<table>
<thead>
<tr>
<th>Title</th>
<th>Efficient positive-real balanced truncation of symmetric systems via cross-riccati equations</th>
</tr>
</thead>
<tbody>
<tr>
<td>Author(s)</td>
<td>Wong, N</td>
</tr>
<tr>
<td>Issued Date</td>
<td>2008</td>
</tr>
<tr>
<td>URL</td>
<td><a href="http://hdl.handle.net/10722/57468">http://hdl.handle.net/10722/57468</a></td>
</tr>
<tr>
<td>Rights</td>
<td>This work is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International License.; ©2008 IEEE. Personal use of this material is permitted. However, permission to reprint/republish this material for advertising or promotional purposes or for creating new collective works for resale or redistribution to servers or lists, or to reuse any copyrighted component of this work in other works must be obtained from the IEEE.</td>
</tr>
</tbody>
</table>
Efficient Positive-Real Balanced Truncation of Symmetric Systems Via Cross-Riccati Equations

Ngai Wong, Member, IEEE

Abstract—We present a highly efficient approach for realizing a positive-real balanced truncation (PRBT) of symmetric systems. The solution of a pair of dual algebraic Riccati equations in conventional PRBT, whose cost constrains practical large-scale deployment, is reduced to the solution of one cross-Riccati equation (XRE). The cross-Riccatian nature of the solution then allows a simple construction of PRBT projection matrices, using a Schur decomposition, without actual balancing. An invariant subspace method and a modified quadratic alternating-direction-implicit iteration scheme are proposed to efficiently solve the XRE. A low-rank variant of the latter is shown to offer a remarkably fast PRBT speed over the conventional implementations. The XRE-based framework can be applied to a large class of linear passive networks, and its effectiveness is demonstrated through numerical examples.

Index Terms—Alternating direction implicit (ADI), cross-Riccati equation (XRE), positive-real balanced truncation (PRBT), Schur decomposition, symmetric systems.

I. INTRODUCTION

VERY LARGE scale integration (VLSI) interconnect and package modeling typically involve massively coupled RLC components. The enormous orders of the models thus formed prohibit direct computer simulation. Model order reduction (MOR) has become a standard routine in simulators wherein the original system is approximated by a reduced-order model with little degradation in time/frequency-domain accuracy. In addition, the reduced-order model must preserve critical properties like stability and passivity [1] of the original model to ensure a correct global simulation [2]–[4]. In particular, a passive system is one that does not generate energy internally. A strictly passive system is a dissipative system and is automatically stable. In linear systems, passivity is equivalent to positive realness [1], [4], [5].

Numerous MOR schemes provide various tradeoffs between accuracy and computation. Among them, balanced truncation (BT) offers superior accuracy and closed-form error bounds [3]–[11]. The key idea of BT is to align and sort the internal states of the original model based on their importance in input–output state activities or energy transfer. The least important states are then truncated with little impact on the system response. In a standard BT, the bottleneck is to solve two linear matrix equations, which are called the Lyapunov equations, for obtaining the controllability and observability Gramians. The cross product of these Gramians is then used to find low-order projection matrices to reduce (truncate) the original system. However, the standard BT does not guarantee passivity. Positive-real BT (PRBT) is another BT variant that preserves both passivity and stability [5] and has no special semidefiniteness constraints on the internal state-space matrices [3]. However, it requires solving a pair of algebraic Riccati equations (AREs) whose complexity is even higher than that of the Lyapunov equations.

In the standard BT of single-input–single-output (SISO) or symmetric multiple-input–multiple-output (MIMO) systems, information in the cross product of the controllability and observability Gramians can be directly obtained by solving a linear matrix equation called the Sylvester equation [6], [11]–[13]. Since only one matrix equation is solved, computation is practically halved, with further advantages like better consistency and numerical robustness [6], [11], [14]. The quadratic counterpart of the Sylvester equation, which is called the cross-Riccati equation denoted by XRE in this paper, appeared in [15]–[19] on control topics like feedback control, bounded-real systems, and discrete stochastic processes. Its integration with the PRBT, however, has not been fully appreciated by the Electronic Design Automation community. Moreover, a few research works have been done on the computationally competitive algorithms for solving large-scale XREs frequently encountered in the VLSI MOR.

