

# Two Algorithms for Fast and Accurate Passivity-Preserving Model Order Reduction

Ngai Wong, *Member, IEEE*, Venkataramanan Balakrishnan, *Member, IEEE*,

Cheng-Kok Koh, *Member, IEEE*, and Tung-Sang Ng, *Fellow, IEEE*

## Abstract

This paper presents two recently developed algorithms for efficient model order reduction. Both algorithms enable the fast solution of continuous time algebraic Riccati equations (CAREs) that constitute the bottleneck in the passivity-preserving balanced stochastic truncation (BST). The first algorithm is a Smith-method-based Newton algorithm, called Newton/Smith CARE or NSCARE, that exploits low rank matrices commonly found in physical system modeling. The second algorithm is a project-and-balance scheme that utilizes dominant eigenspace projection, followed by simultaneous solution of a pair of dual CAREs through completely separating the stable and unstable invariant subspaces of a Hamiltonian matrix. The algorithms can be applied individually or together. Numerical examples show the proposed algorithms offer significant computational savings and better accuracy in reduced order models over those from conventional schemes.

## Index Terms

Balanced stochastic truncation, algebraic Riccati equation, Newton method, Smith method, SR algorithm

\*\*MANUSCRIPT (CONTROL NO. 2646) SUBMITTED FOR PUBLICATION AS A REGULAR PAPER.

This material is based upon work supported in part by the Hong Kong Research Grants Council under Project HKU 7173/04E, in part by the University Research Committee of The University of Hong Kong, and in part by the National Science Foundation of the United States of America under Grant Nos. ECS-0200320 and CCR-0203362. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the funding agencies.

N. Wong and T.-S. Ng are with the Department of Electrical and Electronic Engineering, The University of Hong Kong. Phone: ++852 +2859 1914 Fax: ++852 +2559 8738 Email: {nwong, tsng}@eee.hku.hk

V. Balakrishnan and C.-K. Koh are with the School of Electrical and Computer Engineering, Purdue University. Phone: +1 765 494 0728 Fax: +1 765 494 6951 Email: {ragu, chengkok}@ecn.purdue.edu

# Two Algorithms for Fast and Accurate Passivity-Preserving Model Order Reduction

## I. INTRODUCTION

In backend verification of VLSI design, initial state space modeling of interconnect and pin packages easily involves thousands or millions of state variables, thereby prohibiting direct computer simulation and analysis. Model order reduction (MOR) (e.g., [1]–[20]) has become an integral step wherein the original linear model is reduced to, and approximated by, a much smaller linear model. It is desirable that the reduced order model has small error over the frequency and/or time domains. Important properties such as stability and passivity<sup>1</sup> must also be preserved along the reduction process in order for the reduced order models to be useful [1], [2].

MOR techniques include transfer function moment matching (e.g., asymptotic waveform evaluation (AWE) [3]), Krylov subspace projection (e.g., Padé Approximation via Lanczos (PVL) [4], matrix PVL (MPVL) [5]), and the passivity-preserving congruence transform (e.g., PRIMA [1]). These schemes can be implemented by computationally efficient Krylov iterations [21], and often work well; however, they are “feasible” designs, and not “optimal” in terms of any approximation criterion. Another class of techniques stem from control theory. Examples include optimal Hankel-norm approximation [6], standard balanced truncation (BT) [7]–[9], [13], and the passivity-preserving balanced stochastic truncation (BST) [9]–[12], [14], [20]. Merits of these control-theoretic approaches are their superior global accuracy and deterministic error bounds [6], [11]. These schemes are, however, expensive to deploy due to the need of solving large size matrix equations and decompositions. For example, standard BT requires solving a pair of *Lyapunov equations* (linear matrix equations), while BST calls for the solution of a pair of *continuous time algebraic Riccati equations* or CAREs (quadratic matrix equations) [22]. To alleviate the cost of standard BT, a series of recent work took advantage of the low rank input/output matrices arising in many physical systems, and developed the Cholesky factor standard BT variants (e.g., [13], [15]–[17], [23]). These schemes are mainly based on the alternating direction implicit (ADI) method of solving Lyapunov equations and have comparable speed to the popular projection-based methods. However, standard BT does not guarantee passivity. BST preserves both passivity and stability and poses no special structural

<sup>1</sup>A passive system is one that does not generate energy internally. A dissipative system, such as an RLC network, is passive.

requirements on the original state space [12], but suffers from the high computational cost of solving CAREs. In fact, solving even a moderately sized CARE can be computationally intensive [18]. Heuristics in [24] tackle large CAREs with low rank and sparse matrices, but theoretical basis and convergence proof are unavailable.

Standard techniques of solving a CARE include forming a *Hamiltonian matrix* and identifying its stable invariant subspace [22], [25]–[27]. Another way, provided a stabilizing initial condition is known, is to use the Newton method that solves a Lyapunov equation in each iteration [22], [28]. In this paper, we summarize and report our recent work on fast implementations of both the Newton and Hamiltonian approaches in the context of large scale BST [18], [19]. The first contribution is a Smith-method-based Newton algorithm, called the Newton/Smith CARE or *NSCARE* algorithm, for quickly solving a large scale CARE containing low rank input/output matrices [18]. The algorithm uses Krylov subspace iterations and is numerically stable. The second contribution is an effective two-stage project-and-balance reduction algorithm [19], which provides a framework for trading off computational cost against model approximation accuracy. The projection basis in the first stage is formed by the dominant eigenspaces of the controllability and observability Gramians [8], [15], [17]. The projected, intermediate model is then further reduced by BST. A novel observation, which relies on completely separating the stable and unstable invariant subspaces of a Hamiltonian matrix, reveals that two dual CAREs in BST can be jointly solved at the cost of essentially one. Numerical examples show the proposed algorithms exhibit fast reduction and deliver excellent model accuracy.

The paper is organized as follows. Section II presents the problem setting and preliminaries. Section III introduces the NSCARE algorithm and Section IV presents the project-and-balance algorithm. Numerical examples in Section V demonstrate the effectiveness of the proposed algorithms over conventional approaches. Finally, Section VI draws the conclusion.

## II. BACKGROUND AND PRELIMINARIES

A target application of the proposed algorithms is in the reduction of large scale RLC (and therefore passive) circuits commonly encountered in VLSI interconnect and package simulations. Consider a *minimal* state space model of

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

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

where  $A \in \mathbb{R}^{n \times n}$ ,  $B \in \mathbb{R}^{n \times m}$ ,  $C \in \mathbb{R}^{m \times n}$ ,  $D \in \mathbb{R}^{m \times m}$ ,  $B, C$  are of low rank, i.e.,  $m \ll n$ , and  $u \in \mathbb{R}^m$ ,  $y \in \mathbb{R}^m$  are power-conjugate<sup>2</sup>. The system matrix  $A$  is assumed stable, or equivalently its spectrum is in the open left half plane, denoted by  $\text{spec}(A) \subset \mathbb{C}^-$ . We assume that  $D + D^T > 0$ , where the notation  $M > 0$  ( $M \geq 0$ ) means that the symmetric matrix  $M$  is positive definite (positive semidefinite).

RLC state space models from modified nodal analysis (MNA) [29] in VLSI interconnect modeling have the properties of  $A + A^T \leq 0$ ,  $B = C^T$ , and  $D = 0$  [2]. Such system can be recast into an equivalent form with  $D + D^T > 0$  [9], [30]. Moreover, an RLC system in the *descriptor* format [2] with a singular  $E (\in \mathbb{R}^{n \times n})$  preceding  $\dot{x}$  can be transformed into an equivalent minimal form in (1) [13], so the settings in (1) are assumed without loss of generality.

The controllability Gramian,  $W_c$ , and observability Gramian,  $W_o$ , are solved through the following continuous time Lyapunov equations

$$AW_c + W_cA^T + BB^T = 0 \quad (2)$$

$$A^TW_o + W_oA + C^TC = 0. \quad (3)$$

The spans (ranges) of  $W_c$  and  $W_o$  denote the reachable and observable states, respectively. For many physical systems including RLC circuits,  $W_c$  and  $W_o$  are of low numerical rank or approximately so. The implication is that the state activities usually take place in, or through projection can be well captured by, some lower dimensional subspaces [23].

