Title	A fast Newton/Smith algorithm for solving algebraic Riccati equations and its application in model order reduction
Author(s)	Wong, N; Balakrishnan, V; Koh, CK; Ng, TS
Citation	Icassp, Ieee International Conference On Acoustics, Speech And Signal Processing - Proceedings, 2004, v. 5, p. V-53-V-56
Issued Date	2004
URL	http://hdl.handle.net/10722/45855
Rights	Creative Commons: Attribution 3.0 Hong Kong License

A FAST NEWTON/SMITH ALGORITHM FOR SOLVING ALGEBRAIC RICCATI EQUATIONS AND ITS APPLICATION IN MODEL ORDER REDUCTION

Ngai Wong*, Venkataramanan Balakrishnan†, Cheng-Kok Koh†, and Tung-Sang Ng*

*Department of Electrical and Electronic Engineering
The University of Hong Kong
Pokfulam Road, Hong Kong
Emails: {nwong, tsng}@eee.hku.hk

†School of Electrical and Computer Engineering
Purdue University
West Lafayette, IN, USA
Emails: {ragu, chengkok}@ecn.purdue.edu

ABSTRACT

A very fast Smith-method-based Newton algorithm is introduced for the solution of large-scale continuous-time algebraic Riccati equations (CAREs). When the CARE contains low-rank matrices, as is common in the modeling of physical systems, the proposed algorithm, called the Newton/Smith CARE or *NSCARE* algorithm, offers significant computational savings over conventional CARE solvers. Effectiveness of the algorithm is demonstrated in the context of VLSI model order reduction wherein stochastic balanced truncation (SBT) is used to reduce large-scale passive circuits. It is shown that the NSCARE algorithm exhibits guaranteed quadratic convergence under mild assumptions. Moreover, two large-sized matrix factorizations and one large-scale singular value decomposition (SVD) necessary for SBT can be omitted by utilizing the Smith method output in each Newton iteration, thereby significantly speeding up the model reduction process.

1. INTRODUCTION

Solution of the continuous-time algebraic Riccati equation (CARE) has long been an active research topic, perhaps the most prominent applications of it are in Kalman filtering, H_∞-control, and the linear quadratic regulator (LQR) problem [1]. Some standard ways of solving a CARE include the eigenvector and Schur vector methods, the multishift algorithm, and the matrix sign function approach (e.g., [1, 2]). These methods rely on identifying the stable invariant subspace of the corresponding *Hamiltonian* matrix. Alternatively, the Newton algorithm tackles a CARE by solving a Lyapunov equation in each iteration step [1, 3]. This approach is characterized by its high numerical accuracy, but due to the lack of guaranteed convergence, the Newton algorithm is mainly used to refine an approximate solution from a more robust approach. These methods, however, are general CARE solvers and do not exploit the low-rank nature of certain matrices in forming the CARE. Consequently, the solution of a large-scale CARE is highly constrained by its computational complexity. In fact, numerical examples show that even solving a moderately sized CARE is computationally intensive. Heuristic approaches are presented in [4] to handle very large CAREs with low-rank and sparse matrices, but theoretical basis and convergence proof are not available.

Our main contribution is a very fast Smith-method-based Newton algorithm, called the Newton/Smith CARE or *NSCARE* algorithm, for the efficient solution of large-scale CAREs containing low-rank matrices. The algorithm is numerically stable via the use of Krylov subspace method. Moreover, it requires *one and only one* step of $O(n^3)$ work with all other steps being $O(n^2)$, thus providing considerable savings over conventional solvers.

To demonstrate its effectiveness, the NSCARE algorithm is applied in the model reduction of large passive[1] circuits. Specifically, in the context of VLSI synthesis, interconnect and package modelings are critical as circuits are designed with ever-increasing speed and complexity (e.g., [5–10]). Using methods such as sparse tableau and modified nodal analysis (MNA) [11], the RLC interconnect/package model extracted from a netlist often results in an order not amenable to simulation or analysis. A class of control-theoretic techniques, collectively known as the *balanced truncation*, provides a way to reduce high-order models. Examples of these techniques include the optimal Hankel-norm approximation [12], standard balanced truncation (BT) [7,13], and stochastic balanced truncation (SBT) [8–10, 14, 15]. Major merits of these approaches, in contrast to the traditional principle subspace projection (e.g., [5, 6]), are their superior accuracies and availability of approximation error bounds [12, 15]. For a stable and passive original model, balanced truncation automatically preserves stability in the reduced-order model. SBT, among other balanced truncation schemes, is of particular interest because it further guarantees passivity which is necessary for correction simulation of the reduced-order model [5, 16]. Consider a state space model of

