Fast Positive-Real Balanced Truncation Via Quadratic Alternating Direction Implicit Iteration

Ngai Wong and Venkataramanan Balakrishnan

***Abstract*—Balanced truncation (BT), as applied to date in model order reduction (MOR), is known for its superior accuracy and computable error bounds. Positive-real BT (PRBT) is a particular BT procedure that preserves passivity and stability and imposes no structural constraints on the original state space. However, PRBT requires solving two algebraic Riccati equations (AREs), whose computational complexity limits its practical use in large-scale systems. This paper introduces a novel quadratic extension of the alternating direction implicit (ADI) iteration, which is called quadratic ADI (QADI), that efficiently solves an ARE. A Cholesky factor version of QADI, which is called CFQADI, exploits low-rank matrices and further accelerates PRBT.**

***Index Terms*—Alternating direction implicit (ADI), model order reduction (MOR), positive-real balanced truncation (PRBT), Riccati equation.**

I. Introduction

Interconnect simulation after parasitic extraction, despite its computational load, is a critical postlayout verification step in deep submicrometer very large scale integration design. The high data volume and initial model orders, however, forbid direct computer manipulation. Model order reduction (MOR) comes into place whereby a high-order model is reduced to a (considerably) smaller one without much degradation in accuracy [1]. Moreover, stability and passivity of the original model must be preserved to guarantee stable global simulation [1]–[3]. In particular, a passive system is one that does not generate energy internally, e.g., [4]. A strictly passive system is dissipative and is automatically stable. In linear time-invariant systems, passivity is equivalent to positive realness.

Projection-type MOR schemes such as Passive Reduced-order Interconnect MAcromodeling (PRIMA) [2] and pole analysis via congruence transformations [5], which are usually implemented with the computationally efficient Krylov subspace projection, preserve passivity. However, both algorithms assume special state space structures that are not always feasible [3]. Reduced models from these schemes show similar responses to the original systems, but there is neither direct error connection between the two nor optimality guarantee. On the other hand, balanced truncation (BT) schemes, such as standard BT and positive-real BT (PRBT), offer superior accuracy in reduced-order models [3], [6]–[8] with deterministic error bounds [9], [10], but are largely restricted by the complexity of solving high-order matrix equations and factorizations. PRBT is the BT procedure wherein a pair of algebraic Riccati equations (AREs) are solved. It is also commonly called positive-real truncated balanced realization [3]. References [11] and [12] use balanced stochastic truncation (BST) to denote PRBT, although a more common perception of BST refers to the BT approach in which one Lyapunov equation and one ARE are solved [13]. To quickly solve the Lyapunov equations (linear matrix equations) in standard BT, recent advances utilize the alternating direction implicit (ADI) iteration [14], [15] and Smith method [11], [16] (i.e., ADI with one shift) to exploit low-rank input/output matrices that are pertinent to physical models. A Cholesky factor (CF) variant of ADI, which is called CF-ADI [15], directly computes the factored solution. This avoids the matrix factorizations in standard BT and speeds it up to an extent that is comparable to the projection-based methods [15]–[17]. However, standard BT does not necessarily preserve passivity. PRBT guarantees stability and passivity and has no special structural requirements on the initial state space [3], [8], but faces even heavier computation due to the solution of AREs, which are quadratic matrix equations. Conventional ways of solving an ARE include identifying the stable invariant subspace of a Hamiltonian matrix, or using the Newton method that solves a Lyapunov equation in each iteration, e.g., [11], [13], [16], and [18]–[21]. Nonetheless, the Hamiltonian approach, like eigenvector/Schur-vector or matrix sign function methods, etc., do not explicitly utilize sparse/low-rank matrices and are relatively slow. Using efficient iterative solver algorithms for Lyapunov equations, such as ADI and CF-ADI, the Newton method and its variants exploit matrix structures and have been successfully adapted to large-scale AREs. However, these schemes are based on linearization in each Newton step, and their outer (global) convergence is dependent on inner (local) convergence in each step.

The main contribution of this paper is the formulation of a quadratic ADI (*QADI*) algorithm that efficiently solves an (large-size) ARE (an earlier version of this paper is available in [12], and a recent extension to multishift QADI can be found in [22]). Using a linear fractional transformation (LFT) analysis, which largely simplifies the otherwise intractable derivations, a CF version of QADI, which is called *CFQADI*, is introduced, which further exploits low-rank matrices and produces a factored solution that accelerates PRBT. This work parallels and generalizes the results of [15], viz., on standard BT and CF-ADI, to their second-order counterparts, viz., on PRBT and CFQADI. It is shown that (CF)QADI enjoys similar convergence and complexity to (CF-)ADI, and that the PRBT/CFQADI integration constitutes a powerful candidate for high-speed large-scale passivity-preserving MOR.

II. Preliminaries

A. Basics of PRBT

Interconnect and package modelings generally make use of linear strictly passive RLC components. Consider a large-scale RLC network cast into a state space

$$\dot{x} = A_0 x + B_0 u \quad \text{and} \quad y = C_0 x + D_0 u \tag{1}$$

where $A_0 \in \mathbb{R}^{n\times n}$, $B_0,\ C_0^T \in \mathbb{R}^{n\times m}$, $D_0 \in \mathbb{R}^{m\times m}$, $B_0,\ C_0$ are generally of low-ranks (i.e., $m \ll n$), and u and y are power-conjugate [11]. A_0 is (asymptotically) stable, or equivalently, its spectrum is in the open left half plane, which is denoted by $spec(A_0) \subset \mathbb{C}_-$. Let $M > 0$ ($M \geq 0$) denote a positive definite (positive semidefinite) matrix M. We assume without loss of generality that $D_0 + D_0^T > 0$. Otherwise (for example, in modified nodal analysis, where $D_0 = 0$), the reduction technique in [23] is iteratively used to achieve this. In addition, an impulse-free system in the *descriptor* form [1] with a singular E_0 before $\dot{x}$ can be put into the standard form in (1) [3]. Define the matrix root $DD^T = (D_0 + D_0^T)^{-1}$, $B = B_0 D$, $C = D^T C_0$,

Manuscript received March 9, 2006; revised September 26, 2006. This work was supported in part by the Hong Kong Research Grants Council under Project HKU 7173/04E, by the University Research Committee of The University of Hong Kong, and by the National Science Foundation of the United States of America under Grant ECS-0200320. This paper was recommended by Associate Editor L. M. Silveira.