#### A. Smith Method

The Smith method (e.g., [9], [13], [16]) solves a continuous time Lyapunov equation by transforming it into a discrete time version having exactly the same solution. For instance, the following two equations solve the same  $W_c$ :

$$AW_c + W_cA^T + BB^T = 0 \quad (4a)$$

$$A_pW_cA_p^T - W_c + B_pB_p^T = 0 \quad (4b)$$

where  $A_p = (A - pI)(A + pI)^{-1}$ ,  $B_p = \sqrt{-2p}(A + pI)^{-1}B$ ,  $p \in \mathbb{C}^-$  is a shift parameter. It follows that  $W_c = \sum_{i=0}^{\infty} A_p^i B_p B_p^T (A_p^T)^i$ . In practice, we want to minimize the spectral radius of  $A_p$  so the power terms decay quickly and the infinite summation can be well approximated by finite terms. A simple

<sup>2</sup>For every component of  $u$  that is a node voltage (branch current), the corresponding component of  $y$  is a branch current (node voltage) so that  $u^T y$  represents the instantaneous power injected into the system.

possible choice is  $p = -\sqrt{|\lambda_{max}(A)| |\lambda_{min}(A)|}$  [9], where  $\lambda_{max}(\circ)$  and  $\lambda_{min}(\circ)$  denote the maximum- and minimum-modulus eigenvalues. An important observation is that  $W_c$  is naturally cast as a matrix factorization, namely, when the growth of the summation becomes negligible after  $k$  terms,

$$W_c \approx \sum_{i=0}^{k-1} A_p^i B_p B_p^T (A_p^T)^i = \mathcal{K}_k(A_p, B_p) \mathcal{K}_k(A_p, B_p)^T \quad (5)$$

where  $\mathcal{K}_k(A_p, B_p) = \begin{bmatrix} B_p & A_p B_p & \cdots & A_p^{k-1} B_p \end{bmatrix}$  is called the  $k$ -th order Krylov matrix and serves as a Cholesky factor of  $W_c$ . Obviously, this Krylov matrix factorization is computationally advantageous when  $B_p$  is of low rank and when the rank information of the Krylov matrix can be revealed quickly. Application of the Smith method in standard BT of VLSI models can be found in [9], [13]. It should be noted that the Smith method is mathematically equivalent to the ADI method with a single shift parameter [16], [17], [23]. In fact, in our algorithms, the Smith-method-based parts can readily be replaced with the ADI scheme with multiple shifts. Smith method is chosen because of its ease of exposition, and also that it requires only one large scale matrix inversion (in finding  $A_p$ ) that constitutes the most expensive step in both proposed algorithms.

### B. Krylov Subspace Iteration

The structure of a Krylov matrix lends itself to iterative computation. Among others, Arnoldi and Lanczos algorithms [21] are numerically efficient procedures for obtaining the Krylov matrix. We present only the Arnoldi algorithm here due to space limitation. The following MATLAB-style pseudo codes assume a rank-one  $B_p$ , but block versions of Arnoldi and Lanczos algorithms are available for  $B_p$  of arbitrary ranks (e.g., [1], [2]).

**SmithArnoldi:** *Input* ( $A_p$ ,  $B_p$ , *max\_itr*, *tol*)

$j := 1$ ;

$q_1 := B_p / \|B_p\|_2$ ;  $Q_1 := [q_1]$ ;  $\beta := 1$ ;

$H_1 := [ ]$ ;  $R_1 := [\|B_p\|_2]$ ;

$W_1 := B_p B_p^T$ ;

*While*  $j \leq \text{max\_itr}$ ,

*for*  $i := 1$  *to*  $j$

$h_{ij} := q_i^T A_p q_j$ ;

*end for*

$r_{j+1} := A_p q_j - \sum_{i=1}^j h_{ij} q_i$ ;

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

if  $j > 1$

$$R_j := \left[ \begin{array}{c|c} R_{j-1} & \\ \hline \begin{bmatrix} 0 & \cdots & 0 \end{bmatrix} & H_j \begin{bmatrix} R_{j-1}(:, j-1) \\ 0 \end{bmatrix} \end{array} \right];$$

$$w_j := Q_j R_j(:, j);$$

$$W_j := W_{j-1} + w_j w_j^T;$$

if  $(\|w_j\|_2 < tol)$  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  $W_k$ ,  $Q_k$ ,  $R_k$ , and  $H_k$ .

In short, the Arnoldi algorithm iteratively computes the  $k$  orthogonal columns of  $Q_k \in \mathbb{R}^{n \times k}$ , an upper Hessenberg matrix  $H_k \in \mathbb{R}^{k \times k}$ , an upper triangular matrix  $R_k \in \mathbb{R}^{k \times k}$ , and an accumulation matrix  $W_k \in \mathbb{R}^{n \times n}$  such that

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

### C. Balanced Stochastic Truncation

The positive real lemma [2] states that the system in (1) is passive if and only if there exists a  $P(\in \mathbb{R}^{n \times n}) \geq 0$  satisfying the linear matrix inequality (LMI)

$$\begin{bmatrix} A^T P + PA & PB - C^T \\ B^T P - C & -(D + D^T) \end{bmatrix} \leq 0. \quad (6)$$

Using Schur complement, (6) is equivalent to

$$A^T P + PA + (PB - C^T)(D + D^T)^{-1}(B^T P - C) \leq 0. \quad (7)$$

The solution of (7) being zero is a CARE. Taking the matrix root  $LL^T = (D + D^T)^{-1}$  and defining  $\hat{B} = BL$ ,  $\hat{C} = L^T C$ , and  $\hat{A} = A - B(D + D^T)^{-1}C$ , the CARE is expressible as

$$F(P) = \hat{A}^T P + P\hat{A} + P\hat{B}\hat{B}^T P + \hat{C}^T \hat{C} = 0. \quad (8)$$

The solution of (8), if it exists, is not unique. Among them there is a unique *stabilizing solution*,  $P_\infty$ , in the sense that  $\text{spec}(\hat{A} + \hat{B}\hat{B}^T P_\infty) \subset \mathbb{C}^-$ . The mechanics of BST is to first align (balance) the most reachable states with a given input energy, quantified by  $\int_{-\infty}^0 u(t)^T y(t) dt$ , with the states delivering the maximum energy to the output, quantified by  $-\int_0^\infty u(t)^T y(t) dt$  [9], [12]. It starts by finding the stabilizing solutions,  $P_{min}$  and  $Q_{min}$ , to the following dual CAREs

$$\hat{A}^T P_{min} + P_{min} \hat{A} + P_{min} \hat{B} \hat{B}^T P_{min} + \hat{C}^T \hat{C} = 0 \quad (9a)$$

$$\hat{A} Q_{min} + Q_{min} \hat{A}^T + Q_{min} \hat{C}^T \hat{C} Q_{min} + \hat{B} \hat{B}^T = 0. \quad (9b)$$

Let  $Q_{min} = XX^T$  and  $P_{min} = YY^T$  be any Cholesky factorizations, now do the singular value decomposition (SVD)

$$X^T Y = U \Sigma V^T \quad (10)$$

where  $\Sigma \geq 0$  is an ‘‘economy size’’  $k$ -by- $k$  ( $k \leq n$ ) diagonal matrix with singular values in descending magnitudes. Suppose the singular values of  $\Sigma$  are

$$\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_r \gg \sigma_{r+1} \geq \dots \geq \sigma_k. \quad (11)$$

These values quantify the importance of the states in the input-to-output energy transfer. Accordingly, in MOR, the ‘‘tail’’ can be truncated so only the most significant states remain. To do this, define  $I_m$  to be the identity matrix of dimension  $m$ ,  $0_{m \times n}$  an  $m \times n$  zero matrix, and

$$T_L = \begin{bmatrix} I_r & 0_{r \times (k-r)} \end{bmatrix} \Sigma^{-\frac{1}{2}} V^T Y^T, \quad T_R = X U \Sigma^{-\frac{1}{2}} \begin{bmatrix} I_r \\ 0_{(k-r) \times r} \end{bmatrix}. \quad (12)$$

The system  $(T_L A T_R, T_L B, C T_R, D)$  then represents the stochastically balanced (also referred to as positive-real balanced [12], [20]) and truncated model. The best bound to date for the frequency domain approximation error can be found in [11]. BST is preferred to standard BT because it guarantees passivity, in addition to stability, in the reduced order model (e.g., [9]–[12]).

### III. THE NSCARE ALGORITHM

We start with a brief recap of Newton method in solving CAREs, more details can be found in [22], [28]. Advantages of Newton algorithm include its quadratic convergence (once attained) and high numerical accuracy. Let  $P_j$ ,  $j = 0, 1, \dots$ , be the progressive estimates of the stabilizing solution, we define  $P_{j+1} = P_j + \delta P_j$  where  $\delta P_j$  is the search direction or Newton step. Substituting  $P_{j+1}$  into (8),

$$\begin{aligned} F(P_j + \delta P_j) &= F(P_j) + (\hat{A} + \hat{B}\hat{B}^T P_j)^T \delta P_j \\ &\quad + \delta P_j (\hat{A} + \hat{B}\hat{B}^T P_j) + \delta P_j \hat{B}\hat{B}^T \delta P_j. \end{aligned} \quad (13)$$

