now exist well-established numerically viable computational methods for solving these problems for dense matrices (see Patel et al. (1994) and Datta (2002)). Unfortunately, these methods are not suitable for solutions of many large practical problems such as those arising in the design of large sparse structures, power systems, etc. There are two main reasons for this. First, they destroy the sparsity, inherited in most large practical problems and second, they are $O(n^3)$ methods and thus, are computationally prohibitive for large problems.

On the other hand, there have been some fine recent developments in the area of large-scale matrix computations. A class of classical methods known as the **Krylov subspace methods** (Arnoldi (1951) and Lanczos (1950) have been found to be suitable for sparse matrix computations. The reason is that these methods can be implemented using matrix-vector multiplications only; therefore, the sparsity in the original problems can be preserved. The examples are the GMRES **(Generalized Minimal Residual)** and the QMR **(Quasi-Minimal Residual)** methods for linear systems problem; the **Arnoldi, Lanczos**, and the **Jacobi-Davidson** methods, and several variants of them such as the **restarted** and **block Arnoldi methods and band Lanczos method** for eigenvalue problems.

It is only natural to develop algorithms for large-scale control problems using these effective large-scale techniques of matrix computations. Some work to this effect has been done in the last few years.

In this paper, we will briefly review some of these methods. The earlier and recent surveys on this subject include the papers by Boley (1994), Van Dooren (1992), Datta (1989, 1998) and Boley, Datta(1996). In Section 2, we give a brief description of the basic Arnoldi and Lanczos methods to facilitate the understanding of how these methods are applied to solve large-scale control problems. We stress that *the descriptions of our Krylov subspace methods are basic.* For practically implementable versions of these methods and associated software, we refer the readers to the books by Bai et al. (2000), and Saad (1992, 1996). *In particular, the homepage, ETHOME of the book by Bai et al.* (2000) *contains valuable information of available software.* Our only goal of this paper is to show the readers how these modern iterative numerical methods can be gainfully employed to solve some of the large and sparse matrix problems arising in control. Unfortunately, for space limitation, we are also unable to give detailed control-theoretic description of each problem treated in this paper. We refer the readers to control theory books such as Chen (1984), or Kailath (1980) for this purpose. The upcoming book by the author (Datta (2003)) also contains some motivating control-theoretic backgrounds for these problems.

2 The Arnoldi and Block Arnoldi Methods

In this section, we summarize the essentials of the scalar Arnoldi and block Arnoldi methods.

2.1 The Scalar Arnoldi Method

Given an $n \times n$ matrix A, a vector v, and an integer $m \leq n$, the scalar Arnoldi method computes simultaneously a set of orthonormal vectors $\{v_1, ..., v_{m+1}\}$, and an $m \times m$ Hessenberg matrix H_m such that we have $V_m^T A V_m = H_m$ with $V_m = (v_1, v_2, ..., v_m)$. The vectors $\{v_1, ..., v_m\}$ form an orthonormal basis of the Krylov subspace $K_m(A, v_1) = \text{ span } \{v_1, Av_1, ..., A^{m-1}v_1\}$. Furthermore,

$$AV_m = V_m H_m + h_{m+1,m} v_{m+1} e_m^T, \tag{2.1.1}$$

where e_m is the last column of the $m \times m$ identity matrix.

The algorithm breaks down at step j, i.e., $\hat{v}_{j+1} = 0$, if and only if the degree of the minimal polynomial of v_1 is exactly j, that is, it is a combination of j eigenvectors.

2.2 The Block Arnoldi Method

The block Arnoldi method is a generalization of the scalar Arnoldi method. Starting with a block vector V_1 of norm unity, the block Arnoldi methods constructs a set of block vectors $\{V_1, ..., V_{m+1}\}$ such that if $U_m = (V_1, ..., V_m)$, then $U_m^T U_m = I_{mp \times mp}$, and $U_m^T A U_m$ is an upper block Hessenberg matrix $H_m = (H_{ij})$. Furthermore, $A\ U_m - U_m H_m = V_{m+1} H_{m+1,m} E_m^T$, where E_m is the last p columns of the $mp \times mp$ identity matrix. **The block Arnoldi algorithm is particularly suited for handling multivariable control problems.**

Algorithm 2.2.1 ***The Block Arnoldi Algorithm (Modified Gram-Schmidt Version)***

Let V be an $n \times p$ matrix.

Step 0. *Compute the $n \times p$ orthogonal matrix V_1 by finding the QR factorization of V : $V = V_1 R$ (Note that R is here $p \times p$). (Use column pivoting if V does not have full rank).*

Step 1. *For $k = 1, 2, ..., m$ do*
  Compute $\hat{V} = AV_k$.
  For $j = 1, 2, ..., k$ do
  $H_{j,k} = V_j^T \hat{V}$
  $\hat{V} = \hat{V} - V_j H_{j,k}$
  End
 Compute $H_{k+1,k}$ by finding the QR factorization of $\hat{V}$:
 $\hat{V} = V_{k+1} H_{k+1,k}$
 End

The block Arnoldi algorithm clearly breaks down if $H_{k+1,k}$ becomes zero for some k. **Such a breakdown has positive consequences in some applications.** (See Section 4.1.1).

2.3 The Lanczos and Block Lanczos Methods

For a nonsymmetric matrix A, the Lanczos the algorithm constructs, starting with two vectors v_1, and w_1, a **pair** of **biorthogonal bases** $\{v_1, ..., v_m\}$ and

$\{w_1, ..., w_m\}$ for the two Krylov subspaces: $K_m(A, v_1) = \text{ span}\{v_1, Av_1, ..., A^{m-1}v_1\}$ and $K_m(A^T, w_1) = \text{ span } \{w_1, A^T w_1, ..., (A^T)^{m-1}w_1\}$.

Algorithm 2.3.1 ***The Nonsymmetric Lanczos Algorithm***

Step 0. *Scale the vectors v and w to get the vectors v_1 and w_1 such that $w_1^T v_1 = 1$.*
Set $\beta_1 \equiv 0,\ \delta_1 \equiv 0, w_0 = v_0 \equiv 0$.