$$\dot{x} = Ax + Bu \tag{1a}$$

$$y = Cx + Du \tag{1b}$$

where $A \in \Re^{n \times n}$, $B \in \Re^{n \times m}$, $C \in \Re^{m \times n}$, $D \in \Re^{m \times m}$. B, C are of low ranks, i.e., $m \ll n$, and u, y are power-conjugate[2]. Using $M > 0$ ($M \geq 0$) to denote a positive definite (positive semidefinite) matrix M, we assume $(D + D^T) > 0$. For RLC state space models, we also have $A + A^T \leq 0$ [16]. By the positive real lemma [16], a passive system (which is always true for pure RLC circuits) is equivalent to the existence of a $P \in \Re^{n \times n}$, $P = P^T \geq 0$, satisfying the linear matrix inequality (LMI)

$$\begin{bmatrix} A^T P + PA & PB - C^T \\ B^T P - C & -(D + D^T) \end{bmatrix} \leq 0 \tag{2}$$

Note that when MNA formulation is used for RLC modeling, we have $D = 0$ and $B = C^T$. In this case, the system can be transformed into an equivalent lower-order system satisfying the format

---

[1]Roughly, a passive system is one that does not generate energy.

[2]That is, for every component of u that is a node voltage (branch current), the corresponding component of y is a branch current (node voltage).

and conditions in (2) [9], so (2) will be assumed without loss of generality. Using Schur complement, (2) is equivalent to

$$A^T P + PA + (PB - C^T)(D + D^T)^{-1}(B^T P - C) \le 0 \quad (3)$$

The solution of (3) being zero is a CARE. Taking the matrix root $UU^T = (D + D^T)^{-1}$ and defining $\hat{B} = BU$, $\hat{C} = U^T C$, and $\hat{A} = A - \hat{B}\hat{C}$, the CARE is expressible as

$$F(P) = \hat{A}^T P + P\hat{A} + P\hat{B}\hat{B}^T P + \hat{C}^T \hat{C} = 0 \quad (4)$$

The solution of (4), if it exists, is not unique. Among them there is a unique stabilizing solution, P_∞, in the sense that $(\hat{A} + \hat{B}\hat{B}^T P_\infty)$ is stable [1, 3] (i.e., all eigenvalues have negative real parts).

2. PRELIMINARIES

2.1. Stochastic Balanced Truncation

In SBT, the most reachable state with a given input energy, quantified by $\int u(t)^T y(t) dt$, is aligned with the state that delivers maximum energy to the output, quantified by $-\int u(t)^T y(t) dt$ [9]. The least involved states are then truncated to give the best approximating low-order system in this energy sense. The alignment process starts with solving the minimal and maximal solutions, P_{min} and P_{max}, to the LMI (2) such that equality holds. This is equivalent to finding the stabilizing solutions to the two dual CAREs

$$\hat{A}^T P_{min} + P_{min}\hat{A} + P_{min}\hat{B}\hat{B}^T P_{min} + \hat{C}^T \hat{C} = 0 \quad (5a)$$
$$\hat{A}P_{max}^{-1} + P_{max}^{-1}\hat{A}^T + P_{max}^{-1}\hat{C}^T \hat{C}P_{max}^{-1} + \hat{B}\hat{B}^T = 0 \quad (5b)$$

Let $P_{max}^{-1} = XX^T$, $P_{min} = YY^T$ be any root decompositions (X and Y can be of low-rank but are assumed to have the same number of columns k, $k \le n$, otherwise simply pad zero columns to one of them). Now do the singular value decomposition (SVD)

$$X^T Y = U\Sigma V^T \quad (6)$$

where $\Sigma \ge 0$ is a k-by-k diagonal matrix with singular values in descending order. Suppose the singular values of Σ are

$$\sigma_1 \ge \sigma_2 \ge \cdots \ge \sigma_r \gg \sigma_{r+1} \ge \cdots \ge \sigma_k \quad (7)$$