Every Newton iteration step is a first order error correction such that the sum of the first three terms on the right of (13) goes to zero, i.e.,

$$(\hat{A} + \hat{B}\hat{B}^T P_j)^T \delta P_j + \delta P_j (\hat{A} + \hat{B}\hat{B}^T P_j) + F(P_j) = 0 \quad (14)$$

which is simply a Lyapunov equation. After each step, (13) is left with a quadratic residual term. So from the second step, i.e.,  $j = 1$ , onwards,

$$F(P_j) = \delta P_{j-1} \hat{B}\hat{B}^T \delta P_{j-1} \quad (15)$$

which is a low rank matrix due to  $\hat{B}$ . For compactness, define the Lyapunov operator  $\mathcal{L}_A : \mathbb{R}^{n \times n} \rightarrow \mathbb{R}^{n \times n}$  as  $\mathcal{L}_A(P) = A^T P + P A$ , so that (14) is rewritten as

$$\mathcal{L}_{\hat{A} + \hat{B}\hat{B}^T P_j}(\delta P_j) + \delta P_{j-1} \hat{B}\hat{B}^T \delta P_{j-1} = 0 \quad (16)$$

for  $j = 1, 2, \dots$ . This reminds us of the Smith method (c.f. (4)) and its associated Krylov matrix factorization. To take full advantage of the low rank input/output matrices, we note that the Smith transformation involves computing the inverse  $(\hat{A} + \hat{B}\hat{B}^T P_j + pI)^{-1}$ . Letting  $S_p = (\hat{A} + \hat{B}\hat{B}^T P_{j-1} + pI)^{-1}$  and using matrix inversion lemma, we get

$$\begin{aligned} &(\hat{A} + \hat{B}\hat{B}^T P_j + pI)^{-1} \\ &= (\hat{A} + \hat{B}\hat{B}^T P_{j-1} + pI + \hat{B}\hat{B}^T \delta P_{j-1})^{-1} \\ &= S_p - S_p \hat{B}(I + \hat{B}^T \delta P_{j-1} S_p \hat{B})^{-1} \hat{B}^T \delta P_{j-1} S_p. \end{aligned} \quad (17)$$

Therefore, if  $S_p$  is precomputed, the right hand side of (17) requires only an  $m \times m$  inversion in subsequent steps provided the same  $p$  is used. To sum up, the NSCARE algorithm that solves (8) is as follows:

**NSCARE: Input** ( $\hat{A}$ ,  $\hat{B}$ ,  $\hat{C}$ ,  $P_0$ , **max\_itr**, **tol**)

Find the shift  $p$  corresponding to  $\hat{A} + \hat{B}\hat{B}^T P_0$ ;

$$T_p := (\hat{A} + \hat{B}\hat{B}^T P_0 - pI);$$

$$S_p := (\hat{A} + \hat{B}\hat{B}^T P_0 + pI)^{-1};$$

Solve for  $\delta P_0$  in  $\mathcal{L}_{\hat{A} + \hat{B}\hat{B}^T P_0}(\delta P_0) + F(P_0) = 0$  with

standard solvers. In particular, when  $P_0 = 0$ , solve

$$\mathcal{L}_{\hat{A}}(\delta P_0) + \hat{C}^T \hat{C} = 0 \text{ using SmithArnoldi algorithm}$$

with input  $((T_p S_p)^T, \sqrt{-2p} S_p^T \hat{C}^T, \text{max\_itr}, \text{tol})$ ;

$$j := 1;$$

While  $j \leq \text{max\_itr}$ ,

$$P_j := P_{j-1} + \delta P_{j-1};$$

$$\Theta := \hat{B}^T \delta P_{j-1};$$

If convergence is slow, update  $p$  by

Find shift  $p$  corresponding to  $\hat{A} + \hat{B}\hat{B}^T P_j$ ;

$$T_p := (\hat{A} + \hat{B}\hat{B}^T P_j - pI);$$

$$S_p := (\hat{A} + \hat{B}\hat{B}^T P_j + pI)^{-1};$$

else use the same shift  $p$  and

$$T_p := T_p + \hat{B}\Theta;$$

$$S_p := S_p - S_p \hat{B}(I + \Theta S_p \hat{B})^{-1} \Theta S_p;$$

end if

Solve for  $\delta P_j$  in  $\mathcal{L}_{\hat{A} + \hat{B}\hat{B}^T P_j}(\delta P_j) + \Theta^T \Theta = 0$

(i.e., (16)) using SmithArnoldi with input

$$((T_p S_p)^T, \sqrt{-2p} S_p^T \Theta^T, \text{max\_itr}, \text{tol});$$

If the Frobenius norm  $\|\delta P_j\|_F < \text{tol}$

$$P_\infty := P_j + \delta P_j;$$

Break while loop and return  $P_\infty$ ;

end if

$$j := j + 1;$$

end while

Convergence analysis of the NSCARE algorithm follows closely from those in [22], [28]. 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 step are non-singular and  $P_j$ ,  $j = 1, 2, \dots$ , are also stabilizing; **ii)**  $0 \leq \dots \leq P_j \leq P_{j+1} \leq P_\infty$ ; **iii)**  $0 \leq \|P_\infty - P_{j+1}\| \leq \gamma \|P_\infty - P_j\|^2$  where  $\gamma$  is a positive constant, i.e., convergence is quadratic once  $P_j$  falls into the region of convergence.

In practice, the tolerance parameter,  $tol$ , is set to a small value such as the machine precision. In the first call to SmithArnoldi (i.e., finding  $\delta P_0$  or  $\delta P_1$ , depending on the initial condition  $P_0$ ), the number of iterations is the highest, and then decreases in subsequent runs once quadratic convergence is acquired. 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 (6) with  $P = I$  (see also [9], [14]). It follows that  $\hat{A}$  is stable and the initial guess of  $P_0 = 0$  is stabilizing. Moreover, since  $0 \leq P_j \leq P_\infty < I$ , we have  $\|P_\infty - P_j\| < 1$ ,  $j = 0, 1, \dots$ . In other words, under the mild assumption of a strictly passive (dissipative) system, quadratic convergence of the NSCARE algorithm is guaranteed.

*Remarks:*

1. In our NSCARE implementation,  $p$  is approximated by first applying a Lanczos algorithm for  $\kappa$  steps on  $\hat{A} + \hat{B}\hat{B}^T P_j$  to obtain a tridiagonal matrix  $T_\kappa \in \mathbb{R}^{\kappa \times \kappa}$ ,  $\kappa \ll n$ , whose eigenvalues closely approximate the extremal eigenvalues of  $\hat{A} + \hat{B}\hat{B}^T P_j$ . Then a simple (inverse) power iteration [21] is used to estimate the magnitude of the maximum (minimum) eigenvalue of  $T_\kappa$  so as to form  $p$ . The initial Lanczos process has  $O(\kappa n^2)$  work and the power iterations require  $O(\kappa^3)$  work.

2. When matrix inversion lemma is used to update  $S_p$ , full rank inversion is bypassed and the work reduces from  $O(n^3)$  to  $O(m^3)$ . Subsequently, the only  $O(n^3)$  steps in NSCARE are the explicit updates of  $S_p$ . In case of sparse matrices wherein the inversion in  $S_p$  can be done with  $O(n^2)$  work, the whole algorithm will further reduce to an  $O(n^2)$  one. In our examples later on, it is found that only one or two shift updates are needed. This is because when  $P_j$  is converging quadratically, norm of  $\delta P_j$  is small and has little effect on  $p$ .

3. Suppose (9a) is solved with NSCARE 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 SmithArnoldi be  $k_1, k_2, \dots, k_N$ , and  $k_T = k_1 + k_2 + \dots + k_N$ . Then in terms of the outputs of SmithArnoldi,  $P_{min} = \sum_{i=1}^N (Q_{k_i} R_{k_i})(Q_{k_i} R_{k_i})^T = (Q_{k_T} R_{k_T})(Q_{k_T} R_{k_T})^T$  where  $Q_{k_T} = \begin{bmatrix} Q_{k_1} & \dots & Q_{k_N} \end{bmatrix}$  and  $R_{k_T} = \text{diag}(R_{k_1}, \dots, R_{k_N})$ . Thus referring to (10), a factor of  $P_{min}$  is given by  $Y = Q_{k_T} R_{k_T}$ . The factor  $X$  of  $Q_{min}$  is obtained similarly. In our experiments,  $N$  is usually less than 10 and  $k_T$  is in the order of tens or hundreds regardless of the CARE size, thus only a medium size SVD is needed even for high order initial models. Consequently,

the NSCARE algorithm also helps to elude two large size matrix factorizations and one large scale SVD required by the original BST implementation.

#### IV. THE PROJECT-AND-BALANCE ALGORITHM

The NSCARE algorithm is suitable for the BST of medium to large systems (say, orders from hundreds to thousands). For even higher orders (thousands to millions), it is advisable to adopt a two-stage, project-and-balance approach. The idea of a stepwise reduction is not new (e.g., [12], [15]). The attention of our paper is on the efficient implementation of such a scheme. In particular, the first projection step is carried out by the fast Smith method as in NSCARE. And more importantly, in the second BST step, we introduce an innovative way of simultaneously solving the dual CAREs at the cost of essentially one. The step adopts a Hamiltonian approach that does not rely on low rank input/output matrices. As a result, it is by itself an attractive scheme for BST of systems with a large number of input/output ports.