Step 1. *For $j = 1, 2, ..., m$ do*

$$\begin{aligned}
\alpha_j &= w_j^T A v_j \\
\hat{v}_{j+1} &= A v_j - \alpha_j v_j - \beta_j v_{j-1} \\
\hat{w}_{j+1} &= A^T w_j - \alpha_j w_j - \delta_j w_{j-1} \\
\delta_{j+1} &= \sqrt{|\hat{w}_{j+1}^T \hat{v}_{j+1}|} \\
\beta_{j+1} &= \hat{w}_{j+1}^T \hat{v}_{j+1} \Big/ \delta_{j+1} \\
w_{j+1} &= \hat{w}_{j+1} \Big/ \beta_{j+1} \\
v_{j+1} &= \hat{v}_{j+1} \Big/ \delta_{j+1}
\end{aligned}$$

End.

If the algorithm does not break down before completion of m steps, then, defining $V_m = (v_1, ..., v_m)$ and $W_m = (w_1, ..., w_m)$, we obtain (i) $W_m^T A V_m = T_m$, (ii) $AV_m = V_m T_m + \delta_{m+1} v_{m+1} e_m^T$, and $A^T W_m = W_m T_m^T + \beta_{m+1} w_{m+1} e_m^T$, where T_m is tridiagonal $(\alpha_1, ..., \alpha_m;\ \beta_2, ..., \beta_m;\ \delta_2, ..., \delta_m)$.

Breakdown of the Lanczos Method: If neither v_j nor w_j is zero, but $w_j^T v_j = 0$, then we have a breakdown (see Wilkinson (1965, pp. 389)). In that case, the **look-ahead Lanczos** idea has to be applied (See, Bai et al. (2000)).

2.3.1 The Block Lanczos Method.

Starting with $n \times p$ block vectors P_1 and Q_1 such that $P_1^T Q_1 = I$, the block Lanczos method generates right and left Lanczos block vectors $\{Q_j\}$ and $\{P_i\}$ of dimension $n \times p$, and a block-tridiagonal matrix $T_B = \text{ Tridiagonal } (T_1, ..., T_m;\ L_2, ..., L_m;\ M_2, ..., M_m)$ such that defining

$$P_{[m]} = (P_1, P_2, ..., P_m), \text{ and } Q_{[m]} = (Q_1, Q_2, ..., Q_m)$$

we have (i) $Q_{[m]}^T A Q_{[m]} = T_B$, (ii) $AQ_{[m]} = Q_{[m]} T_B + Q_{m+1} M_{m+1} E_m^T$, and (iii) $A^T P_{[m]} = P_{[m]} T_B^T + P_{m+1} L_{m+1}^T E_m^T$.

For details of the algorithm, see Bai et al. (2000) and Golub and Van Loan

(1996). *The block Lanczos method breaks down if $P_{j+1}^T Q_{j+1}$ is singular.* In such a situation, an *adaptively blocked Lanczos method* (Bai et al. (1999)) can be used to deal with the situation of breakdown. ABLE adaptively changes the block size and maintain the full or semi biothorgonality of the block Lanczos vectors.

3 Scopes of using the Krylov Subspace Methods in Control

Since the Krylov subspace methods (such as, the Arnoldi and Lanczos methods) are the projection methods onto K_m, it is only natural to use these methods as the projection techniques to solve large-scale control problems, as has been done in numerical linear algebra for matrix problems.

A **template** is then as follows: First, the original large control problem is projected onto an m-dimensional Krylov subspace by constructing a basis of the subspace. The projected smaller problem is then solved using a standard well-established technique. Finally, an approximate solution of the original problem is recovered form the solution of the projected problem. The solution of the projected problem is constructed such that either a **Galerkin property** is satisfied; that is, the residual is orthogonal to the associated Krylov subspace, or the norm of the residual error is minimized (**GMRES type**). These projected methods usually give cheaply computed residual error norms, which, in turn can be used as a stopping criteria in case the methods need to be restarted.

4 Arnoldi Methods For Lyapunov, Sylvester and Algebraic Riccati Equations

The **Lyapunov equations** $AX + XA^T + BB^T = 0$ **(Continuous-time),** and $AXA^T - X + BB^T = 0$ **(Discrete-time)** arise in a wide variety of control applications (e.g., **controllability, observability, stability and robust stability analyses, model reduction,** etc.). The standard **Schur-method** (Bartels and Stewart (1972)) based on the Schur decomposition of A, is not suitable for sparse problems. Below, we show the use of scalar Arnoldi to solve the single-input continuous-time and that of the block Arnoldi to solve the multi-input discrete-time problem.

The matrix A is assumed to be stable in each case; that is, in the continuous-time case, A is assumed to have all eigenvalue negative real parts and in the discrete case, A is assumed to have all its eigenvalues within the unit circle.

Algorithm 4.0.1 ***An Arnoldi-method for the Single-input Stable Lyapunov Equation***

Step 1. *Run m steps of the Arnoldi algorithm with $v_1 = \dfrac{b}{||b||_2} = \dfrac{b}{\beta}$. Obtain V_m and H_m.*

Step 2. *Solve the projected $m \times m$ Lyapunov matrix equation: $H_m G_m + G_m H_m^T + \beta^2 e_1 e_1^T = 0$, using the Schur-method (see* **Bartels and Stewart (1972)**) *or* ***Chapter 8*** *in Datta (2002)).*

Step 3. *Compute X_m, an approximation to X: $X_m = V_m G_m V_m^T$.*

Galerkin condition, Residual Error and Re-start: (i) It is shown (Saad (1990), Jaimoukho and Kasenally (1994)) that the residual $\mathrm{Res}(X_m) = AX_m + X_m A^T + bb^T$ satisfies: $V_m^T\ \mathrm{Res}(X_m) V_m = 0$ and (ii) the residual error-norm $||\mathrm{Res}(G_m)||_F = \sqrt{2}||h_{m+1,m} e_m^T G_m||_F$. Using this cheaply computed residual error-norm as a stopping criterion, Algorithm 4.0.1 can be restarted at every fixed number (say m_1) of iterations.