N. Wong is with the Department of Electrical and Electronic Engineering, The University of Hong Kong, Hong Kong (e-mail: nwong@eee.hku.hk).

V. Balakrishnan is with the School of Electrical and Computer Engineering, Purdue University, West Lafayette, IN 47907 USA (e-mail: ragu@ecn.purdue.edu).

Digital Object Identifier 10.1109/TCAD.2007.895617

and $A = A_0 - BC$. In PRBT, the unique stabilizing solutions X, $Q(\in \mathbb{R}^{n\times n}) \geq 0$ to the dual AREs, i.e.,

$$A^T X + XA + XBB^T X + C^T C = 0 \tag{2a}$$

$$AQ + QA^T + QC^T CQ + BB^T = 0 \tag{2b}$$

are solved such that $spec(A + BB^T X) \subset \mathbb{C}_-$ and $spec(A^T + C^T CQ) \subset \mathbb{C}_-$. Let $X = ZZ^T$ and $Q = YY^T$ be any Cholesky factorizations; the "economic" singular value decomposition (SVD) of the following cross product is found:

$$Y^T Z = U\Sigma V^T, \qquad \Sigma = \mathrm{diag}(\sigma_1, \sigma_2, \ldots, \sigma_k) \geq 0; \quad k \leq n. \tag{3}$$

Suppose that $\sigma_1 \geq \cdots \geq \sigma_r \gg \sigma_{r+1} \geq \cdots \geq \sigma_k$. Let I_m be an identity matrix of dimension m and $0_{m\times n}$ be an $m \times n$ zero matrix. Define the left and right projection matrices to be $T_L = [I_r \quad 0_{r\times(k-r)}]\Sigma^{-1/2}V^T Z^T$ and $T_R = YU\Sigma^{-1/2}[I_r \quad 0_{r\times(k-r)}]^T$, respectively; the system $(T_L A_0 T_R, T_L B_0, C_0 T_R, D_0)$ is the positive-real balanced truncated model whose states are aligned in descending involvement in the energy transfer process [3].

B. Basics of ADI

In general, ADI iteration [14], [15] solves the Lyapunov equation

$$A^T W + WA + C^T C = 0 \tag{4}$$

where the matrix dimensions are consistent with those in (2a). Here, A is assumed to be stable, so there exists a unique $W(\in \mathbb{R}^{n\times n}) \geq 0$ that solves (4). The basic ADI consists of two iterative half-steps, i.e.,

$$\left(A^T + p_j I\right) W_{j-\frac{1}{2}}^T = -C^T C - W_{j-1}^T (A - p_j I) \tag{5a}$$

$$\left(A^T + p_j I\right) W_j = -C^T C - W_{j-\frac{1}{2}} (A - p_j I) \tag{5b}$$

where $W_0 = 0$, and the shift parameters $p_j \in \mathbb{C}_-$ $(j = 1, 2, \ldots)$ appear as real numbers or conjugate pairs. For compactness, we define $S_j = (A + p_j I)^{-1}$ and $T_j = (A - p_j I)$. A useful fact is that, for any integers m and n, the multiplication among S_m, T_n, and A are commutative, and similarly for S_m^T, T_n^T, and A^T. It can be verified that

$$W_j = -\sum_{i=1}^{j} 2p_i \left(\prod_{k=1}^{i-1} S_k^T T_k^T\right) S_i^T C^T C S_i \left(\prod_{k=1}^{i-1} T_k S_k\right). \tag{6}$$

In [15], it is shown that the ordering of the p_j's in (6) is immaterial. Combining (4) and (5), we get

$$W - W_j = \left(\prod_{k=1}^{j} S_k^T T_k^T\right) W \left(\prod_{k=1}^{j} T_k S_k\right). \tag{7}$$

Since A is stable, it is easily shown that $\rho(T_k S_k) < 1$, where $\rho(\circ)$ denotes the spectral radius. Convergence of this form is termed as *superlinear* [15]. To achieve the fastest convergence in, for example, L runs of (5), p_j's are chosen (or approximately chosen) according to the minimax problem, namely,

$$\min_{\{p_1, p_2, \ldots, p_L\}} \left(\max_{\lambda_i \in spec(A)} \left|\prod_{j=1}^{L} \frac{p_j - \lambda_i}{p_j + \lambda_i}\right|\right) \tag{8}$$

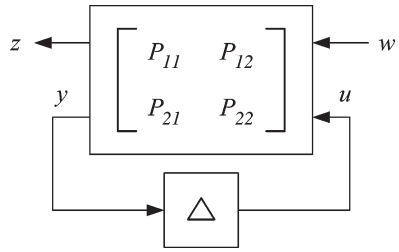

Fig. 1. Lower LFT system.

which is a well-studied, although not well-solved, problem. Popular estimation schemes or heuristics can be found in [15] and [16]. From (7) we can also derive a "residual error" expression for the Lyapunov operator, namely,

$$A^T W_j + W_j A + C^T C = \left(\prod_{k=1}^{j} S_k^T T_k^T\right) C^T C \left(\prod_{k=1}^{j} T_k S_k\right). \tag{9}$$

Because $\rho(T_k S_k) < 1$, the norm of the right-hand side of (9) approaches zero when j tends to infinity.