Define I_m to be the identity matrix of dimension m, $0_{m \times n}$ an $m \times n$ zero matrix, and

$$T_L = \begin{bmatrix} I_r & 0_{r \times (k-r)} \end{bmatrix} \Sigma^{-\frac{1}{2}} V^T Y^T \quad (8a)$$

$$T_R = XU\Sigma^{-\frac{1}{2}} \begin{bmatrix} I_r \\ 0_{(k-r) \times r} \end{bmatrix} \quad (8b)$$

The system $(T_L A T_R, T_L B, C T_R, D)$ represents the stochastically balanced reduced-order model whose states are aligned in descending "energy-wise importance". The best bound to date for the frequency domain approximation error can be found in [15].

2.2. Newton Algorithm and the Smith Method

Advantages of using Newton algorithm in solving CAREs include its quadratic convergence (once attained) and high numerical accuracy [1, 3]. Let P_j be the current estimation of the solution, we

define $P_{j+1} = P_j + \delta P_j$ where δP_j is the search direction or Newton step. Substituting P_{j+1} into (4) we get

$$F(P_j + \delta P_j) = F(P_j) + (\hat{A} + \hat{B}\hat{B}^T P_j)^T \delta P_j$$
$$+ \delta P_j(\hat{A} + \hat{B}\hat{B}^T P_j) + \delta P_j \hat{B}\hat{B}^T \delta P_j \quad (9)$$

Every Newton iteration is a first-order error correction such that the sum of the first three terms on the right of (9) is zeroed, i.e.,

$$(\hat{A} + \hat{B}\hat{B}^T P_j)^T \delta P_j + \delta P_j(\hat{A} + \hat{B}\hat{B}^T P_j) + F(P_j) = 0 \quad (10)$$

This is simply a Lyapunov equation. After each step, (9) is left with a quadratic residual term. So from the second step onwards,

$$F(P_j) = \delta P_{j-1}\hat{B}\hat{B}^T \delta P_{j-1} \quad (11)$$

$j = 1, 2, \cdots$. The Smith method (e.g., [7–9]) provides a way to solve (10) by transforming a continuous-time Lyapunov equation into a discrete-time version having the same solution. For instance, the following two equations solve the same W_o:

$$A_c^T W_o + W_o A_c + C_c^T C_c = 0 \quad (12a)$$
$$A_z^T W_o A_z - W_o + C_z^T C_z = 0 \quad (12b)$$

where $A_z = (pI + A_c)(pI - A_c)^{-1}$, $C_z = -\sqrt{2p}C_c(pI - A_c)^{-1}$, $p > 0$ is a shift parameter, and the subscripts c and z stand for continuous and discrete, respectively. Subsequently, $W_o = \sum_{i=0}^{\infty} (A_z^T)^i C_z^T C_z A_z^i$. In practice we want to minimize the norm of A_z so the power terms decay quickly and the infinite summation can well be represented by finite terms. A good choice of p to achieve this is $p = \sqrt{|\lambda_{max}(A_c)| |\lambda_{min}(A_c)|}$ [9], where $\lambda_{max}(\circ)$ and $\lambda_{min}(\circ)$ denote the maximum and minimum eigenvalues, respectively. An important observation is that W_o is naturally cast as a matrix factorization, namely, when the growth of the summation decays to machine precision after k terms,

$$W_o \approx \sum_{i=0}^{k-1} (A_z^T)^i C_z^T C_z A_z^i = \mathcal{K}_k(A_z^T, C_z^T)\mathcal{K}_k(A_z^T, C_z^T)^T \quad (13)$$

where the factor $\mathcal{K}_k(A, B) = \begin{bmatrix} B & AB & \cdots & A^{k-1}B \end{bmatrix}$ is called kth-order Krylov matrix. Application of the Smith method in standard BT of VLSI models can be found in [7, 9].