Algorithm 4.0.2 ***A Block Arnoldi Algorithm for Stable Discrete-time Lyapunov Equation***

Step 1. *Find the QR factorization of B to compute V_1:*

$$B = V_1 R.$$

Step 2. *Run m steps of the block Arnoldi algorithm to obtain H_m, U_m, and $H_{m+1,m}$ with V_1 as obtained in Step 1.*

Step 3. *Obtain an $mp \times mp$ matrix G_m by solving the projected discrete Lyapunov equation using the Schur-method:*

$$H_m G_m H_m^T + \begin{pmatrix} R \\ 0 \end{pmatrix} \begin{pmatrix} R^T & 0 \end{pmatrix} = G_m.$$

Step 4. *Compute the approximate solution $X_m = U_m G_m U_m^T$.*

Galerkin Condition, Residual Error Norm and Restart:

1. The residual $Res(X_m) = AX_m A^T - X_m + BB^T$ satisfies the **Galerkin property**: $U_m^T Res(X_m) U_m = 0$.

   Furthermore, the **residual error norm** for the solution of the projected problem is given by (Jaimoukha and Kasenally (1994)):

$$||Res(G_m)||_F = \left\| H_{m+1,m} E_m^T G_m \left(\sqrt{2} H_m^T \quad E_m H_{m+1,m}^T \right) \right\|_F .$$

2. The error bound $||X - X_m||_2$ converges to zero as m increases (Boley (1994)).
3. *As in the continuous-time case, the cheaply computed residual can be used to restart the process if necessary.*

4.1 Arnoldi Methods for Sylvester Equation

Let A, B, and C be the matrices of order n. (**Note that the matrix B here is not the usual control matrix.**) The Sylvester equation: $AX - XB = C$ can be written as the following linear systems of equations: $(I \otimes A - B^T \otimes I)x = c$, where $\otimes$ denotes the Kronecker product, and x and c are vectors with n^2 components. Hu and Reichel (1992) have proposed a method to solve this system requiring a considerable reduced amount of storage space. Their idea is to replace the Krylov subspace $K_m(I \otimes A - B^T \otimes I,\ r_0)$ with a subspace of the form $K_m(B^T, g) \otimes K_m(A, f)$ for certain vectors f and g. The vectors f and g are chosen so that the initial residual vector $r_0 = b - Ax_0$, where x_0 is the initial approximate solution, lies in the Krylov subspace $K_m(B^T, g) \otimes K_m(A, f)$.

Algorithm 4.1.1 ***A Restarted Arnoldi Algorithm for the Sylvester Equation $AX - XB = C$ (Galerkin type).***

Step 1. *Choose x_0 and compute $r_0 \equiv c - (I \otimes A - B^T \otimes I)x_0$.*

Step 2. *If $||r_0||_2 \leq \epsilon$, then compute the approximate solution matrix X_0 of the equation $AX - XB = C$ from the entries of x_0.*

Step 3. *Choose f and g using the following scheme:*
Let R_0 be defined by: $e_j^T R_0 e_k = e_{j+n(k-1)}^T r_0, 1 \leq j, k \leq n$.
Then, if $||R_0||_1 \geq ||R_0||_\infty$, determine $g = \dfrac{R_0^T f}{||f||^2}$, taking f as the column of R_0 of the largest norm. Else, determine $f = \dfrac{R_0 g}{||g||^2}$, taking g as a row of R_0 of the largest norm.
Using the Arnoldi algorithm, compute the orthonormal bases of $K_{m+1}(A, f)$ and $K_{m+1}(B^T, g)$; that is, obtain $H_A, H_B, \tilde{H}_A, \tilde{H}_B, V_m, V_{m+1}, W_m, W_{m+1}$.

Step 4. *Compute $\tilde{r}_0 = (W_m \otimes V_m)^T r_o$.*

Step 5. *Determine* Q_A *and* R_A *from* H_A*, and* Q_B *and* R_B *from* H_B *by Schur factorizations. That is, find* $Q_A, R_A; Q_B, R_B$ *such that* $H_A = Q_A R_A Q_A^*$ *and* $H_B = Q_B R_B Q_B^*$*. Compute* $r_0' = (Q_B \otimes Q_A^*)\tilde{r}_0$.

Step 6. *Solve the triangular or the quasi-triangular system:* $(I \otimes R_A - R_B \otimes I)y_0' = r_0'$ *and compute* $y_0 = (Q_B \otimes Q_A)y_0'$.

Step 7. *Compute the correction vector:* $z_0 = (W_m \otimes V_m)y_0$ *and update the solution:* $x_0 \equiv x_0 + z_0$.

Step 8. *Compute the updated residual vector:* $r_0 \equiv r_0 - (W_m \otimes V_{m+1}\tilde{H}_A)y_0 + (W_{m+1}\tilde{H}_B \otimes V_m)y_0$ *and go to Step 2.*

A **breakdown of the algorithm** occurs **when the matrix of the linear system in Step 6 becomes singular.** In this case, one can either reduce m or restart the algorithm with different f and g, e.g., random vectors. The same action should be taken when $\dim K_m(A, f) < m$ or $\dim K_m(B^T, g) < m$.

4.1.1 Block Arnoldi Methods for Sylvester Equation

While the Hu-Reichel algorithm is a projection algorithm on the linear algebraic system associated with the Sylvester equation, projection algorithms on the actual Sylvester equation have recently been developed (El Guennouni, et al. (2002), Robbe and Sadkane (2002)). Furthermore, the Hu-Reichel algorithm has been extended to the block form by Simoncini (1996). Let $A \in \mathbb{R}^{n\times n}$, $B \in \mathbb{R}^{p\times p}$.

Algorithm 4.1.2 ***Block Arnoldi Methods for the Sylvester Equation*** $\boldsymbol{AX + XB = C}$

Step 1. *Choose* X_0 *and compute the residual matrix* $R_0 = C - (AX_0 - X_0B)$*. Assume that rank*$(R_0) = q$.

Step 2. *Obtain an upper triangular matrix* Λ_1 *by computing the full rank QR factorization of* R_0*:* $R_0 = V_1\Lambda_1$ *and run* m *steps of the block Arnoldi algorithm with* V_1 *to obtain* U_m, H_m*, and* $\tilde{H}_m$.