III. QADI

We focus on the following ARE, which has the form of (2a):

$$A^T X + XA + XBB^T X + C^T C = 0. \tag{10}$$

A is assumed to be stable, and a stabilizing solution $X \geq 0$ exists such that $spec(A + BB^T X) \subset \mathbb{C}_-$. The bounded real lemma [19] states that such an X exists if and only if $\sup \bar{\sigma}(C(j\omega - A)^{-1}B) < 1\ \forall \omega \in \mathbb{R}$, where $\bar{\sigma}(\circ)$ denotes the maximum singular value. The following second-order generalization of ADI, which is called *QADI*, is proposed for solving (10):

$$\left(A^T + X_{j-1}^T BB^T + p_j I\right) X_{j-\frac{1}{2}}^T = -C^T C - X_{j-1}^T (A - p_j I) \tag{11a}$$

$$\left(A^T + X_{j-\frac{1}{2}} BB^T + p_j I\right) X_j = -C^T C - X_{j-\frac{1}{2}} (A - p_j I) \tag{11b}$$

where $X_0 = 0$, and $p_j \in \mathbb{C}_-$, $j = 1, 2, \ldots$, are either real or conjugate pairs. Apparently, (11) reduces to (5) when $B = 0$. For ease of illustration, we will assume, for the rest of this paper, that all p_j's are negative real. However, all qualitative results hold true for conjugate pairs if we combine two runs of (11) into one, and in that case, all quantities remain real. More detail may be found in [22] wherein the formulation and effect of multiple shifts are studied. This seemingly simple modification of ADI, however, gives rise to complicated derivations due to its quadratic nature. An insight that greatly simplifies the analysis of QADI is to recognize (11) as LFTs [19]. Referring to Fig. 1, with

$$P = \begin{bmatrix} P_{11} & P_{12} \\ P_{21} & P_{22} \end{bmatrix}$$

we define the (lower) LFT, which is denoted by $F_l(P, \Delta)$, as the transfer matrix from w to z, i.e., $z = F_l(P, \Delta)w$, which is a mapping of Δ, i.e.,

$$F_l(P, \Delta) = P_{11} + P_{12}\Delta(I - P_{22}\Delta)^{-1} P_{21}$$

where the matrix dimensions are implicitly assumed to be compatible and the matrix inverse well defined. With the definitions $S_j = (A + p_j I)^{-1}$ and $T_j = (A - p_j I)$, it can be shown that

$$X_{j-\frac{1}{2}} = F_l(P_j, X_{j-1}) \quad \text{and} \quad X_j = F_l\left(P_j^T, X_{j-\frac{1}{2}}\right) \tag{12}$$

where

$$P_j = \begin{bmatrix} -C^T C S_j & -T_j^T + C^T C S_j B B^T \\ S_j & -S_j B B^T \end{bmatrix}.$$

Note that we are slightly abusing the LFT notion as it normally denotes a transfer matrix rather than matrix operation. However, all algebras in LFT are applicable as long as the matrix operands are compatibly dimensioned. The chief property of LFTs is that their interconnection again results in an LFT. In particular, the nested connection $X_j = F_l(P_j^T, F_l(P_j, X_{j-1}))$, which is called a *Redheffer Star Product* [19], is an LFT. This enables the combination of the two half-steps in (11) into one, i.e.,

$$X_j = M_{11}^{(j)} + M_{12}^{(j)} X_{j-1} \left(I - M_{22}^{(j)} X_{j-1}\right)^{-1} \left(M_{12}^{(j)}\right)^T \tag{13}$$

$$M_{11}^{(j)} = -2p_j S_j^T C^T \left(I - C S_j B B^T S_j^T C^T\right)^{-1} C S_j \tag{14a}$$

$$M_{22}^{(j)} = -2p_j S_j B \left(I - B^T S_j^T C^T C S_j B\right)^{-1} B^T S_j^T \tag{14b}$$

$$\begin{aligned} M_{12}^{(j)} &= I - 2p_j S_j^T \left(I - C^T C S_j B B^T S_j^T\right)^{-1} \\ &= I - 2p_j S_j^T + S_j^T C^T C M_{22}^{(j)}. \end{aligned} \tag{14c}$$

It can be seen that a symmetric X_{j-1} implies a symmetric X_j. Since $X_0 = 0$, all X_j's are symmetric.

A. Well Posedness

The matrix inverses in (14a)–(14c) are always well defined because the (strict) passivity assumption ensures the existence of a stabilizing solution, which in turn guarantees $\bar{\sigma}(C S_j B) < 1$ due to the bounded real lemma. It remains to show that the inverse $(I - M_{22}^{(j)} X_{j-1})^{-1}$ in (13) is always well defined, and in fact, $X_{j-1}(I - M_{22}^{(j)} X_{j-1})^{-1} \geq 0$, thereby verifying the well posedness of QADI. To begin with, we state two lemmas that are useful for the proof. As noted, p_j's are assumed to be negative real for ease of illustration.