##### A. Eigenspace Projection

The first stage of reduction is to select an appropriate subspace onto which the original high order system is projected. It is therefore well-justified to use the (approximate) spans of  $W_c$  and  $W_o$  as they capture (nearly) all state activities. This idea appeared as *dominant subspaces projection* in [8] and also as *dominant Gramian eigenspaces method* in [17]. The Smith method together with Krylov iteration are attractive candidates for the task. For example, the SmithArnoldi algorithm can be used for extracting the span of, say,  $W_c$  in (4). Suppose the algorithm converges in  $\tau$  steps rendering  $\mathcal{K}_\tau(A_p, B_p) = Q_\tau R_\tau$ , it is obvious that  $Q_\tau$  spans the column range of  $W_c$ . A counterpart,  $Q_v$ , corresponding to the column range of  $W_o$  is obtained similarly. A Gram-Schmidt (GS) orthogonalization of  $Q_v$  against  $Q_\tau$  (columns in  $Q_\tau$  are already orthogonal) produces an orthogonal  $Q_k = GS(\begin{bmatrix} Q_\tau & Q_v \end{bmatrix}) \in \mathbb{R}^{n \times k}$ ,  $k \leq \tau + v$ , which can be taken as the projection basis to generate an intermediate model of order  $k$ . Referring to (6), RLC models obtained from MNA have the properties  $A + A^T \leq 0$ ,  $B = C^T$ , and  $D = 0$  [14]. Passivity of the circuit is then borne out by the fact that  $P = I$  is a solution satisfying (6). Performing a congruence transformation of compatible dimensions, we have

$$\begin{bmatrix} Q_k^T & 0 \\ 0 & I \end{bmatrix} \begin{bmatrix} A + A^T & B - C^T \\ B^T - C & 0 \end{bmatrix} \begin{bmatrix} Q_k & 0 \\ 0 & I \end{bmatrix} \leq 0. \quad (18)$$

It is easily seen that the system  $(Q_k^T A Q_k, Q_k^T B, C Q_k, 0)$  inherits passivity from its parent. Using techniques in [9], [30], this system with a zero  $D$  matrix can be transformed into an equivalent one with  $D + D^T > 0$ , from which the CAREs for BST can be derived.

### B. Solving Dual CAREs

The intermediate passive model from projection is then subject to BST to achieve further reduction with guaranteed passivity. Many algorithms exist for solving a CARE via identifying the stable invariant subspace of an associated Hamiltonian matrix, (e.g., [22], [25], [31]). While this is sufficient for the stabilizing solution, information about the unstable invariant subspace is just a few steps away and not utilized. We show that with slight extra effort, the stable and unstable invariant subspaces can be completely separated, which in turn enables joint solution of the dual CAREs in (9). Consider the Hamiltonian matrices,  $H$  and  $H'$ , corresponding to (9a) and (9b) respectively:

$$H = \begin{bmatrix} \hat{A} & \hat{B}\hat{B}^T \\ -\hat{C}^T\hat{C} & -\hat{A}^T \end{bmatrix}, H' = \begin{bmatrix} \hat{A}^T & \hat{C}^T\hat{C} \\ -\hat{B}\hat{B}^T & -\hat{A} \end{bmatrix}. \quad (19)$$

If  $\lambda$  is an eigenvalue of a Hamiltonian matrix, then so is  $-\lambda$ . Since  $H$  and  $H'$  are real, eigenvalues apart from the real and imaginary axes occur even in quadruple  $(\lambda, -\lambda, \bar{\lambda}, -\bar{\lambda})$ . By our assumption of a minimal passive system,  $H$  has no eigenvalues on the imaginary axis, and the stable and unstable invariant subspaces can be decoupled, namely,

$$H \begin{bmatrix} X_{11} & X_{12} \\ X_{21} & X_{22} \end{bmatrix} = \begin{bmatrix} X_{11} & X_{12} \\ X_{21} & X_{22} \end{bmatrix} \begin{bmatrix} \Lambda_s & 0 \\ 0 & \Lambda_u \end{bmatrix} \quad (20)$$

where  $\Lambda_s$  contains the stable eigenvalues and  $\Lambda_u$  the unstable ones. A well-known fact is that  $X_{11}$  is invertible and  $P_{min} = X_{21}X_{11}^{-1} \geq 0$ . A key observation is that from  $H' = \begin{bmatrix} 0 & I \\ I & 0 \end{bmatrix} (-H) \begin{bmatrix} 0 & I \\ I & 0 \end{bmatrix}$ , we also get  $Q_{min} = X_{12}X_{22}^{-1} \geq 0$ . In other words, all information about  $P_{min}$  and  $Q_{min}$  are contained in (20). Decoupling of the invariant subspaces can be achieved by standard ways (e.g., Grassmann manifolds [32]). Nonetheless, for completeness of this paper and interested practitioners, we propose a fast converging quadruple-shift bulge-chasing SR algorithm for completely separating the stable and unstable invariant subspaces of a Hamiltonian matrix. Implementation details of this special SR algorithm are given in the appendices.

As in NSCARE, the most expensive step in this two-stage algorithm is the full matrix inversion ( $O(n^3)$  work) in computing  $A_p$  in the first-stage Smith transformation. Apart from this, all other operations in the projection stage are at most  $O(n^2)$ . Intermediate model in the second-stage BST is much smaller ( $k \ll n$ ) and imposes minor burden on the overall cost. It should be stressed that the subspace separation technique here is independent of the projection applied in Section IV-A. As a result, it can also be employed in direct BST to approximately double the speed of solving dual CAREs.

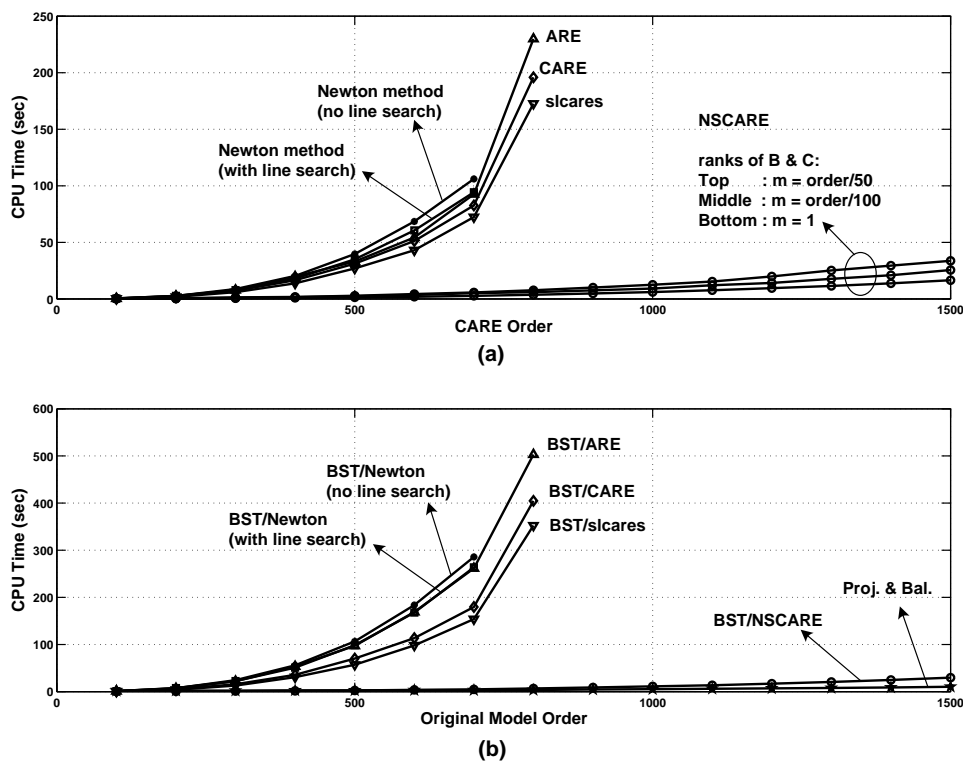


Fig. 1. (a) CPU time for solving a single CARE. (b) CPU time for doing model order reduction.

## V. NUMERICAL EXAMPLES

All numerical experiments in this section were carried out in the MATLAB R14 environment on a 3GHz, 3G-RAM PC with many applications running. The NSCARE algorithm, the project-and-balance algorithm together with the quadruple-shift SR procedure in the appendices, as well as the Newton method with and without line search [28] were coded and executed as MATLAB script (text) files. For comparison we also employed the MATLAB CARE solver routines *ARE* and *CARE* that implement the Hamiltonian-based Schur vector and eigenvalue methods, respectively (see [22] and references therein). In addition, the prebuilt FORTRAN 77 routine *slcares*, a numerically reliable and efficient implementation of the Schur vector method, was invoked from the SLICOT library [27] via a MATLAB gateway.