Step 3. *(**Galerkin-type**): Compute the approximate solution:* $X_m^G = X_0 + U_m Z_m$, *obtaining* Z_m *by solving the Sylvester equation using the Hessenberg-Schur method (Golub et al. (1979)):*

$$H_m Z_m - Z_m B = \Lambda, \ \textit{where } \Lambda = \begin{pmatrix} \Lambda_1 \\ 0 \end{pmatrix} \in \mathbb{R}^{mq \times p}.$$

*(**GMRES-type**): Compute the approximate solution:* $X_m^{GM} = X_0 + U_m Z_m$, *obtaining* Z_m *by solving the minimization problem:*

$$\min_{Z \in R^{mq \times p}} \|\bar{\Lambda} - \bar{S}_m(Z)\|_F, \ \textit{where } \bar{\Lambda} = \begin{pmatrix} \Lambda_1 \\ 0 \end{pmatrix} \in \mathbb{R}^{(m+1)q \times p}$$

$$\textit{and } \bar{S}_m(Z) = \tilde{H}_m Z - \begin{pmatrix} Z \\ 0 \end{pmatrix} B.$$

Residuals and Restart: It can be shown (Robbé and Sadkane (2002)) that the residuals $R_m^{GM} = S(X_m^{GM}) - C$ and $R_m^G = S(X_m^G) - C$ satisfy:

$$\|R_m^{GM}\|_F = \|\bar{\Lambda} - \bar{S}_m(Z_m)\|_F, \ \text{and } \|R_m^G\|_F = \|H_{m+1,m} Z_m^L\|_F,$$

where Z_m^L is the last $q \times p$ block of Z_m. Using these easily computed residuals, the method should be periodically restarted with $X_0 = X_m^G$ or $X_0 = X_m^{GM}$, where $X_m^G | X_m^{GM}$ is the last computed approximate solution with Galerkin/GMRES method.

Convergence Analysis (Robbé and Sadkane (2002)):

(1) The GMRES algorithm converges if the field of values of A and B are disjoint. If the Galerkin algorithm converges, then the GMRES algorithm also converges. However, if GMRES stagnates (i.e. $\|R(X_m^{GM})\|_F = \|R_0\|_F$), then the Galerkin algorithm fails.
Note: It is assumed that R_0 and the parameter m are the same in both these algorithms.

(2) (**Breakdown**). If the block Arnoldi algorithm breaks down at iteration m; that is, if $H_{m+1,m} = 0$, then the approximate solutions computed by GMRES and the Galerkin algorithm are exact; that is, $X_m^G = X_m^{GM} = X$.

4.1.2 Arnoldi-Method for Sylvester-Observer Equation (single-output case)

The Sylvester-observer equation $A^T X - XH = C^T G$ arises in the construction of Luenberger observer (Chen (1984), Datta (1994)). For a full-rank solution X, it is necessary that (A, C) is observable and (H^T, G) is controllable. If H is an upper Hessenberg matrix (as in the case of the scalar Arnoldi), then in the single-output case g can be chosen to be $e_n = (0, 0, ..., 0, 1)$ and the Sylvester-observer equation in this case reduces to $A^T X - XH = (0, 0, ..., 0, c^T)$. An Arnoldi method was developed by Datta and Saad (1991) to solve this equation by observing the striking resemblance of this equation with the Arnoldi equation: $AV_m - V_m H_m = (0, 0, ..., 0, h_{m+1,m} v_{m+1})$. Naturally, the Arnoldi vector v_1 should be chosen so that the last vector v_{m+1} becomes the vector c, given a priori. This is done by observing that, apart from a multiplicative scalar, the polynomial $p_m(x)$ such that $v_{m+1} = p_m(A)v_1$, is the characteristic polynomial of H_m. The matrix H_m is constructed to have a pre-assigned spectrum $\{\mu_1, ..., \mu_m\}$ for which an eigenvalue assignment algorithm (e.g. Datta (1987)) is invoked at the end of $(m-1)$ steps of the Arnoldi algorithm with the chosen vector v_1.

Algorithm 4.1.3 ***An Arnoldi Algorithm for Single-output Sylvester-observer Equation***

Step 1. *Solve the linear system:* $q(A^T)x = c^T$, *and compute* $v_1 = x/||x||$; *where* $q(t) = (t-\mu_1)(t-\mu_2)...(t-\mu_m)$.

Step 2. *Run* $m-1$ *steps of the Arnoldi method on* A^T *with* v_1 *as the initial vector to generate* V_m *and the first* $m-1$ *columns of* H_m. *Let* $\tilde{H}_{m-1}$ *denote the matrix of the first* $m-1$ *columns of* H_m.

Step 3. *Find a column vector* y *such that* $\Omega([\tilde{H}_{m-1}, y]) = \Omega(H_m) = \{\mu_1, ..., \mu_m\}$, *where* $\Omega(K)$ *denotes the spectrum of the matrix* K.

Step 4. *Compute* $\alpha = (c')^T c'/||c||^2$, *where* c' *is the last column of* $A^T V_m - V_m H_m$.

Step 5. *Set* $X_m = (1/\alpha)V_m$.

Solving the Equation $q(A^T)x = c^T$ using the Partial Fraction Approach: A partial-fraction approach suggested in Datta and Saad (1991) to solve the above polynomial system of equations consists in decomposing the system into m linearly independent systems: $(A^T - \mu_i I)x_i = c^T, i = 1, ..., m$ and then obtaining the solution x as the linear combination: $x = \sum_{i=1}^{n} \frac{1}{q'(\mu_i)} x_i$, where $q'(\mu_j) = \Pi_{i=1,...,m, i \neq j}(\mu_i - \mu_j)$. Each of these systems can be solved by applying k steps of the Arnoldi method, constructing an orthonomial basis V_k of the span $\{c, A, c, ..., A^{k-1}c\}$ and then solving k independent small $m \times m$ Hessenberg linear systems. The bulk of the work is in constructing V_k, and this is done only once. A detailed stability (numerical) property of the approach studied in Calvetti, Gallopoulas, and Reichel (1995), Calvetti and Reichel (1997) and Calvetti, Lewis and Reichel (2001), shows that the performance of the scheme can be improved by choosing $\mu_i s$ as the equidistant points on a circle or on the zeros of a certain Chebyshev polynomial.

Remarks:
(i) Observe that the solution obtained by this algorithm has the nice additional property of being **orthonormal**. (ii) A full Arnoldi-type of method $(m = n)$ for the construction of an orthogonal solution to the multi-output Sylvester-observer equation has been developed by Datta and Hetti (1997). Also, there now exists an SVD-based algorithm (Datta and Sarkissian (2000)) for solving the multi-output Sylvester-observer equation, which might be suitable for large-scale computing.