3. THE NSCARE ALGORITHM

Defining the Lyapunov operator $\mathcal{L}_A : \Re^{n \times n} \to \Re^{n \times n}$ as $\mathcal{L}_A(P) = A^T P + PA$, the NSCARE algorithm that solves (4) is given here:

Algorithm 1 : NSCARE: Input ($\hat{A}$, $\hat{B}$, $\hat{C}$, P_0, max_itr, tol)
1. Find the shift p corresponding to $\hat{A}$;
2. Compute $\Phi = (pI + \hat{A})$ and $\Psi = (pI - \hat{A})^{-1}$;
3. Solve for δP_0 in $\mathcal{L}_{\hat{A}+\hat{B}\hat{B}^T P_0}(\delta P_0) + F(P_0) = 0$ with standard solvers, or when $P_0 = 0$, solve $\mathcal{L}_{\hat{A}}(\delta P_0) + \hat{C}^T \hat{C} = 0$ using Algorithm 2 with input $((\Phi\Psi)^T, (-\sqrt{2p}\hat{C}\Psi)^T$, max_itr, tol);
4. Set $j = 1$;
5. while $j \le$ max_itr;
   5.1 Set $P_j = P_{j-1} + \delta P_{j-1}$;
   5.2 Set $\Gamma = \hat{B}^T P_j$ and $\Theta = \hat{B}^T \delta P_{j-1}$;
   5.3 Set $A_z = (\Phi + \hat{B}\Gamma)(\Psi + \Psi\hat{B}(I - \Gamma\Psi\hat{B})^{-1}\Gamma\Psi)$;
   5.4 Set $C_z = -\sqrt{2p}\Theta(\Psi + \Psi\hat{B}(I - \Gamma\Psi\hat{B})^{-1}\Gamma\Psi)$;
   5.5 Solve for δP_j in $\mathcal{L}_{\hat{A}+\hat{B}\Gamma}(\delta P_j) + \Theta^T \Theta = 0$ using

Algorithm 2 with input (A_z^T, C_z^T, max_itr, tol);
5.6 if the Frobenious norm $\|\delta P_j\|_F < tol$
    5.6.1 Set $P_\infty = P_j + \delta P_j$;
    5.6.2 Break while loop and return P_∞;
end if
5.7 Set $j = j + 1$;
end while

Arnoldi and Lanczos algorithms [17] are numerically well-conditioned algorithms to obtain the Krylov matrix in the Smith method. We present only the Arnoldi algorithm due to space limitation. The following codes assume a rank-one input B_p, but block versions of the Arnoldi and Lanczos algorithms are readily available to deal with B_p of arbitrary ranks (e.g., [5, 16]). In summary, the Arnoldi algorithm iteratively computes the k orthogonal columns of $Q_k \in \Re^{n \times k}$, an upper triangular matrix $R_k \in \Re^{k \times k}$, and an accumulation matrix $\Omega_k \in \Re^{n \times n}$ such that

- $Q_k^T Q_k = I_k$;
- $H_k = Q_k^T A_p Q_k$ is a k-by-k upper Hessenberg matrix;
- $\mathcal{K}_k(A_p, B_p) = \begin{bmatrix} B_p & A_p B_p & \cdots & A_p^{k-1} B_p \end{bmatrix} = Q_k R_k$ is a QR factorization;
- Q_k spans the range of $\mathcal{K}_k(A_p, B_p)$;
- $\Omega_k = \mathcal{K}_k(A_p, B_p)\mathcal{K}_k(A_p, B_p)^T = (Q_k R_k)(Q_k R_k)^T$.

Algorithm 2: SmithLyapunov: Input (A_p, B_p, max_itr, tol)
$j = 1$;
$q_1 = B_p/\|B_p\|_2$; $\beta = 1$; $Q_1 = q_1$; $R_1 = \|B_p\|_2$;
$H_1 = [\ ]$; $\Omega_1 = B_p B_p^T$;
while $j \le$ max_itr {
  for $i = 1 : j$
      $h_{ij} = q_i^T A_p q_j$;
  end
  $r_{j+1} = A_p q_j - \Sigma_{i=1}^{j} h_{ij} q_i$;

$$H_j = \begin{bmatrix} & H_{j-1} & & \begin{bmatrix} h_{1j} \\ \vdots \\ h_{j-1,j} \end{bmatrix} \\ \begin{bmatrix} 0 & \cdots & 0 & \beta \end{bmatrix} & & h_{jj} \end{bmatrix};$$