Lemma 1: Assume that (10) has a stabilizing solution X. Define $\tilde{A} = A + BB^T X$, so that $spec(\tilde{A}) \subset \mathbb{C}_-$. Let $\tilde{S}_j = (\tilde{A} + p_j I)^{-1}$ and $\tilde{T}_j = (\tilde{A} - p_j I)$; we have

$$X - X_j = F_l\left(\begin{bmatrix} 0 & \tilde{S}_j^T \tilde{T}_j^T \\ \tilde{T}_j \tilde{S}_j & 2p_j \tilde{S}_j B B^T \tilde{S}_j^T \end{bmatrix}, X - X_{j-1}\right) \tag{15}$$

and it follows that $X - X_{j-1} \geq 0$ implies $X - X_j \geq 0$.

Proof: We first rewrite (11a) and (11b) by the knowledge of (10). Let $\tilde{P}_j = \begin{bmatrix} 0 & -\tilde{T}_j^T \\ \tilde{S}_j & \tilde{S}_j B B^T \end{bmatrix}$, we have

$$X - X_{j-\frac{1}{2}} = F_l(\tilde{P}_j, X - X_{j-1}) \tag{16a}$$

$$X - X_j = F_l\left(\tilde{P}_j^T, X - X_{j-\frac{1}{2}}\right). \tag{16b}$$

Applying the star product, we get (15), and the symmetry follows. ■

Lemma 2: Assume that (10) has a stabilizing solution X. Let $S_j = (A + p_j I)^{-1}$ and $T_j = (A - p_j I)$, we have $\bar{\sigma}(B^T S_j^T (C^T C - 2p_j X) S_j B) < 1$.

Proof: Rearranging (10), we get

$$\begin{aligned} &(A + p_j I)^T X + X(A + p_j I) + XBB^T X \\ &\qquad + \begin{bmatrix} C \\ \left\{\sqrt{-2p_j} Z^T\right\} \end{bmatrix}^T \begin{bmatrix} C \\ \left\{\sqrt{-2p_j} Z^T\right\} \end{bmatrix} = 0 \end{aligned} \tag{17}$$

where $X = ZZ^T$. Obviously, X is also a stabilizing solution to (17). The bounded real lemma implies

$$\bar{\sigma}\left(\begin{bmatrix} C \\ \left\{\sqrt{-2p_j} Z^T\right\} \end{bmatrix} (A + p_j I)^{-1} B\right) = \bar{\sigma}\left(\begin{bmatrix} C \\ \left\{\sqrt{-2p_j} Z^T\right\} \end{bmatrix} S_j B\right) < 1$$

and the proof follows. ■

The next lemma then proves the well posedness.

Lemma 3: Assume that (10) has a stabilizing solution X. Then, in each QADI iteration, $X_{j-1}(I - M_{22}^{(j)} X_{j-1})^{-1}$ is well defined and is positive semidefinite.

Proof: First, we note that

$$M_{22}^{(j)} = -2p_j S_j B \left(I - B^T S_j^T C^T C S_j B\right)^{-1} B^T S_j^T \geq 0.$$

Expanding $X_{j-1}(I - M_{22}^{(j)} X_{j-1})^{-1}$, we get (18), shown at the bottom of the page. In going from the second to the third line of (18), we have used the matrix inversion lemma. Clearly, Lemma 3 holds if $\bar{\sigma}(B^T S_j^T (C^T C - 2p_j X_{j-1}) S_j B) < 1$ and $X_{j-1} \geq 0$ for $j = 1, 2, \ldots$, which we will prove inductively.

Set $j = 1$ in (13). By noting $X_0 = 0$ and $M_{11}^{(j)} \geq 0$ for all j's, X_1 is well defined and positive semidefinite. By Lemma 1, we have $X \geq X_1$; also from Lemma 2, we have

$$\begin{aligned} &\bar{\sigma}\left(B^T S_2^T \left(C^T C - 2p_2 X_1\right) S_2 B\right) \\ &\qquad \leq \bar{\sigma}\left(B^T S_2^T \left(C^T C - 2p_2 X\right) S_2 B\right) < 1. \end{aligned}$$

Set $j = 2$ in (13). From the preceding, X_2 is well defined and positive semidefinite. By Lemma 1, we have $X \geq X_2$; also from Lemma 2, we have

$$\begin{aligned} &\bar{\sigma}\left(B^T S_3^T \left(C^T C - 2p_3 X_2\right) S_3 B\right) \\ &\qquad \leq \bar{\sigma}\left(B^T S_3^T \left(C^T C - 2p_3 X\right) S_3 B\right) < 1. \end{aligned}$$

The argument extends to all j's similarly. ■

$$\begin{aligned} &X_{j-1}\left(I - (-2p_j) S_j B \left(I - B^T S_j^T C^T C S_j B\right)^{-1} B^T S_j^T X_{j-1}\right)^{-1} \\ &\quad = X_{j-1}\left(I - \sqrt{-2p_j} S_j B \left(I - B^T S_j^T C^T C S_j B\right)^{-\frac{1}{2}} \sqrt{-2p_j} \left(I - B^T S_j^T C^T C S_j B\right)^{-\frac{1}{2}} B^T S_j^T X_{j-1}\right)^{-1} \\ &\quad = X_{j-1} + (-2p_j) X_{j-1} S_j B \left(I - B^T S_j^T \left(C^T C - 2p_j X_{j-1}\right) S_j B\right)^{-1} B^T S_j^T X_{j-1} \end{aligned} \tag{18}$$