First, we study randomly generated strictly passive systems, with  $A + A^T < 0$  and rank-one  $B$  and  $C$ , satisfying the settings in (1). The CARE in (9a) was formed and solved by various solvers, as depicted in Fig. 1(a). For fairness, solutions from NSCARE were computed to the same or better accuracy than

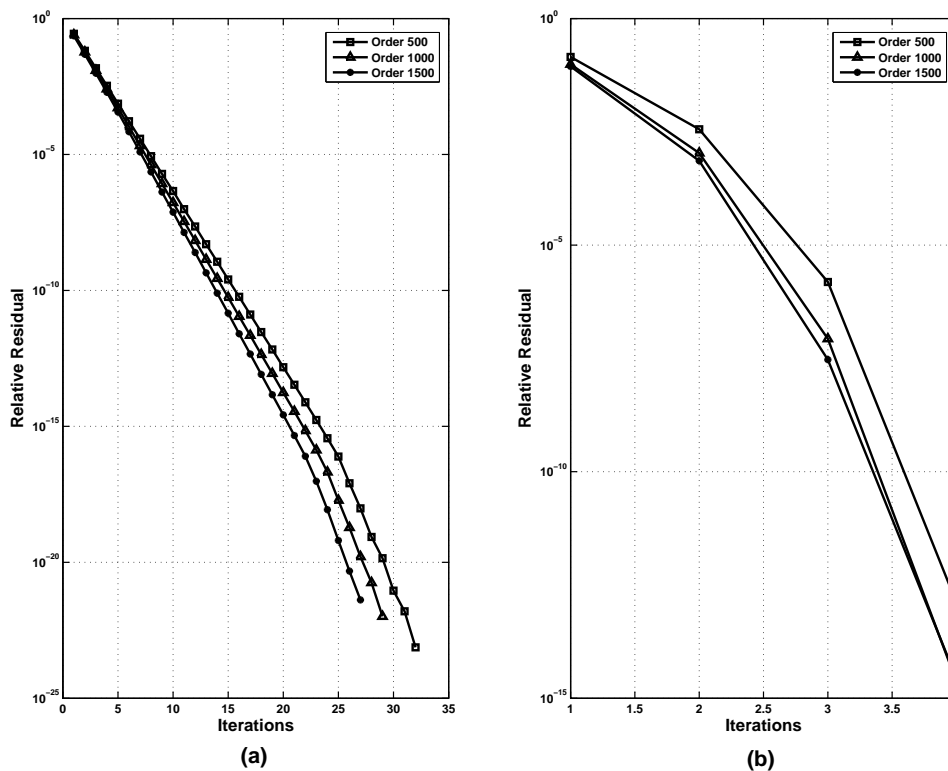


Fig. 2. (a) Superlinear convergence of iterates in SmithArnoldi algorithm and (b) quadratic convergence of iterates in NSCARE, at several CARE orders.

those by other solvers. Though all these solver algorithms are of  $O(n^3)$  complexity, they behave quite differently. In particular, NSCARE easily handles orders as high as 1500, while others require intensive computation well below 1000. Perhaps more importantly, NSCARE scales favorably with increasing model order, e.g., at 500, it is at least 20 times faster, and 50 times faster at 800. To investigate the effect of increasing ranks in  $B$  and  $C$ , Fig. 1(a) also includes the curves whereby the ranks of  $B$  and  $C$  equal the CARE order divided by 100 and 50, respectively. Block Arnoldi algorithms were used in these cases. It is seen that the growth in computation is relatively mild and is more obvious at higher orders. In practical systems, however, the ranks of  $B$  and  $C$  (associated with number of input/output ports) usually remain constant and seldom grow with model order. This justifies the use of NSCARE whenever ranks of  $B$  and  $C$  are low. Fig. 2(b) shows the convergence of the NSCARE iterates at several CARE orders, wherein the relative residual  $\|P_j - P_{min}\|_F / \|P_{min}\|_F$  ( $\|\circ\|_F$  being the Frobenius norm of a matrix) is

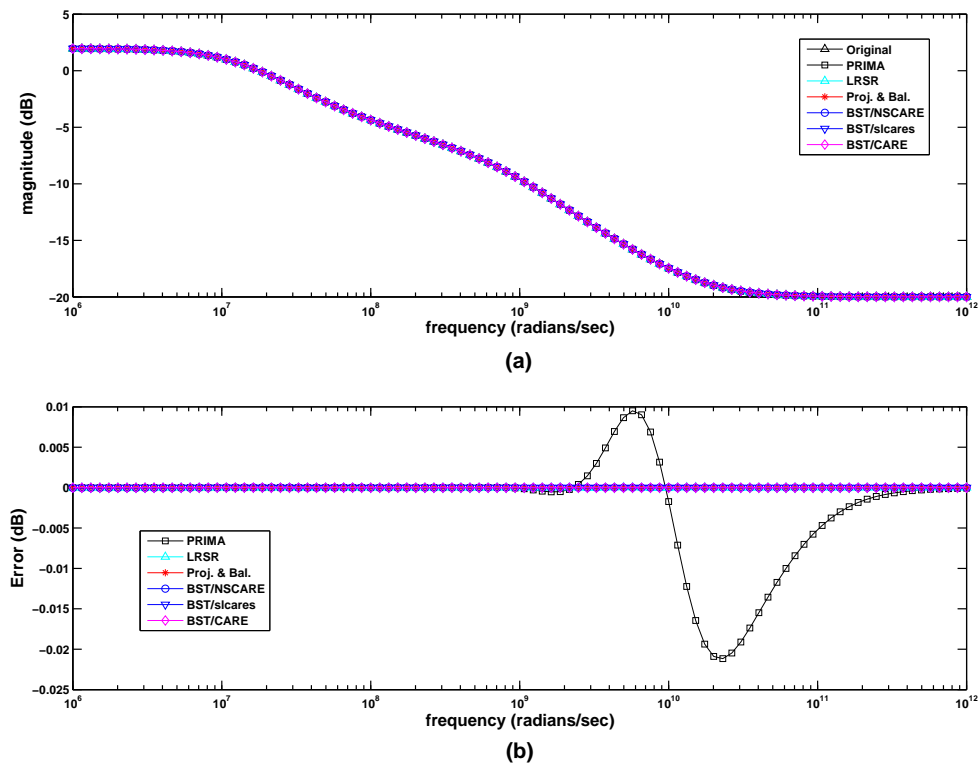


Fig. 3. (a) Frequency responses of the spiral inductor and reduced order models. (b) Deviation from the original response.

plotted. Expectedly, a quadratic rate is observed since NSCARE is simply an efficient implementation of the Newton method. Fig. 2(a) plots the convergence of the iterates in SmithArnoldi algorithm when solving for  $\delta P_0$  in the first step of NSCARE (that for  $\delta P_j$ s are similar). The rate is superlinear which is again expected because Smith method, being a special case of ADI, inherits the superlinear convergence of the latter.

Fig. 1(b) plots the CPU time for realizing BST and the project-and-balance algorithm. Rank-one  $B$  and  $C$  are used. BST requires solving two CAREs plus matrix factorizations, so the time curves corresponding to different solvers, compared with their counterparts in Fig. 1(a), are generally more than double. As stated in the remarks in Section III, for models with low rank input/output matrices, BST/NSCARE has the additional advantage of eluding large scale matrix factorizations and SVDs. On the other hand, the project-and-balance algorithm employed an intermediate model of order about 50 at all CARE orders. Its curve corresponds to the sum of projection time and BST time, in contrast to the direct BST in

TABLE I

CPU TIMES AND REDUCED MODEL ORDERS (TIME / ORDER) FOR VARIOUS SCHEMES WITH THE TIME OF PRIMA  
NORMALIZED TO ONE.

	PRIMA	LRSR	Proj. & Bal.	BST/NSCARE	BST/slcares	BST/CARE
Inductor		(1.76)	(1.58)	(7.65)	(66.67)	(79.94)
Order=500		(0.02)	(1.00)	(0.73)	(6.04)	(5.84)
	1 / 10	1.78 / 9	2.58 / 10	8.38 / 10	72.71 / 11	85.78 / 11
Wire 1		(1.19)	(5.05)	(7.31)	(107.75)	(117.33)
Order=1000		(0.003)	(0.15)	(0.003)	(6.87)	(7.23)
	1 / 5	1.19 / 6	5.20 / 8	7.31 / 5	114.62 / 5	124.56 / 8
Wire 2		(1.26)	(4.83)	(7.41)	(103.53)	(115.45)
Order=1000		(0.003)	(0.12)	(0.006)	(6.82)	(6.17)
	1 / 6	1.26 / 8	4.95 / 9	7.42 / 6	110.35 / 6	121.62 / 9

Note:

1. The upper bracket in Proj. & Bal. is the time for projection and the lower bracket is the time for BST.
2. The upper bracket in the direct BST (LRSR) is the time for solving dual CAREs (Lyapunov equations) and the lower bracket is the time for matrix factorizations.

other curves. As the intermediate model order remains almost constant, the time for BST (using the proposed SR algorithm) is independent of the original model order. From the figure, it can be seen that NSCARE and project-and-balance algorithm consume similar resources and are superior to conventional BST implementations.