4.2 Arnoldi-Method for Continuous-time Algebraic Riccati Equation

The continuous-time algebraic Riccati equation, (CARE): $XA + A^T X - XBB^T X + LL^T = 0$ arises in the **optimal control, Kalman filtering, LQG design**, etc. (see, Datta (2002)). *The widely-used Schur method or similar methods based on matrix decompositions are not practical for large problems.* An idea to solve the CARE using the block Arnoldi method due to Jaimoukha and Kasenally (1994) is as follows:

Algorithm 4.2.1 ***An Arnoldi Algorithm for Continuous-time Algebraic Riccati Equation (Galerkin-type).***

Step 1. *Compute $U_m, H_m, H_{m+1,m}$ by running m steps of the block Arnoldi method starting with V_1 given by: $L = V_1 R$ (**QR factorization of B**). Define B_m by $U_m^T B = B_m$ and L_m by $U_m L_m = L$.*

Step 2. *Solve the projected equation for G_m:*

$$G_m H_m + H_m^T G_m - G_m B_m B_m^T G_m + L_m L_m^T = 0$$

Step 3. *Compute approximation X_m of X : $X_m = U_m G_m U_m^T$*

Galerkin Condition and Restart:

i. Provided that the associated Hamilton matrix does not have a purely imaginary eigenvalue, **the residual norm satisfies $\mathbf{Res}(X_m)$ the Galerkin property: $V_m^T \mathbf{Res}(X_m) V_m = 0$.**
ii. Algorithm 4.2.1 can be **restarted** by using the cheaply-computed residual error norm: $||Res(G_m)||_F = \sqrt{2} \parallel H_{m+1,m} E_m^T G_m \parallel_F$, as a stopping criterion.

5 Arnoldi Method for Partial Eigenvalue Assignment

Given an $n \times n$ large and sparse matrix A whose spectrum $\Omega(A) = \{\lambda_1, ..., \lambda_p; \lambda_{p+1}, ..., \lambda_n)$, an $n \times m$ control matrix B and a set of self-conjugate scalars $\{\mu_1, ..., \mu_p\}$, the partial eigenvalue assignment problem is the one of finding a feedback matrix K such that $\Omega(A - Bk) = \{\mu_1, ..., \mu_p; \lambda_{p+1}, ..., \lambda_n)$. The problem naturally arises in feedback stabilization of large systems such as large space structures, power plants.

A projection method due to Saad (1988), which can be implemented using the Arnoldi method is based on computing an orthonormal basis for the left invariant subspace associated with the k eigenvalues that are to be reassigned.

Algorithm 5.0.1 ***A Projection Algorithm for Partial Pole-Placement***

Step 1. *Compute the partial Schur decomposition: $A^T Q = QR$ associated with the eigenvalues $\lambda_1, \lambda_2, \cdots, \lambda_k$.*

Step 2. *Compute $S_0 = Q^T B$ and solve the projected $k \times k$ eigenvalue assignment problem. That is, find a matrix G such that $\Omega(R^T - S_0 G^T) = \{\mu_1, \mu_2, \cdots, \mu_k\}$, using a* ***standard multi-input eigenvalue assignment method*** *(See Datta* (2002) *or Patel, et al.* (1994)*).*

Step 3. *Form the feedback matrix: $K = (QG)^T$*

Remarks:

1. In Datta and Sarkissian (2002), new theoretical and computational results for both partial eigenvalue and eigenstructure assignments have been given. **The theoretical results include conditions** for **existence** and **uniqueness** of **solutions of these problems**. The algorithms, although not Krylov subspace-based, are practical for large problems.

2. An Implicitly Restarted Arnoldi (IRA) method for partial eigenvalue assignment problem has been proposed in Calvetti, Lewis, and Reichel (2001).

6 Lanczos and Arnoldi Methods for Model Reduction

Model reduction is a procedure to obtain a reduced-order model (ROM) from the original large and sparse model in such a way that the transfer functions match closely. Thus, if $G(s) = \sum_{i=0}^{\infty} \frac{G_i}{s^{i+1}}$ and $\hat{G}(s) = \sum_{i=1}^{\infty} \frac{\hat{G}_i}{s^{i+1}}$ are the transfer functions of the original and reduced-order model, respectively, then the procedure should yield $\hat{G}_i = G_i$ for $i = 1, ..., k$ for some given k. It is easy to see that if (A, B, C) is the state-space realization of the original model, then

$$G(s) = C(sI - A)^{-1}B = \sum_{i=0}^{\infty} \frac{CA^iB}{s^{i+1}}$$

for $|s| > \rho(A)$. Similarly for the ROM. The quantities $CA^iB = H_i$ are called **Markov-parameters.** Thus, a ROM $(\hat{A}, \hat{B}, \hat{C})$ should be constructed such that $H_i = \hat{H}_i, i = 1, 2, ..., k$ for some k. The standard methods for model reduction, based on **internal balancing** and **Schur decomposition** of A (see, Datta (2002)), are not practical for large-scale models.

Intuitively, it is clear that a Krylov subspace method will be a perfect candidate for constructing a reduced order model. Indeed, several Krylov subspace methods for model reduction have been developed in recent years. These include the Padé via Lanczos (PVL) approach (Van Dooren (1995), Feldman and Freund (1995), Freund (1999), etc.), the interpolation approach (Grimme (1994), Gallivan, Grimme and Van Dooren (1994, 1996)) based on the rational Krylov method of Ruhe, implicitly restarted Lanczos method (Grimme, Sorensen and Van Dooren (1996)) and Arnoldi and implicitly restarted dual Arnoldi methods (Jaimoukha and Kasenally (1995, 1997), Su and Craig (1991), Papakos (2001), Papakos and Jaimoukha (2001, 2002)). *The PVL technique has been proven to be effective in circuit simulation and the multipoint rational interpolation approach has been successful in moment matching of the transfer function at selected frequencies.* The machinery needed to describe these techniques has not been developed here and, therefore, we have to skip the descriptions of these techniques. For state-of-the-art survey on this topic, see Antoulas (2002) and Van Dooren (2000).

We will describe here only a basic Lanczos and an Arnoldi method for model reduction in the SISO case and just mention the existence of the block Lanczos and band Lanczos methods in the MIMO case.