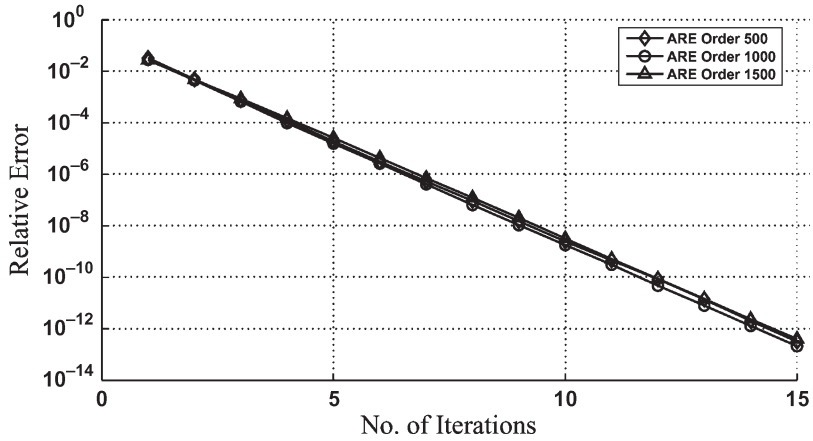

Fig. 2. (CF)QADI: convergence of X_j to the stabilizing X at several orders.

B. Convergence

Analogous to ADI, QADI exhibits superlinear convergence. To show this, we apply (15) recursively to itself. Since $X_0 = 0$,

$$X - X_j = \Pi_j^T X (I + \Omega_j X)^{-1} \Pi_j \tag{19}$$

with

$$\Pi_j = \left(\prod_{k=1}^{j} \tilde{T}_k \tilde{S}_k \right)$$

$$\Omega_j = -\sum_{i=1}^{j} 2p_i \left(\prod_{k=1}^{i-1} \tilde{S}_k \tilde{T}_k \right) \tilde{S}_i B B^T \tilde{S}_i^T \left(\prod_{k=1}^{i-1} \tilde{T}_k^T \tilde{S}_k^T \right).$$

An interesting observation is that Ω_j is exactly the jth iterate of the ADI solution to the Lyapunov equation [cf. (6)]

$$\tilde{A}\Omega + \Omega \tilde{A}^T + BB^T = 0. \tag{20}$$

So, we have $\Omega_j \to \Omega$ as $j \to \infty$. In addition, it is easily proven that $X \geq X(I + \Omega_j X)^{-1}$, which renders $X - X_j \leq \Pi_j^T X \Pi_j$. Comparing this to (7), the error bound from ADI may be borrowed: To achieve the fastest convergence in, for example, L runs of QADI, p_j's are chosen (or approximately chosen) according to the minimax problem, i.e.,

$$\min_{\{p_1, p_2, \ldots, p_L\}} \left(\max_{\lambda_i \in spec(\tilde{A})} \left| \prod_{j=1}^{L} \frac{p_j - \lambda_i}{p_j + \lambda_i} \right| \right) \tag{21}$$

which is effectively a minimax problem on the spectral radius of Π_j. This shows the superlinear convergence of QADI with the difference that the shifts p_j's are now determined from the spectrum of $\tilde{A}$ instead of that of A. Fortunately, although $\tilde{A} = A + BB^T X$ is self-referential to X, its spectrum is known *a priori* (e.g., [19] and [24]). Specifically,

$$spec(\tilde{A}) = spec(H) \cap \mathbb{C}_-,$$

$$\text{where} \quad H = \begin{bmatrix} A & BB^T \\ -C^T C & -A^T \end{bmatrix} \tag{22}$$

is the *Hamiltonian* matrix that is associated with (10).

C. CF Variant

Analogous to CF-ADI [15], when low-rank B and C are present, it is desirable for QADI to work with the CF iterate Z_j where $X_j = Z_j Z_j^T$. Utilizing (13) and (14), we formulate a CF variant of QADI called *CFQADI*. In particular, setting $Z_0 = 0$, for $j = 1, 2, \ldots$,

$$\left(M_{11}^{(j)} \right)^{\frac{1}{2}} = \sqrt{-2p_j} S_j^T C^T \left(I - C S_j B B^T S_j^T C^T \right)^{-\frac{1}{2}} \tag{23a}$$

$$M_{22}^{(j)} = -2p_j S_j B \left(I - B^T S_j^T C^T C S_j B \right)^{-1} B^T S_j^T \tag{23b}$$

$$M_{12}^{(j)} = I - 2p_j S_j^T + S_j^T C^T C M_{22}^{(j)} \tag{23c}$$

$$Z_j = \left[\left(M_{11}^{(j)} \right)^{\frac{1}{2}} \quad M_{12}^{(j)} Z_{j-1} \left(I - Z_{j-1}^T M_{22}^{(j)} Z_{j-1} \right)^{-\frac{1}{2}} \right]. \tag{23d}$$

Each sweep of (23) increases the number of columns in Z_j by that in C^T. Low-rank B and C also allow the use of matrix inversion lemma in (23a)–(23c) to reduce arithmetics. All properties of QADI carry over to CFQADI since they are mathematically equivalent. Consequently, for low-rank input/output matrices, CFQADI provides significant computational and memory savings as only low-rank factors are stored. Symmetry of X_j is perfectly preserved by reconstruction from Z_j. Moreover, the converged factor Z, where $X = ZZ^T$, can readily be adapted to PRBT.