Next, we apply NSCARE and the project-and-balance scheme towards some large scale MOR problems. The first example results from the extraction of an on-chip planar square spiral inductor suspended over a copper plane [23]. The initial model of order 500 is reduced using different schemes including PRIMA [1], low rank square root (LRSR) [13], [23], the proposed project-and-balance algorithm [19], BST with NSCARE [18], and BST with conventional solvers. The PRIMA model is set to the same order as that from BST/NSCARE. The frequency response and error plots are shown in Fig. 3, and the CPU times and final model orders are tabulated in Table I. It should be noted that if the Hamiltonian solver routines (in this case *slcares* and *CARE*) could be modified to implement the complete subspace separation of Section IV-B, the time in solving the dual CAREs would have been approximately halved, but even so BST/NSCARE is still more efficient due to exploitation of low rank  $B$  and  $C$ . From the figure and table, with comparable model order, PRIMA exhibits a bigger mismatch over the frequency

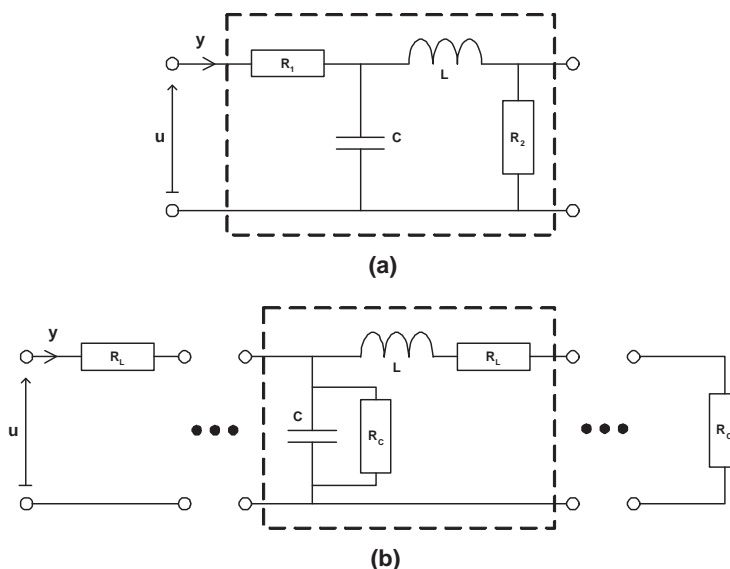


Fig. 4. (a) One RLC section of the wire modeling. (b) Another RLC modeling. For simplicity the following values are used in all sections:  $R_1 = R_L = 0.1$ ,  $R_2 = R_C = 1.0$ ,  $C = 0.1$  and  $L = 0.1$ .

axis. Models from BST, unsurprisingly, tend to have better global accuracy [12], [19]. In our two-stage project-and-balance implementation, the initial model was first reduced to an intermediate model of order 100 (same in the next two examples), followed by BST using the proposed SR algorithm. The excellent accuracy can be attributed to the effectiveness of dominant eigenspace projection in capturing most state activities, as has been observed in [17]. Another possible reason is the better numerical conditioning in a stepwise reduction.

The second example is the simulation of a wire model with 500 repeated RLC sections in Fig. 4(a), producing a model of order 1000 [20]. For simplicity identical sections are used. The input and output are taken as the voltage and current into the first section, respectively. The reduction results are shown in Fig. 5 and Table I. As before, PRIMA is less accurate in approximating the original response, while models from other schemes have almost indistinguishable responses from the original. The third example, in Fig. 4(b), depicts another wire modeling wherein the center loop is repeatedly inserted to generate a model of order 1000. Again, similar observations can be drawn from Fig. 6 and Table I. To this end, a few technical details are worth mentioning:

- Runtimes of BST/NSCARE and project-and-balance algorithm are comparable to “fast” algorithms such as PRIMA and LRSR. In fact, BST/NSCARE is about an order faster than conventional BST

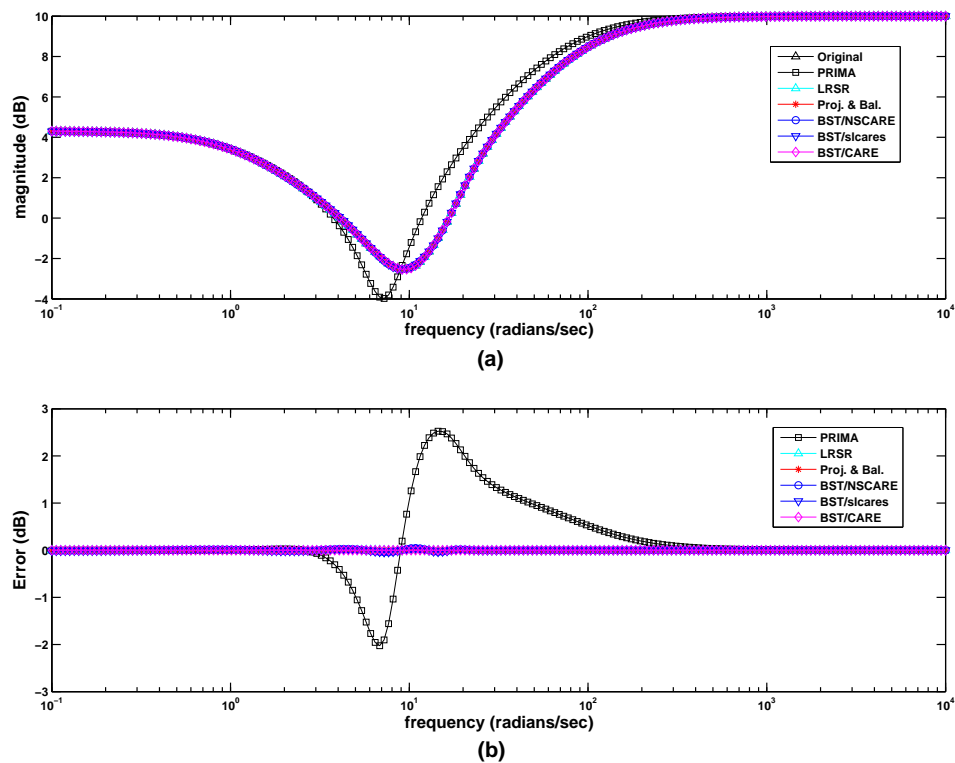


Fig. 5. (a) Frequency responses of the first wire model and its reduced order models. (b) Deviation from the original response.

realizations.

- LRSR does not guarantee passivity of the reduced state space model. Passivity test and enforcement may be needed before the reduced model is connected for global simulation.
- Projection-type algorithms like PRIMA and the first stage of project-and-balance algorithm require the initial (passive) state space to be in a certain form [2], [12] (essentially for (18) to hold) and this is not always convenient or feasible. BST, on the other hand, poses no constraints on the internal structure of the state space model.
- BST avoids the the selection of expansion points and final model order as in PRIMA which would involve *a priori* knowledge of the original response.
- From our experiments it is seen that reduced order models from BST/NSCARE generally has a lower order for the same accuracy.

To summarize, both NSCARE and project-and-balance algorithm are important candidates in large

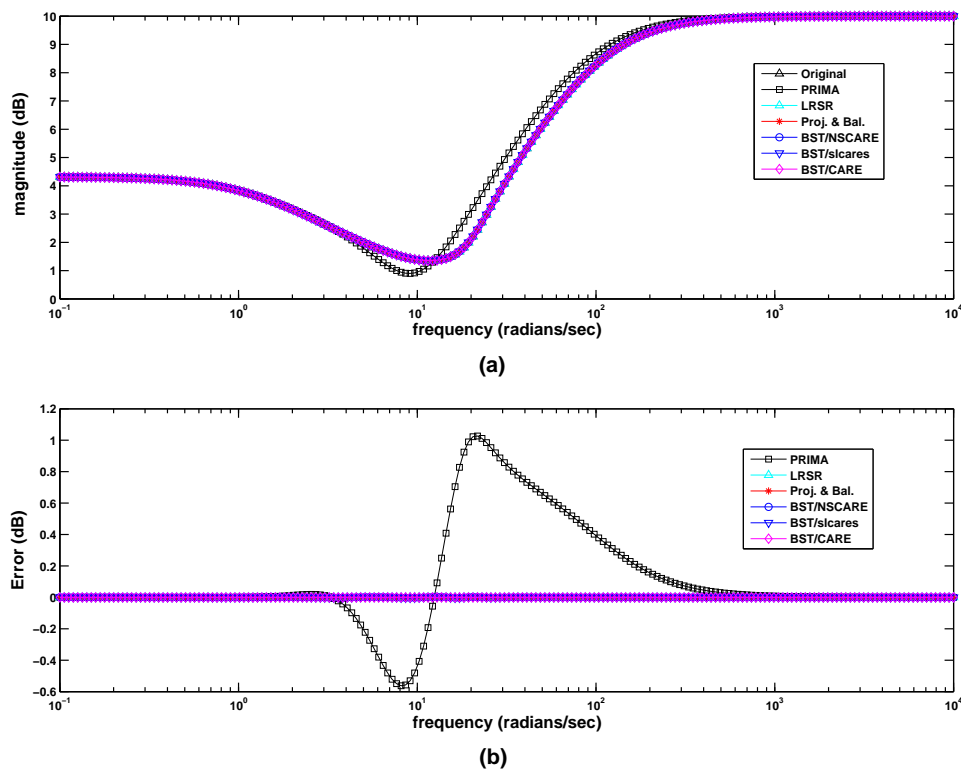


Fig. 6. (a) Frequency responses of the second wire model and its reduced order models. (b) Deviation from the original response.

scale reduction problems. These control-theoretic approaches generally produce reduced order models of high global accuracy. The algorithms can be applied individually or together to enable the deployment of previously impractical large scale passivity-preserving MOR. As a general guideline, for very high model order (millions), projection-type method is used to bring the order down to thousands or hundreds. It is then followed by BST to further compress the order down to hundreds or tens. When the ranks of input/output matrices are low, NSCARE provides a fast means for implementing BST. When there are a large number of input/output ports, NSCARE may not be advantageous and solution of dual CAREs through subspace separation may be considered due to its independence on the number of ports. Preservation of passivity, in addition to stability, in the reduced order models is guaranteed throughout the process.