6.1 Lanczos Methods for Model Reduction

Algorithm 6.1.1 ***A Lanczos Algorithm for SISO Model Reduction***

Step 0. *Scale the vectors b and c to obtain the vectors v_1 and w_1 such that $w_1^T v_1 = 1$.*

Step 1. *Run k steps of the Lanczos algorithm (**Algorithm 2.3.1**) to generate the matrices W_k and V_k and then compute $A_k = W_k^T A V_k, \quad b_k = W_k^T b, \quad c_k = c V_k$.*

Step 2. *Form the reduced-order model (A_k, b_k, c_k).*

It can be shown that the reduced-order model defined by $(\hat{A}_k, \hat{b}_k, \hat{c}_k)$ preserves the first $2k$ Markov parameters of the original system. (See Gragg (1974), Gragg and Lindquist (1983)). That is, $cA^{i-1}b = c_k A_k^{i-1} b_k, i = 1_2, 2, \cdots, 2k$.

Numerical Disadvantages and Possible Cures

There are several numerical difficulties with the above algorithm: first, there can be serious "breakdowns" in the Lanczos process due to the ill-conditioning of the submatrices in the system's Hankel matrix; second, the steady state error can be large; third, the stability of the reduced-order model is not guaranteed even though the original model is stable. An **implicit restated Lanczos scheme** due to Grimme, et.al. (1996), to stabilize the reduced-order model is as follows: Suppose that the matrix A_k is not stable and assume that there are q unstable modes: $\mu_1, \cdots, \mu_q$. Then the idea is to restart Algorithm 6.1.1 with the new starting vectors $\bar{v}_1 = \bar{p}_\vartheta (A - \mu_q I) \cdots (A - \mu_1 I) v_1$, and $\bar{w}_1 = \bar{p}_w (A^T - \mu_q I) \cdots (A^T - \mu_1 I) w_1$, where $\bar{p}_\vartheta$ and $\bar{p}_w$ are certain scalars. The scheme is implemented implicitly using a technique similar to the one proposed in Sorensen (1992). There also exist relations between the modified Markov parameters of the original system and the above restarted Lanczos model (see Grimme et al. (1996)).

6.2 Block Lanczos and Band Lanczos Methods for MIMO Model Reduction

In the MIMO case, when $m = r$, the **block Lanczos** method can be used. Specifically, the following result (see Boley (1994)) can be proved.

Theorem 6.2.1 *Let j steps of the block Lanczos method be applied to the MIMO system (A, B, C), starting with block vectors generated from the QR decompositions of B and C, obtaining the matrices $P_{[j]}$ and $Q_{[j]}$. Define $\hat{A} = Q_{[j]}^T A P_{[j]}$, $\hat{B} = Q_{[j]}^T B, \hat{C} = C P_{[j]}$. Then the reduced-order model $(\hat{A}, \hat{B}, \hat{C})$ has the following properties: $\hat{C}\hat{A}^i\hat{B} = CA^iB$ for $i = 0, 1, ..., 2(j-1)$.*

The band Lanczos method is an extension of the standard nonsymmetric Lanczos method for single vectors to blocks of starting vectors of different sizes. This method is thus ideal for the MIMO case when $m \neq r$. For space limitations, the detailed description of the algorithm can not be given here. For description of the algorithm, we refer the readers to the paper by Aliga et al. (2000). For application of the band Lanczos algorithm to the MIMO model reduction, see Freund (1999) and the paper by Freund in Bai et al. (2000, pp. 205-216).

6.3 An Arnoldi Method for SISO Model Reduction

The idea is to use the Arnoldi method simultaneously on (A, b) and (A^T, c^T) and then combine the results to obtain reduced-order models. The reduced-order models have been shown to satisfy the Galerkin conditions (Jaimoukha and Kasenally (1997)).

Algorithm 6.3.1 ***An Arnoldi Algorithm for SISO Model Reduction***

Step 1. *Perform m steps of the Arnoldi method with (A, b) to obtain $H_m, \tilde{H}_m, V_m, \tilde{V}_m$ and l_m, with $v_1 = b/||b||_2$. ($\tilde{V}_m = v_{m+1}$, $\tilde{H}_m = h_{m+1,m}e_m^T$ and $l_m = ||b||_2 e_1$).*

Step 2. *Perform m steps of the Arnoldi method with (A^T, c^T) to produce $G_m, \tilde{G}_m, W_m, \tilde{W}_m$ and k_m, with $w_1 = \dfrac{c^T}{||c||_2}$, $((\tilde{G}_m)^T = g_{m,m+1}e_m^T, \tilde{W}_m = w_{m+1}$ and $k_m = ||c||_2 e_1^T)$.*

Step 3. *Form* $T_m = W_m^T V_m$, $\hat{H}_m = T_m^{-1} W_m^T A V_m = H_m + T_m^{-1} W_m^T \tilde{V}_m \tilde{H}_m$ *and* $\hat{G}_m = W_m^T A V_m T_m^{-1} = G_m + \tilde{G}_m W_m^T V_m T_m^{-1}$

Step 4: *Form the reduced-order model* $(\hat{H}_m, l_m, k_m T_m)$ *or* $(\hat{G}_m, T_m l_m, k_m)$. *(Note that* $l_m = e_1 \|b\|_2$ *and* $k_m = \|c\|_2 e_1^T$*).*

Galerkin Conditions and Residual Errors: Let $h_m(s) = (sI - \hat{H}_m)^{-1} l_m$ and $g_m(s) = k_m(sI - \hat{G}_m)^{-1}$. Then the Galerkin conditions $W_m^T((sI - A)V_m h_m(s) - b) = 0$, and $(g_m(s) W_m^T (sI - A) - c)V_m = 0, \forall s$ are satisfied.

Remarks:

(i) *Jaimoukha and Kasenally (1997) have described a restarted Arnoldi framework which may be employed to make the reduced-order models stable and to remove redundant modes in the models.* For space limitation, we skip the description of this implicit method here.

(ii) Antoulas, Sorensen and Gugercin (2001), have recently proposed a restarted Arnoldi method, closely related to the one described above, based on the concept or the **Cross Grammian**. For space limitation, we skip the description here and refer the readers to the above paper.