  if $j > 1$
$$R_j = \begin{bmatrix} \begin{matrix} R_{j-1} \\ \begin{bmatrix} 0 & \cdots & 0 \end{bmatrix} \end{matrix} & \Big| & H_j \begin{bmatrix} R_{j-1}(:, j-1) \\ 0 \end{bmatrix} \end{bmatrix};$$
    $\omega_j = Q_j R_j(:, j)$;
    $\Omega_j = \Omega_{j-1} + \omega_j \omega_j^T$;
    if ($\|\omega_j\|_2 < tol$) break while loop;
  end if
  $\beta = \|r_{j+1}\|_2$;
  if ($\beta < tol$) break while loop;
  $q_{j+1} = r_{j+1}/\beta$;
  $Q_{j+1} = \begin{bmatrix} Q_j & q_{j+1} \end{bmatrix}$;
  $j = j + 1$;
end while
$k = $ number of columns in R_j;
Return Ω_k, Q_k, R_k, and H_k.

Several remarks are in order:

1. In practice the tolerance parameter, *tol*, in both algorithms are set to a small value (e.g., the machine precision). In the first call to Algorithm 2 (i.e., finding δP_0 or δP_1, depending on the initial P_0), the number of iterations is usually the highest, and then decreases in the following runs as quadratic convergence is acquired;

2. In Step 1 of the NSCARE algorithm, p is approximated by first applying a Lanczos algorithm for κ steps on $\hat{A}$ to obtain a tridiagonal matrix $T_\kappa \in \Re^{\kappa \times \kappa}$, $\kappa \ll n$, whose eigenvalues closely approximate those extremal eigenvalues of $\hat{A}$. Then a simple (inverse) power iteration [17] is used to estimate the magnitude of the maximum (minimum) eigenvalue of T_κ so as to form p. The initial Lanczos process has $O(\kappa n^2)$ work and the power iterations require $O(\kappa^3)$ work (in our implementation, $\kappa = 50$ and each power iteration consists of 30 iterative steps);

3. In Steps 5.3 and 5.4 of the NSCARE algorithm, the matrix inversion lemma is used to calculate $(pI - (\hat{A} + \hat{B}\hat{B}^T P_j))^{-1} = (\Psi^{-1} - \hat{B}\Gamma)^{-1} = (\Psi + \Psi\hat{B}(I - \Gamma\Psi\hat{B})^{-1}\Gamma\Psi)$. So the actual matrix inversion is bypassed and the work involved is reduced from $O(n^3)$ down to $O(mn^2)$;

4. Theoretically, for each P_j update, the shift p should be recalculated regarding the extremal eigenvalues of $(\hat{A} + \hat{B}\hat{B}^T P_j)$. However, when P_j is quadratically converging, the norm of δP_j is small and has little effect on p. Subsequently, the same p can be used throughout the algorithm so that the matrix inversion lemma can be utilized for low-rank updating of $(\Psi^{-1} - \hat{B}\Gamma)^{-1}$;

5. The *one and only one* step in the NSCARE algorithm that requires $O(n^3)$ work is the finding of Ψ (Step 2). In case of sparse matices wherein the inversion can be done with $O(n^2)$ work, the whole algorithm will further reduce to an $O(n^2)$ algorithm;

6. Convergence analysis of the NSCARE algorithm follows closely from those in [1, 3]. To save space, the main results are given without elaboration: i) for a stabilizing initial guess P_0, the subsequent Lyapunov operators in each Newton iteration are nonsingular and P_j, $j = 1, 2, \cdots$, are also stabilizing; ii) $0 \le \cdots \le P_j \le P_{j+1} \le P_\infty$; iii) $0 \le \|P_\infty - P_{j+1}\| \le \gamma \|P_\infty - P_j\|^2$ where γ is a positive constant, i.e., convergence is quadratic once P_j falls into the region of convergence;