## VI. CONCLUSION

This paper has described two algorithms for fast and accurate passivity-preserving model order reduction. The first algorithm is a Newton method variant based on Smith and Krylov methods, called NSCARE, that exploits low rank input/output matrices to quickly solve a CARE. It can help avoid two large size matrix factorizations and one large size SVD in traditional BST. The second algorithm is an efficient implementation of a two-stage, project-and-balance reduction procedure. The first stage consists of dominant eigenspace projection, again, using the fast Smith method. Moreover, applying the novel idea of separating the stable and unstable invariant subspaces of a Hamiltonian matrix, the second stage solves two dual CAREs in BST at the cost of slightly more than one. An effective quadruple-shift SR algorithm has also been introduced for the operation. The proposed techniques can be applied individually or together. Numerical examples have confirmed their computational efficiency and excellent reduction accuracy over conventional realizations.

## ACKNOWLEDGMENTS

The authors are grateful to the Associate Editor and anonymous reviewers for their helpful and constructive comments, particularly with the experimental part of a previous manuscript.

## REFERENCES

- [1] 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.
- [2] Z. Bai, P. M. Dewilde, and R. W. Freund, "Reduced-order modeling," *Numerical Analysis Manuscript No. 02-4-13, Bell Laboratories*, Mar. 2002.
- [3] L. T. Pillage and R. A. Rohrer, "Asymptotic waveform evaluation for timing analysis," *IEEE Trans. Computer-Aided Design*, vol. 9, no. 4, pp. 352–366, Apr. 1990.
- [4] 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.
- [5] —, "Reduced-order modeling of large linear subcircuits via a block Lanczos algorithm," in *Proc. ACM/IEEE Design, Automation Conf.*, June 1995, pp. 474–479.
- [6] K. Glover, "All optimal Hankel-norm approximation of linear multivariable systems and their  $L^\infty$ -error bounds," *Int. J. Control*, vol. 39, no. 6, pp. 1115–1193, June 1984.
- [7] 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.
- [8] T. Penzl, "Algorithms for model reduction of large dynamical systems," Sonderforschungsbereich 393 Numerische Simulation auf massiv parallelen Rechnern, TU Chemnitz, 09107 Chemnitz, FRG, Tech. Rep. SFB393/99-40, 1999.
- [9] Q. Su, "Algorithms for model reduction of large scale RLC systems," Ph.D. dissertation, School of ECE, Purdue University, Aug. 2002.

- [10] M. Green, "Balanced stochastic realizations," *Linear Algebra Appl.*, vol. 98, pp. 211–247, 1988.
- [11] X. Chen and J. T. Wen, "Positive realness preserving model reduction with  $H_\infty$  norm error bounds," *IEEE Trans. Circuits Syst. I*, vol. 42, no. 1, pp. 23–29, Jan. 1995.
- [12] 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.
- [13] 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.
- [14] —, "A factorization-based framework for passivity-preserving model reduction of RLC systems," in *Proc. IEEE Design, Automation Conf.*, June 2002, pp. 40–45.
- [15] J. Li and J. White, "Efficient model reduction of interconnect via approximate system gramians," in *Proc. IEEE Int. Conf. Computer-Aided Design*, Nov. 1999, pp. 380–383.
- [16] T. Penzl, "A cyclic low rank Smith method for large sparse Lyapunov equations with applications in model reduction and optimal control," *SIAM J. SCI. COMPUT.*, vol. 21, no. 4, pp. 1401–1418, 2000.
- [17] J. Li, "Model reduction of large linear systems via low rank system gramians," Ph.D. dissertation, Department of Mathematics, Massachusetts Institute of Technology, Sept. 2000.
- [18] N. Wong, V. Balakrishnan, C.-K. Koh, and T. S. Ng, "A fast Newton/Smith algorithm for solving algebraic Riccati equations and its application in model order reduction," in *Proc. IEEE Conf. Acoustics, Speech, and Signal Processing*, May 2004, pp. 53–56.
- [19] N. Wong, V. Balakrishnan, and C.-K. Koh, "Passivity-preserving model reduction via a computationally efficient project-and-balance scheme," in *Proc. IEEE Design, Automation Conf.*, June 2004, pp. 369–374.
- [20] S. Gugercin and A. C. Antoulas, "A survey of model reduction by balanced truncation and some new results," *Int. J. Control*, vol. 77, no. 8, pp. 748–766, 2004.
- [21] G. Golub and C. V. Loan, *Matrix Computations*, 2nd ed. Baltimore: JohnsHopkins Univ. Press, 1989.
- [22] 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.
- [23] J. Li and J. White, "Low-rank solution of Lyapunov equations," *SIAM Review*, vol. 46, no. 4, pp. 693–713, 2004.
- [24] 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.
- [25] 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.
- [26] P. Benner, V. Mehrmann, and H. Xu, "A new method for computing the stable invariant subspace of a real Hamiltonian matrix," *J. Comput. Appl. Math.*, vol. 86, no. 1, pp. 17–43, Nov. 1997.
- [27] P. Benner, V. Mehrmann, V. Sima, S. Van-Huffel, and A. Varga, "SLICOT—a subroutine library in systems and control theory," *Appl. and Comput. Contr., Signals, and Circuits*, vol. 1, pp. 499–539, 1999.
- [28] 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.
- [29] J. Vlach and K. Singhal, *Computer Methods for Circuit Analysis and Design*. Kluwer Academic Publishers, July 1993.
- [30] H. Weiss, Q. Wang, and J. L. Speyer, "System characterization of positive real conditions," *IEEE Trans. Automat. Contr.*, vol. 39, no. 3, pp. 540–544, Mar. 1994.

- [31] P. Benner and H. Faßbender, “An implicitly restarted symplectic Lanczos method for the Hamiltonian eigenvalue problem,” *Linear Algebra and its Applications*, vol. 263, pp. 75–111, 1997.
- [32] P.-A. Absil, R. Mahony, and R. Sepulchre, “Riemannian geometry of Grassmann manifolds with a view on algorithmic computation,” *Acta Appl. Math.*, vol. 80, no. 2, pp. 199–220, Jan. 2004.
- [33] A. Bunse-Gerstner and V. Mehrmann, “A symplectic QR like algorithm for the solution of the real algebraic Riccati equation,” *IEEE Trans. Automat. Contr.*, vol. 31, no. 12, pp. 1104–1113, Dec. 1986.



- *Algorithm G – Gaussian Elimination*

$$G(i, v) = \begin{bmatrix} \Theta & \Phi \\ 0 & \Theta^{-1} \end{bmatrix}, G(i, v)^{-1} = \begin{bmatrix} \Theta^{-1} & -\Phi \\ 0 & \Theta \end{bmatrix}. \quad (24)$$

Here  $\Theta = I_k + ((1+v^2)^{-1/4} - 1)(e_{i-1}e_{i-1}^T + e_i e_i^T)$  and  $\Phi = (v(1+v^2)^{-1/4})(e_{i-1}e_i^T + e_i e_{i-1}^T)$ . Algorithm G zeroes a single entry in the upper half of a column of the Hamiltonian matrix when  $y_{k+i} = 0$  (Algorithm J does not work) and  $y_{k+i-1} \neq 0$ . Given  $i$ ,  $2 \leq i \leq k$ ,  $x \in \mathbb{R}^{2k}$ , we have  $G(i, v)x = y$  where  $y_i = 0$ .

Algorithm J and Algorithm H use orthogonal symplectic matrices, while Algorithm G uses a nonorthogonal symplectic matrix of condition number  $\text{cond}_2(G(i, v)) = (1 + v^2)^{1/2} + |v|$ .

- *Implicit Quadruple-Shift SR Algorithm*

As in modern implementations of the QR algorithm [21], the SR counterpart utilizes *Implicit S* bulge-chasing such that all computations are in the real domain. Single- and double-shift strategies are investigated in the technical report version of [31], in which the shifts are chosen from the real and imaginary axes only. Our implementation waives this constraint, and complies better with the quadruple occurrence of eigenvalues away from the axes. A proven heuristic to speed up convergence is to choose the four shifts as eigenvalues of the  $4 \times 4$  subblock (c.f. (21))

$$N_j = \left[ \begin{array}{cc|cc} a_j & 0 & c_j & b_j \\ 0 & a_{j+1} & b_j & c_{j+1} \\ \hline q_j & 0 & -a_j & 0 \\ 0 & q_{j+1} & 0 & -a_{j+1} \end{array} \right] \quad (25)$$

where  $j = k - 1$  in the first iteration, and gradually decreases when the J-tridiagonal matrix deflates [21], [33]. Defining the expression  $\alpha_j = a_j^2 + c_j q_j$ , the characteristic polynomial of (25) is found to be

$$s^4 - (\alpha_j + \alpha_{j+1})s^2 + \alpha_j \alpha_{j+1} - b_j^2 q_j q_{j+1} = 0. \quad (26)$$

The roots of (26) are used as shifts. Analogous to the Implicit Q theorem in QR algorithm, the first column of the following matrix product is required for Implicit S similarity transform

$$\begin{aligned} p(\lambda) &= (H - \lambda I)(H + \lambda I)(H - \bar{\lambda} I)(H + \bar{\lambda} I) \\ &= H^4 - 2\text{Re}(\lambda^2)H^2 + |\lambda|^4 I \\ &= H^4 - (\alpha_j + \alpha_{j+1})H^2 + (\alpha_j \alpha_{j+1} - b_j^2 q_j q_{j+1})I. \end{aligned} \quad (27)$$