7 Conclusion and Some Research Problems

We have given a very brief overview of some of the existing Krylov subspace methods in control. The research in this area is still in its infancy and clearly much, both for problems discussed here and those not included in our survey (e.g., large-scale **frequency response computations, model identifications**, and others), remains to be done. Based on our discussions and observations, we propose some research problems below for further investigations, which we hope, will be of interests and motivating to both computational mathematicians and control and systems researchers for further research in this area. We also feel that *there is a need for developing a high-quality software package for large-scale computing in control for use of researchers and practicing engineers.* Of course, any such project can successfully be implemented only with the combined expertise between control and systems theorists, computational mathematicians and practicing control and software engineers.

Proposed Research Problems

(1) Develop a block-Arnoldi type algorithm to solve the multi-output Sylvester-observer equation $AX - XB = GC$, analogous to single-output algorithm (Algorithm 4.1.3.)
(2) Develop a block Arnoldi algorithm for the discrete-time Algebraic Riccati equation (DARE): $A^TXA - X + Q - A^TXB(R + B^TXB)^{-1}B^TXA = 0$, analogous to Algorithm 4.2.1 in the continuous-time case.
(3) Develop an Arnoldi algorithm (scalar or block) for the generalized Sylvester equation: $AXB - X = C$.
(4) Develop a block Arnoldi-algorithm for MIMO model reduction that preserves stability of the original system.

REFERENCES

(1) J.I. Aliga, D.L. Boley, R.W. Freund, and V. Hernández, A Lanczos-type method for multiple starting vectors, Math. Comp., vol. 69, pp. 1577-1601, 2000.
(2) A.C. Antoulas, D.C. Sorensen and S. Gugercin, A Survey of model reduction methods for large-scale systems, *Contemporary Mathematics* (V. Olshevsky, Editor), AMS, Providence, vol. 280, pp. 193-219, 2001.
(3) A.C. Antoulas, *Lectures on Approximations of Large-Scale Dynamical Systems,* SIAM, Philadelphia, 2002 (To appear).
(4) W.E. Arnoldi, The principle of minimized iterations in the solution of the matrix eigenvalue problem, *Quart. Appl. Math.,* vol. 9, pp. 17-29, 1951.
(5) Z. Bai, D. Day and Q. Ye, ABLE: An adaptive block Lanczos method for nonhermitian eigenvalue problem, *SIAM J. Matrix Anal*, pp. 1060-1082, 1999.
(6) Z. Bai, J. Demmel, J. Dongarra, A. Ruhe, and H. Van der Vorst, *Templates for the Solution of Algebraic Eigenvalue Problems, A Practical Guide*, SIAM, Philadelphia, 2000.
(7) R.H. Bartels and G.W. Stewart, Solution of the equation $AX + XB = C$, Comm. ACM 15, pp.820-826, 1972.
(8) D.L. Boley, Krylov space methods on state-space control models, *Circ. Syst. & Signal Proc.*, vol. 13, pp. 733–758, 1994.
(9) D. Boley and B.N. Datta, Numerical methods for linear control systems, in *Systems and Control in the Twenty-First Century*, Edited by C. Byrnes, B. Datta, D. Gilliam, and C. Martin, Boston, Birkhauser, pp. 51-74, 1996.
(10) D. Boley and G. Golub, The nonsymmetric Lanczos algorithm and controllability, *Systems Control Lett.*, vol. 16, pp. 97-105, 1991.
(11) D. Calvetti and L. Reichel, Numerical aspects of solution methods for large Sylvester-like observer-equations, *Proc. IEEE Conference on Decision and Control*, pp. 4389–4397, 1997.
(12) D. Calvetti, E. Gallopoulas, and L. Reichel, Incomplete partial fractions for parallel evaluations of rational matrix functions, *J. Comp. Appl. Math.*, vol. 59, pp. 349-380, 1995.

(13) D. Calvetti, B. Lewis and L. Reichel, On the solution of large Sylvester-observer equation, *Num. Lin. Alg. Appl.*, vol. 8, pp. 435–451, 2001.
(14) D. Calvetti, B. Lewis, and L. Reichel, Partial eigenvalue assignment for large linear control systems, *Contemporary Mathematics*, (V. Olshevsky, Editor), American Mathematical Society, Providence, vol. 28, pp. 24-254, 2001.
(15) C.T. Chen, *Linear Systems Theory*, CBS College Publishing, New York, 1984.
(16) B.N. Datta, An algorithm to assign eigenvalues in a Hesseberg matrix: single-input case, *IEEE Trans. Autom. Control*, AC-32, pp. 414-417, 1987.
(17) B.N. Datta, Parallel and large-scale matrix computations in control: some ideas, *Lin. Alg. Appl.*, vol. 121, pp. 243-264, 1989.
(18) B.N. Datta, *Numerical Methods for Linear Control Systems Design and Analysis*, Elsevier, 2003.
(19) B.N. Datta, Krylov-subspace methods for control: An overview, *Proc. IEEE Conference on Decision and Control*, 1997.
(20) B.N. Datta and C. Hetti, Generalized Arnoldi methods for the Sylvester-observer matrix equation and the multi-input eigenvalue assignment problems, *Proc. IEEE Conference on Decision and Control*, pp. 4379-4383, 1997.
(21) B.N. Datta, Large-scale matrix computations in control, Applied Numerical Mathematics, vol. 30, pp. 1-11, 1998.
(22) B.N. Datta and Y. Saad, Arnoldi methods for large Sylvester-like observer matrix equations, and an associated algorithm for partial spectrum assignment, *Lin. Alg. Appl.*, vol. 154-156, pp. 225-244, 1991.
(23) B.N. Datta and D. Sarkissian, Partial eigenvalue assignment: Existence, uniqueness, and numerical solutions, *Proc. Mathematical Theory of Networks and Systems*, (MTNS' 2002), August, 2002.
(24) B.N. Datta and D. Sarkissian, Block algorithms for state estimation and functional observers, *Proc. Int. Symposium Computer-Aided Control Systems Design*, Alaska, 2000.
(25) A. El Guennouni, K. Jbilou and J. Riquet, Block Krylov subspace methods for solving large Sylvester equations, preprint, LMPA, No. 132, Université du Littoral, 2001 (To appear in *Num. Algorithms*), 2002.
(26) P. Feldmann and R.W. Freund, Efficient linear circuit analysis by Padé approximation via the Lanczos process. *IEEE Trans. Computer-Aided Design*, vol. 14, pp. 639-649, 1995.
(27) L. Fortuna, G. Nunnari, and A. Gallo, *Model Order Reduction Techniques with Applications in Electric Engineering*, Springer-Verlag, London, UK, 1992.
(28) R.W. Freund, Reduced-order modeling techniques based on Krylov subspace methods and their uses in circuit simulation, *Applied and Computational Control, Signals, and Circuits,* Vol. 1 (B.N. Datta, ed.), Birkhauser, Boston, pp. 435-498, 1999.
(29) K. Gallivan, E.J. Grimme, and P. Van Dooren, Asymptotic waveform

evaluation via a restarted Lanczos method, Appl. Math. Lett., vol. 7, pp. 75-80, 1994.