IV. Numerical Example

We study the CPU times of different PRBT implementations. Characterization of QADI and CFQADI as standalone ARE solvers may be found in [12]. On the one hand, PRBT is realized in the conventional way whereby two AREs are solved, followed by CF and SVD computation (cf., Section II-A). The ARE solvers that were used include the Matlab subroutine `aresolv` with the `schur` and `eigen` flags chosen in turn. The former implements the Schur-vector method, while the latter one uses the eigenvector method [18]. Two other Fortran 77 subroutines `slcares` (Schur-vector method) and `slcaregs` (generalized Schur-vector method) are invoked from the SLICOT library [21] via a Matlab gateway. On the other hand, fast PRBT implementations utilizing CF iterates are deployed using CFQADI and the recently proposed NSCARE algorithm [11]. In line with the approach in [13] and [16], NSCARE is a Newton method variant for solving an (particularly large scale) ARE. It uses the Smith method and constructs a CF solution to the Lyapunov equation in each Newton step, thereby indirectly forming a concatenated CF solution to an ARE. Both CFQADI and NSCARE are coded in Matlab m-script (text) files [in solving AREs of the form (2), we neither assume nor exploit any structure, such as bands or sparsity, in A]. All experiments were done

TABLE I
CPU TIMES (IN SECONDS) OF VARIOUS PRBT IMPLEMENTATIONS AND PRIMA

MOR Schemes	Spiral Inductor	RLC Ladder
PRBT/aresolv(schur)	†(98.4+28.1) 126.5	(2286.6+136.0) 2422.6
PRBT/aresolv(eigen)	(53.3+26.1) 79.4	(403.3+125.9) 529.2
PRBT/slcares	(46.0+4.2) 50.2	(390.0+17.4) 407.4
PRBT/slcaregs	(85.2+3.9) 89.1	(904.2+17.8) 922.0
PRBT/NSCARE	(7.19+0.97) 8.16	(20.56+0.80) 21.36
PRBT/CFQADI	(2.84+0.02) 2.86	(2.67+0.02) 2.69
PRIMA	0.67	2.53

† PRBT time breakdown: (Two AREs + Matrix Factorizations) Total

in the Matlab R14 (SP2) environment on a 3-GHz personal computer with 3 G random access memory. Both NSCARE and CFQADI are non-Hamiltonian solvers, while others are based on identifying the stable invariant subspace of a Hamiltonian matrix. Fig. 2 plots the metric $\|X_j - X\|_F/\|X\|_F$, with $\|\circ\|_F$ being the Frobenius norm, in typical ARE solutions by QADI (or equivalently CFQADI) at several ARE orders. Superlinear convergence of (CF)QADI can be observed from these virtually straight curves.

PRBT is performed on two benchmarks: 1) a spiral inductor model of order 500 and 2) an RLC ladder circuit of order 800 [11]. CFQADI has been used since its CF iterates can take advantage of low-rank input/output matrices to reduce computation and memory space. Table I tabulates the CPU times of various PRBT implementations and also that by the projection-based PRIMA [2] algorithm. All reduced-order models are passive. Figs. 3(a) and 4(a) show the frequency responses of the original and reduced-order models, while Figs. 3(b) and 4(b) plot the approximation errors. It is seen that the PRBT curves by different solvers virtually overlap because they all solve the same set of AREs, while the PRIMA curves exhibit relatively larger errors. This is expected as reduced-order models from PRBT tend to have excellent global accuracy [3], [11]. Moreover, PRBT avoids the selection of frequency expansion points and final model order as in PRIMA. Solutions from CFQADI and NSCARE are computed to the same or better accuracy than those by other PRBT algorithms. Specifically, for these CF-iterate approaches in which Z_j and Y_j ($X_j = Z_j Z_j^T$ and $Q_j = Y_j Y_j^T$) are progressively computed, the cross-product stopping criterion [17], which monitors the Frobenius norm update in $Y_j^T Z_j$ [cf., (3)], has been used. In addition, it can be further shown that the set of singular values of $Y_j^T Z_j$ thus obtained, i.e., $\{\hat{\sigma}_i\}$, approaches that in $Y^T Z$, i.e., $\{\sigma_i\}$, exponentially [17].

Moreover, among all PRBT implementations, NSCARE and CFQADI exhibit superior speed and scalability over others, with CFQADI being the fastest. This is even more obvious in high-order examples, including some that were not reported here, where CFQADI approaches the speed of PRIMA. Despite the comparable speed of NSCARE to CFQADI, the final CF solution from CFQADI always has much fewer columns and thus lower numerical ranks. For example, in the spiral inductor case, the size of the terminating Z_j and Y_j by NSCARE is about 500×380 and only about 500×100 for CFQADI. In the ladder circuit case, they are about 800×450 and 800×25, respectively. This can be attributed to the strength of PRBT/CFQADI in capturing the fast decaying singular values of the cross product and also explains the remarkable speed of CFQADI. On the other hand, NSCARE builds the CF solution to an ARE progressively from intermediate Lyapunov equations. Subsequently, convergence of NSCARE is dependent on the convergence in respective Lyapunov equation solutions, while that of CFQADI is reliant on the spectrum of the Hamiltonian matrix (which determines the shifts) that is formed directly from the original ARE matrices. Another major merit of these CF-type algorithms, as seen from the breakdowns in Table I, is the avoidance of the (large-scale) CF and SVD factorizations. Although [11] has shown that PRBT time by Hamiltonian-based solvers can almost be halved by complete subspace separation, the speed improvement by CFQADI is much more than double, e.g., PRBT/CFQADI is more than 150× faster than PRBT/`slcares` in the second benchmark.