Reusing the definition of  $\alpha_j$  and a MATLAB-style representation, the first columns of  $H^2$  and  $H^4$  are

$$H^2(:, 1) = \begin{bmatrix} \alpha_1 \\ b_1 q_1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}, H^4(:, 1) = \begin{bmatrix} \alpha_1^2 + b_1^2 q_1 q_2 \\ b_1 q_1 (\alpha_1 + \alpha_2) \\ b_1 q_1 b_2 q_2 \\ 0 \\ \vdots \\ 0 \end{bmatrix}. \quad (28)$$

Setting  $H_1 := H$  and using Algorithm H to find an  $H(1, 3, w)$  such that  $H(1, 3, w)p(\lambda)e_1$  is a multiple of  $e_1$ , the bulge-chasing begins by forming  $H_2 := H(1, 3, w)H_1H(1, 3, w)^T$  and  $\Pi := H(1, 3, w)^T$ . An example  $H_2 \in \mathbb{R}^{12 \times 12}$  looks like

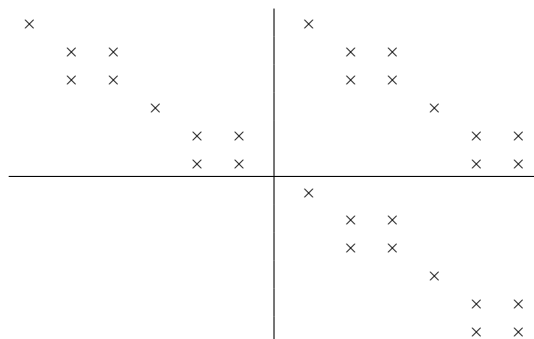
$$\begin{array}{ccc|ccc} \times & \times & \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times & \times & \times \\ & & & \times & & & \\ & & & & \times & & \\ & & & & & \times & \times \\ & & & & & & \times \\ \hline \times & \times & \times & \times & \times & \times & \\ \times & \times & \times & \times & \times & \times & \\ \times & \times & \times & \times & \times & \times & \\ & & & \times & & & \\ & & & & \times & & \\ & & & & & \times & \\ & & & & & & \times \end{array}$$

To restore the J-tridiagonal structure, we refer to the following matrix. Circles represent zeroing of entries and asterisks stand for newly generated entries. The (9, 1) entry is zeroed using  $H_3 := J(3, c, s)H_2J(3, c, s)^T$  and the update  $\Pi := \Pi J(3, c, s)^T$ . Entry at (7, 3) is automatically zeroed due to the (Hamiltonian) structure-preserving symplectic transform. Similarly, Algorithm J is used to zero (8, 1). Then (3, 1) is zeroed by Algorithm H with  $H(2, 2, w)$ , followed by Algorithm G for (2, 1). Next, on the right half, (9, 7) and (8, 7) are zeroed by two times of Algorithm J, and on the upper right quadrant, (4, 7) and (3, 7) are zeroed with Algorithm H with  $H(2, 3, w)$ . Consequently, the bulge is pushed to the lower right and the process is continued until it is completely driven out.

$$\begin{array}{ccc|ccc} \times & \otimes & \otimes & \times & \times & \otimes & \otimes \\ \otimes & \times & \times & * & \times & \times & \times & * \\ \otimes & \times & \times & * & \otimes & \times & \times & * \\ & & * & * & \times & \times & \times & \times \\ & & & & \times & * & * & \times & \times \\ & & & & & & \times & \times & \times \\ \hline \times & \otimes & \otimes & \times & \otimes & \otimes & & & \\ \otimes & \times & \times & * & \otimes & \times & \times & * & \\ \otimes & \times & \times & * & \otimes & \times & \times & * & \\ & & * & * & \times & * & * & \times & \\ & & & & \times & & & \times & \\ & & & & & & & & \times \end{array}$$

As iteration proceeds, some of the  $b_j$ s become negligibly small and the problem size deflates as in the QR algorithm. Ultimately, the SR algorithm reduces the J-tridiagonal matrix into decoupled  $2 \times 2$  and  $4 \times 4$  subblocks. Stability and passivity of the intermediate model (Section IV-A) implies the absence of purely imaginary eigenvalues. Using the procedures in [33], the  $2 \times 2$  ( $4 \times 4$ ) subblock can then be transformed into an upper (block) triangular form with the upper left (block) entry containing the eigenvalue(s) with negative real part(s). The example below shows an upper block triangular form with interleaved  $2 \times 2$  and  $4 \times 4$  subblocks (lifted into appropriate planes). The subblocks in the upper left (lower right) quadrant

contain all the stable (unstable) eigenvalues. Appendix II studies zeroing of the upper right quadrant to eventually arrive at (20).



*Remarks:*

1. An alternative to perform the projection in the project-and-balance algorithm is by an implicitly restarted Lanczos algorithm [31]. In that case,  $H$  is readily in J-tridiagonal form, but the projection basis may not be as good as the dominant eigenspace in capturing state transitions.

2. The *JHESS* algorithm in [33] can be used to transform a Hamiltonian matrix into J-tridiagonal form. Existence of this transformation is strongly dependent on the first column of the similarity transform matrix [31]. The set of these breakdown-free vectors is dense in  $\mathbb{R}^{2k}$ . Should breakdown (or near-breakdown) occurs due to high condition numbers in Algorithm G, a different projection basis  $Q_k$  in Section IV-A is chosen by varying the order and/or number of columns in  $Q_\tau$  and  $Q_\nu$ . If the implicitly restarted Lanczos algorithm is used [31], then it is a simple matter of invoking an implicit restart.

3. Convergence of the quadruple-shift SR algorithm is excellent (usually within 10 iterations) under mild conditions. In the few cases where Algorithm G produces a very large condition number (only during early iterates), an exceptional shift is performed and the process is continued [33]. In BST of the intermediate model, the transformation to J-tridiagonal form requires  $O(k^3)$  work (not required in implicitly restarted Lanczos), while that of the SR algorithm is  $O(k^2)$ . As mentioned,  $k \ll n$  and the cost of the second stage of the project-and-balance algorithm is insignificant.

## APPENDIX II

### SEPARATION OF INVARIANT SUBSPACES

We introduce additional symplectic transforms for each type of subblock, at a small cost, to completely separate the stable and unstable invariant subspaces. This brings about solution of the dual CAREs in (9) at the complexity of practically one.

- $2 \times 2$  Subblock

Let  $N_j$  be an ordered subblock taken from the  $j, k + j$  plane of the  $2k \times 2k$  matrix:

$$N_j = \begin{bmatrix} -\lambda_j & x_j \\ 0 & \lambda_j \end{bmatrix} \quad (29)$$

where  $-\lambda_j < 0$  and  $x_j$  is non-zero (otherwise no processing is required). Define the  $2 \times 2$  symplectic matrix

$$T_j = \begin{bmatrix} 1/2\lambda_j & x_j \\ 0 & 2\lambda_j \end{bmatrix}, \quad (30)$$

it is easy to verify that  $T_j^{-1}N_jT_j$  gives the diagonal matrix  $diag(-\lambda_j, \lambda_j)$ . Lifting  $T_j$  into the  $j, k + j$  plane and updating  $\Pi$  completes the subspace separation in this subblock.

- $4 \times 4$  Subblock

Let  $N_j$  be an ordered subblock taken from the  $j, j + 1, k + j, k + j + 1$  plane of the  $2k \times 2k$  matrix:

$$N_j = \begin{bmatrix} \Delta_j & \Omega_j \\ 0 & -\Delta_j^T \end{bmatrix} \quad (31)$$

where  $\Delta_j, \Omega_j (= \Omega_j^T)$  are  $2 \times 2$  matrices. Assume  $\Delta_j$  contains the stable eigenvalues  $-\lambda_j, -\bar{\lambda}_j$  whose real parts are negative. The key to separating the subspaces is to realize that the column range of  $U_j = (N_j + \lambda_j I)(N_j + \bar{\lambda}_j I)$  spans the unstable invariant subspace. Simple manipulation shows

$$span(U_j) = span \left( \begin{bmatrix} \Delta_j \Omega_j - \Omega_j \Delta_j^T + 2 \operatorname{Re}(\lambda) \Omega_j \\ -4 \operatorname{Re}(\lambda) \Delta_j^T \end{bmatrix} \right). \quad (32)$$

On the right hand side of (32), denoting the upper partition by  $Z_1$  and the lower partition by  $Z_2$ , we define

$$F_j = \begin{bmatrix} Z_2^{-T} & Z_1 \\ 0 & Z_2 \end{bmatrix}. \quad (33)$$

It is easy to see that  $F_j$  is well defined ( $Z_2$  invertible) and symplectic. Moreover,  $F_j^{-1}N_jF_j$  gives  $diag(\Delta_j, -\Delta_j^T)$ . Lifting  $F_j$  into the  $j, j + 1, k + j, k + j + 1$  plane and updating  $\Pi$  completes the subspace separation in this subblock. Finally, we have  $H\Pi = \Pi diag(\Lambda_s, -\Lambda_s^T)$ , and solutions to the dual CAREs can be extracted from  $\Pi$  as in (20).