7. For a strictly dissipative system such as an RLC circuit modeled to high fidelity, it can be shown that there exists a representation such that strict inequality is satisfied in (2) with $P = I$ (c.f., [8, 9]). It follows that $\hat{A}$ is stable and the initial guess of $P_0 = 0$ is stabilizing. Moreover, since $0 \le P_j \le P_\infty < I$, we have $\|P_\infty - P_j\| < 1$, $j = 0, 1, \cdots$. In other words, under the mild assumption of a strictly passive (dissipative) system, quadratic convergence of the NSCARE algorithm is guaranteed;

8. Suppose (5a) is solved with the NSCARE algorithm in N Newton steps with $P_0 = 0$. Let the stabilizing solution be $P_{min} = P_\infty$ and the number of iterations in each call to Algorithm 2 be k_1, k_2, $\cdots$, k_N, and $k_T = k_1 + k_2 + \cdots + k_N$. Then in terms of the output of Algorithm 2, $P_{min} = \sum_{i=0}^{N}(Q_{k_i} R_{k_i})(Q_{k_i} R_{k_i})^T = (Q_{k_T} R_{k_T})(Q_{k_T} R_{k_T})^T$ where $Q_{k_T} = \begin{bmatrix} Q_{k_1} & \cdots & Q_{k_N} \end{bmatrix}$ and $R_{k_T} = diag(R_{k_1}, \cdots, R_{k_N})$. Thus referring to (6), a factor of P_{min} is given by $Y = Q_{k_T} R_{k_T}$. The factor X of P_{max}^{-1} is obtained in a similar way. In practice $N \le 5$ and $k_T \le 100$ regardless of the CARE size, thus only a moderately sized (≤ 100) SVD is needed even for high-order initial models. As a result, the NSCARE algorithm helps to completely elude two large-sized matrix factorizations and one large-scale SVD in the traditional SBT implementation.

4. NUMERICAL EXAMPLES

We study randomly generated strictly passive systems, with $A + A^T < 0$ and rank-one B, C, satisfying the conditions in (1)-(4). CAREs of various orders are formed and the NSCARE algorithm

(a) CPU time for solving one CARE

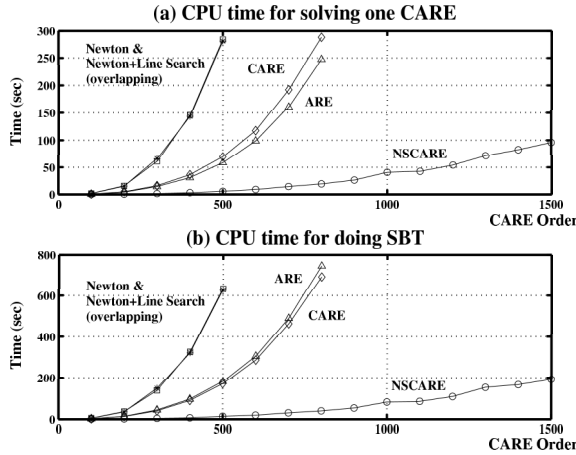**(b) CPU time for doing SBT**

Fig. 1. CPU time measurements of various solvers.

is tested against common solvers of $O(n^3)$ complexity. They include the Newton algorithm with and without line search, and the MATLAB functions *ARE* and *CARE* implementing the Schur vector and eigenvalue methods, respectively (see [1, 3] and the references therein). Numerical experiments were done in MATLAB on a 1.8GHz 512MB-RAM notebook with other applications running. The CPU time for solving a single CARE is shown in Fig. 1(a). It is seen that the NSCARE algorithm can easily handle CAREs of order as high as 1500, while other solvers face computational difficulty well below 1000 (at the order of 800, the NSCARE algorithm is at least 10 times faster). Fig. 1(b) shows the CPU time for realizing SBT. As outlined in the remarks, significant computational saving is possible by utilizing intermediate outputs of the Smith method. For example, at the order of 800, the NSCARE algorithm is 17 times faster than the next best candidate. These results confirm the effectiveness of the NSCARE algorithm both as a CARE solver (when convergence can be sought), and as a particularly powerful tool for carrying out SBT model reduction of large-scale passive systems such as the RLC circuits frequently encountered in VLSI synthesis.