V. REMARKS

(CF)QADI is a (large-scale) ARE solver algorithm that features simple codes. To our knowledge, CFQADI is the first algorithm that directly computes the CF solution to an ARE through CF iterates, instead of the concatenated CF solution from the Newton method [11], [16]. With low-rank input/output matrices, the CF solution thus obtained is usually of low rank, thereby avoiding large-size matrix factorizations in the original PRBT procedure. The low-rank factors also reduce memory requirement and improve scalability. The runtimes of QADI and CFQADI are dominated by the number of shifts. The most expensive step is the matrix inversion in finding S_j for each p_j, which takes roughly $3n^3$ flops in the most general case when A is dense. If the number of shifts is L (which equals one in our experiments), the work of both algorithms is proportional to $3Ln^3$. All other operations in CFQADI are of $O(n^2)$ work due to exploitation of low-rank matrices. In contrast, the complexities of the Schur-vector and eigenvector methods are roughly $50n^3 - 150n^3$ flops. Therefore, the work of (CF)QADI increases in a cubic manner but much more slowly than that of conventional solvers. If matrix inversion can be done in $O(n^2)$ work, e.g., when A is sparse or banded, then (CF)QADI will reduce to an $O(n^2)$ algorithm. Regarding memory, CFQADI requires $O(nm)$ space (usually $m \ll n$) due to its storage of CF iterates. Most conventional algorithms require $O(n^2)$ space due to the storage of square matrices.

For simplicity and demonstration, only a single shift has been used in CFQADI in our examples, which is analogous to the Smith method as a special case of ADI [11], [16]. Referring to (22), we have chosen $p = -(\rho(\tilde{A})/\rho(\tilde{A}^{-1}))^{1/2}$ [11]. Owing to the symmetry in the spectrum of H, we also have $p = -(\rho(H)/\rho(H^{-1}))^{1/2}$, which is then estimated through simple power iterations. The extension to multiple-shift (CF)QADI can be found in our recent work in [22]. Some useful facts are in order. In solving the dual AREs in (2) via CFQADI, the following Hamiltonian matrices are set up to find the shifts:

$$H = \begin{bmatrix} A & BB^T \\ -C^TC & -A^T \end{bmatrix} \quad \text{and} \quad H' = \begin{bmatrix} A^T & C^TC \\ -BB^T & -A \end{bmatrix}$$

corresponding to (2a) and (2b), respectively. With some care, it can be shown that $spec(H) = -spec(H') = spec(H')$. Therefore, the same set of shifts p_j's for (2a) can be reused in (2b). In addition, the efficient implicitly restarted shift-and-invert Arnoldi algorithm in [24], which is called SHIRA, is particularly suitable for the computation of the extremal eigenvalues of H.

When CFQADI is terminated before convergence (due to slow convergence or time constraint), the PRBT/CFQADI-reduced model may only be near passive, and passivity enforcement is needed. This can be done by, e.g., the algorithm in [4]. Fortunately, the reduced models are of low orders (a few tens), rendering such enforcement computationally fast. Generally, (CF)QADI performs better in damped systems than in lightly damped systems, where the spectral radius $\rho(\tilde{T}_k \tilde{S}_k)$ and that of Π_j [cf. (19)] are near unity. Such problem can be mitigated by choosing multiple p_j's to accelerate the convergence of (CF)QADI [22] or by parallel computing, despite a bigger overhead in finding p_j's and more explicit inversions. However, these topics are beyond the scope of this paper and will not be elaborated.

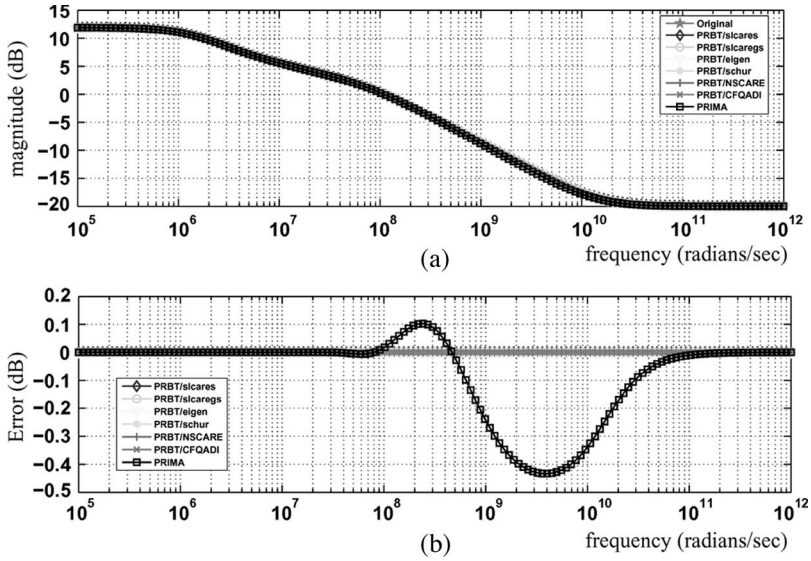

Fig. 3. (a) Frequency responses of the spiral inductor model (order = 500) and the reduced-order models (order = 9). (b) Approximation errors.

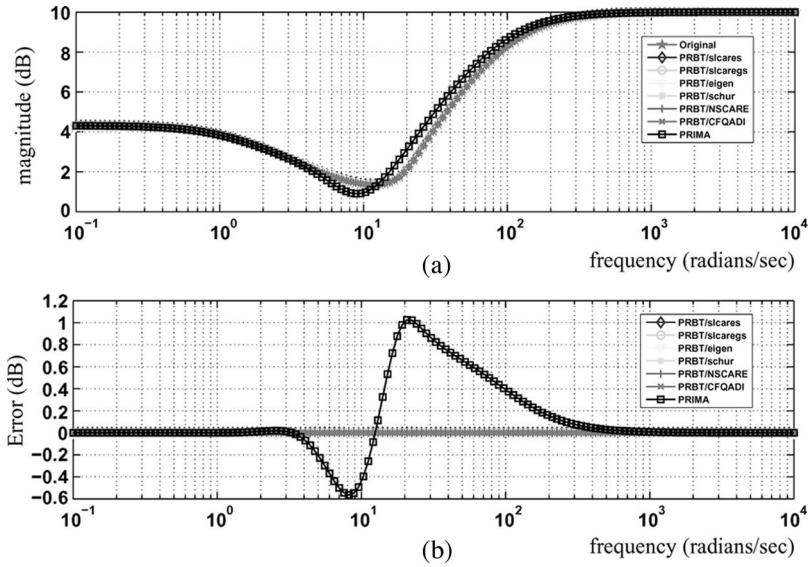

Fig. 4. (a) Frequency responses of the RLC ladder model (order = 800) and the reduced-order models (order = 6). (b) Approximation errors.