(30) K. Gallivan, E.J. Grimme, and P. Van Dooren, A rational Lanczos algorithm for model reduction, *Numer. Algorithm*, vol. 12, pp. 33-63, 1996.

(31) G. Golub and C. Van Loan, *Matrix Computations*, 3rd Edition, Johns Hopkins University, Baltimore, 1996.

(32) G. Golub, S. Nash, and C. Van Loan , A Hessenberg-Schur method for the problem $AX + XB = C$, *IEEE Trans. Automat. Control*, vol. AC-24, 909-913, 1979.

(33) W.B. Gragg, Matrix interpolations and applications of the continued fraction algorithm, *Rocky Mountain J. Math.*, vol. 4, pp. 213-225, 1974.

(34) W.B. Gragg and A. Lindquist, On the partial realization problem, *Lin. Alg. Appl.*, vol. 50, pp. 277-319, 1983.

(35) E.J. Grimme, D.C. Sorensen and P. Van Dooren, Model reduction of state space systems via an implicitly restarted Lanczos method, *Num. Alg.*, vol. 1, pp. 1-32, 1996.

(36) E. Grimme, *Krylov Projection Methods for Model Reduction*, Ph.D. Thesis, University of Illinois at Urbana-Champaign, Urbana, Illinois, 1994.

(37) D.Y. Hu and L. Reichel, Krylov subspace methods for the Sylvester equation, *Lin. Alg. Appl. and Appl.*, vol. 172, pp. 283-313, 1992.

(38) I.M. Jaimoukha, A general minimal residual Krylov subspace method for large-scale model reduction, *IEEE Trans. Automat. Control,* vol. 42, pp. 1422-1427, 1997.

(39) I.M. Jaimoukha and E.M. Kasenally, Krylov subspace methods for solving large Lyapunov equations, *SIAM J. Num. Anal.*, vol. 31, pp. 227-251, 1994.

(40) I.M. Jaimoukha and E.M. Kasenally, Implicitly restarted Krylov Subspace methods for stable partial realizations, *SIAM J. Matrix Anal. Appl.*, vol. 18, No. 3, pp. 633–652, 1997.

(41) I.M. Jaimoukha and E.M. Kasenally, Oblique projection methods for large scale model reduction, *SIAM J. Matrix Anal. Appl.*, vol. 16, pp. 602–627, 1995.

(42) T. Kailath, *Linear Systems*, Prentice Hall, Englewood Cliffs, N.J., 1980.

(43) C. Lanczos, An iteration method for the solution of the eigenvalue problem of linear differential and integral operators, *J. Res. Nat. Bur. Standards*, vol. 45, pp. 255-282, 1950.

(44) V. Papakos and I.M. Jaimoukha, Implicitly restarted Lanczos algorithm for model reduction, *Proc. 40th IEEE Conference and Decision and Control*, pp. 3671-3672, 2001.

(45) V. Papakos, *An implicitly restarted nonsymmetric Lanczos algorithm for model reduction*, MPhil to Ph.D. Transfer Report, Imperial College, London, U.K., 2001.

(46) V. Papakos and I. Jaimoukha, Model reduction via an LFT-based explicitly restarted Lanczos algorithm, *Proc. Mathematical Theory of Networks and Systems* (MTNS' 2002), 2002.

(47) R.V. Patel, A.J. Laub, P. M. Van Dooren (edited), *Numerical Linear Algebra Techniques for Systems and Control*, Reprint Book, IEEE press, Piscataway, N.J., 1994.

(48) M. Robbé and M. Sadkane, A convergence analysis of GMRES and FOM methods for Sylvester equations, To appear in *Num. Algorithms*, 2002.

(49) Y. Saad, Projection and deflation methods for partial pole assignment in linear state feedback. *IEEE Trans. Auto Control*, vol. 33, pp. 290-297, 1988.

(50) Y. Saad, Numerical Solutions of Large Lyapunov Equations., *Signal Processing, Scattering, Operator Theory, and Numerical Methods,* edited by M.A. Kaashoek, J. H. Van Schuppen and A. C. Ran, Birkhauser, pp. 503-511, 1990.

(51) Y. Saad, *Iterative methods for sparse linear systems,* PWS, Boston, 1996.

(52) Y. Saad, *Numerical Method for Large Eigenvalue Problems; Theory and Algorithms.* John Wiley, New York, 1992.

(53) V. Simoncini, On the numerical solution of $AX - XB = C$, *BIT*, vol. 36, pp. 814-830, 1996.

(54) D.C. Sorensen, Implicit application of polynomial filters in a k-step Arnoldi method, *SIAM J. Matrix Anal. Appl.*, vol. 13, pp. 357–385, 1992.

(55) T.-J. Su and R.R. Craig, Jr., Model reduction and control of flexible structures using Krylov vectors, *J. Guidance Control Dynamics,* vol. 14, pp. 260-267, 1991.

(56) P. Van Dooren, *The Lanczos Algorithm and Padé Approximations*, Short Course, Benelux Meeting on Systems and Control, 1995.

(57) P. Van Dooren, *Gramian Based Model Reduction of Large-scale Dynamical Systems,* Short Course SIAM Annual Meeting, San Juan, Puerto Rico, July, 2000.

(58) P. Van Dooren, Numerical linear algebra techniques for large scale matrix problems in systems and control, *Proc.* 31st *Conference on Decision and Control,* Tucson, AZ, 1992.

(59) J.H. Wilkinson, *The Algebraic Eigenvalue Problem*, Clarendon, Oxford, 1965.