5. CONCLUSION

This paper has introduced a very fast NSCARE algorithm for solving continuous-time algebraic Riccati equations (CAREs). It exploits the low-rank nature of the input and output matrices commonly found in physical state space models. The algorithm is generic in nature but guaranteed convergence can be proven, with mild assumptions, in the context of stochastic balanced truncation (SBT) of large-scale passive models. We have also demonstrated how two large-sized matrix factorizations and one large-scale singular value decomposition (SVD) necessary for SBT can be completely avoided. This has rendered the NSCARE algorithm highly efficient both as a CARE solver and as a versatile tool for performing SBT. Numerical examples have confirmed the effectiveness of the NSCARE algorithm over conventional CARE solvers.

6. REFERENCES

[1] V. Mehrmann, *The Autonomous Linear Quadratic Control Problem: Theory And Numerical Solution*, ser. No. 163 Lecture Notes in Control and Information Sciences. Berlin, Heidelberg: Springer-Verlag, Nov. 1991.

[2] G. S. Ammar, P. Benner, and V. Mehrmann, "A multishift algorithm for the numerical solution of algebraic Riccati equations," *Electr. Trans. Num. Anal.*, vol. 1, pp. 33–48, Sept. 1993.

[3] P. Benner and R. Byers, "An exact line search method for solving generalized continuous-time algebraic Riccati equations," *IEEE Trans. Automat. Contr.*, vol. 43, no. 1, pp. 101–107, Jan. 1998.

[4] A. S. Hodel and K. R. Poolla, "Heuristic approaches to the solution of very large sparse Lyapunov and algebraic Riccati equations," in *Proc. Conf. on Decision and Control*, Dec. 1988, pp. 2217–2221.

[5] A. Odabasioglu, M. Celik, and L. T. Pileggi, "PRIMA: Passive reduced-order interconnect macromodeling algorithm," *IEEE Trans. Computer-Aided Design*, vol. 17, no. 8, pp. 645–654, Aug. 1998.

[6] P. Feldmann and R. W. Freund, "Efficient linear circuit analysis by Padé approximation via the Lanczos process," *IEEE Trans. Computer-Aided Design*, vol. 14, no. 5, pp. 639–649, May 1995.

[7] Q. Su, V. Balakrishnan, and C.-K. Koh, "Efficient approximate balanced truncation of general large-scale RLC systems via Krylov methods," in *Proc. ASPDAC/Int. Conf. VLSI Design*, Jan. 2002, pp. 311–316.

[8] ——, "A factorization-based framework for passivity-preserving model reduction of RLC systems," in *Proc. Design Automation Conference*, June 2002, pp. 40–45.

[9] Q. Su, "Algorithms for model reduction of large scale RLC systems," Ph.D. dissertation, School of ECE, Purdue University, Aug. 2002.

[10] J. R. Phillips, L. Daniel, and L. M. Silveira, "Guaranteed passive balancing transformations for model order reduction," *IEEE Trans. Computer-Aided Design*, vol. 22, no. 8, pp. 1027–1041, Aug. 2003.

[11] J. Vlach and K. Singhal, *Computer Methods for Circuit Analysis and Design*. Kluwer Academic Publishers, July 1993.

[12] K. Glover, "All optimal Hankel-norm approximation of linear multivariable systems and their L^∞-error bounds," *Int. J. Control*, vol. 39, no. 6, pp. 1115–1193, June 1984.

[13] B. Moore, "Principal component analysis in linear systems: Controllability, observability, and model reduction," *IEEE Trans. Automat. Contr.*, vol. 26, no. 1, pp. 17–32, Feb. 1981.

[14] M. Green, "Balanced stochastic realizations," *Linear Algebra Appl.*, vol. 98, pp. 211–247, 1988.

[15] X. Chen and J. T. Wen, "Positive realness preserving model reduction with H_∞ norm error bounds," *IEEE Trans. Circuits Syst. I*, vol. 42, no. 1, pp. 23–29, Jan. 1995.

[16] Z. Bai, P. M. Dewilde, and R. W. Freund, "Reduced-order modeling," *Numerical Analysis Manuscript No. 02-4-13, Bell Laboratories*, Mar. 2002.

[17] G. Golub and C. V. Loan, *Matrix Computations*, 2nd ed. Baltimore: JohnsHopkins Univ. Press, 1989.