VI. Conclusion

This paper has presented a highly efficient PRBT implementation based on the fast ARE solver called QADI iteration. Well posedness and convergence of QADI have been analytically proved. QADI facilitates a CF variant, which is called CFQADI, that exploits low-rank matrices and avoids large-scale matrix factorizations, thereby resulting in fast PRBT computation and significant memory savings. Numerical examples have verified the remarkable efficacy of the PRBT/CFQADI integration over conventional PRBT realizations.

References

[1] Z. Bai, P. M. Dewilde, and R. W. Freund, *Reduced-Order Modeling*. Murray Hill, NJ: Bell Lab, Mar. 2002. Numerical Analysis Manuscript 02-4-13.

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

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

[4] S. Grivet-Talocia, "Passivity enforcement via perturbation of Hamiltonian matrices," *IEEE Trans. Circuits Syst. I, Fundam. Theory Appl.*, vol. 51, no. 9, pp. 1755–1769, Sep. 2004.

[5] K. J. Kerns and A. T. Yang, "Stable and efficient reduction of large, multiport RC networks by pole analysis via congruence transformations," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 16, no. 7, pp. 734–744, Jul. 1997.

[6] L. M. Silveira, I. M. Elfadel, J. K. White, M. Chilukuri, and K. S. Kundert, "Efficient frequency-domain modeling and circuit simulation of transmission lines," *IEEE Trans. Compon., Packag., Manuf. Technol. B*, vol. 17, no. 4, pp. 505–513, Nov. 1994.

[7] P. Rabiei and M. Pedram, "Model order reduction of large circuits using balanced truncation," in *Proc. Asia and South Pacific Des. Autom. Conf.*, Jan. 1999, pp. 237–240.

[8] P. Heydari and M. Pedram, "Model-order reduction using variational balanced truncation with spectral shaping," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 53, no. 4, pp. 879–891, Apr. 2006.

[9] 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, Jun. 1984.

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

[11] N. Wong, V. Balakrishnan, C.-K. Koh, and T. S. Ng, "Two algorithms for fast and accurate passivity-preserving model order reduction," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 25, no. 10, pp. 2062–2075, Oct. 2006.

[12] N. Wong and V. Balakrishnan, "Fast balanced stochastic truncation via a quadratic extension of the alternating direction implicit iteration," in *Proc. Int. Conf. Comput.-Aided Des.*, Nov. 200, pp. 801–805.

[13] P. Benner, E. S. Quintana-Ortí, and G. Quintana-Ortí, "Efficient numerical algorithms for balanced stochastic truncation," *Int. J. Appl. Math. Comput. Sci.*, vol. 11, no. 5, pp. 1123–1150, 2001.

[14] A. Lu and E. L. Wachspress, "Solution of Lyapunov equations by alternating direction implicit iteration," *Comput. Math. Appl.*, vol. 21, no. 9, pp. 43–58, 1991.

[15] J. Li and J. White, "Low-rank solution of Lyapunov equations," *SIAM Rev.*, vol. 46, no. 4, pp. 693–713, 2004.

[16] T. Penzl, (2000, Aug.). *LYAPACK users guide—A MATLAB toolbox for large Lyapunov and Riccati equations, model reduction problems, and linear-quadratic optimal control problems*. [Online]. Available: http://www.tu-chemnitz.de/sfb393/sfb00pr.html

[17] V. Balakrishnan, Q. Su, and C.-K. Koh, "Efficient balance-and-truncate model reduction for large scale systems," in *Proc. Amer. Control Conf.*, Jun. 2001, pp. 4746–4751.

[18] I. W. F. Arnold, "Generalized eigenproblem algorithms and software for algebraic Riccati equations," *Proc. IEEE*, vol. 72, no. 12, pp. 1746–1754, Dec. 1984.

[19] K. Zhou, J. C. Doyle, and K. Glover, *Robust and Optimal Control*. Upper Saddle River, NJ: Prentice-Hall, 1996.

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

[21] P. Benner, V. Mehrmann, V. Sima, S. Van-Huffel, and A. Varga, "SLICOT—A subroutine library in systems and control theory," in *Applied and Computational Control, Signals and Circuits*, vol. 1, B. N. Datta, Ed., Basel, Switzerland: Birkhauser, 2006, pp. 499–539. [Online]. Available: http://www.amazon.com/Applied-Computational-Control-Signals-Circuits/dp/0817639543

[22] N. Wong and V. Balakrishnan, "Multi-shift quadratic alternating direction implicit iteration for high-speed positive-real balanced truncation," in *Proc. IEEE Des., Autom. Conf.*, Jun. 2006, pp. 257–260.

[23] H. Weiss, Q. Wang, and J. L. Speyer, "System characterization of positive real conditions," *IEEE Trans. Autom. Control*, vol. 39, no. 3, pp. 540–544, Mar. 1994.

[24] V. Mehrmann and D. Watkins, "Structure-preserving methods for computing eigenpairs of large sparse skew-Hamiltonian/Hamiltonian pencils," *SIAM J. Sci. Comput.*, vol. 22, no. 6, pp. 1905–1925, 2001.