By collecting distributed results and introducing new techniques, this paper generalizes, in the context of symmetric MIMO systems, the cross-Gramian framework in the standard BT [6] to the cross-Riccatian counterpart in the PRBT. To begin with, a standard invariant subspace method [15], [20] for solving the XRE, with new spectral characterization and derivations in relation to the dual AREs, is described. A Schur decomposition method, which is borrowed from the standard BT procedure [6], then allows a simple construction of projection matrices to obtain equivalent PRBT-reduced models without actual balancing. Next, to address the computational bottleneck in solving the XRE, a recently proposed fast ARE solver algorithm called the quadratic alternating-direction-implicit (QADI) iteration [8]–[10] is extended to XQADI (“X” for “cross” Riccati) which solves an XRE efficiently. A linear-fractional-transformation (LFT) analysis shows that the XQADI inherits the superlinear convergence of QADI and enjoys simple coding. Nonetheless, the real benefit of the XQADI comes from a mathematically equivalent but low-rank
implementation of the XQADI called the LRXQADI (“LR” for low rank or left–right). In parallel to the Cholesky-factor QADI (CFQADI) [8]–[10], the LRXQADI exploits and utilizes low-rank factored iterates to significantly speed up computation and reduce memory storage during the iterative XRE solution. The factored cross Riccatian (namely, the solution to the XRE) further permits rapid construction of PRBT projection matrices. The end result is a PRBT/XRE/LRXQADI flow that exhibits a remarkable efficiency comparable with the moment-matching projection-based algorithms such as PRIMA [2], but with the superior global accuracy and deterministic error bound pertinent to the PRBT. Although the proposed approach applies to symmetric systems only, its scope encompasses the large class of passive linear networks, which are commonly used in interconnect and package modeling, that always have symmetric admittance or impedance matrices due to reciprocity. A reciprocal system is the one whose voltage and current values at any two points in the network can be interchanged [21], [22].

This paper is an extension of the earlier work in [20] and is organized as follows. Section II briefly reviews the PRBT. The use of the Sylvester equation in the standard BT of symmetric systems is succinctly discussed. Section III presents the XRE and a standard invariant subspace approach for solving it, together with some new spectral characterizations. The Schur decomposition method for constructing the projection matrices and obtaining the PRBT-equivalent reduced-order model is also described. Section IV introduces the XQADI and the LRXQADI and shows how the low-rank nature of the latter gives rise to significant computational savings. The proposed PRBT/XRE flow is then applied to some practical examples in Section V. Finally, Section VI draws the conclusion.

II. BACKGROUND AND PRELIMINARIES

A. PRBT

Consider the state space of a minimal and passive MIMO square system

\[
\begin{align*}
\dot{x} &= A_0 x + B_0 u \\
y &= C_0 x + D_0 u
\end{align*}
\]

where \(A_0 \in \mathbb{R}^{n \times n}\), \(B_0, C_0^T \in \mathbb{R}^{n \times m}\), \(D_0 \in \mathbb{R}^{m \times m}\), and usually, \(m \ll n\). The transfer function/matrix is \(G(s) = D_0 + C_0 (s I - A_0)^{-1} B_0\). The input \(u\) and output \(y\) are assumed power conjugated. For every entry of \(u\) that is a node voltage (branch current), the corresponding entry of \(y\) is a branch current (node voltage) so that \(u^T y\) represents a power metric. We assume that \(A_0\) is asymptotically stable, i.e., its spectrum is in the open left half-plane, denoted by \(\text{spec}(A_0) \subset \mathbb{C}_-\).

Let \(M > 0(\geq 0)\) denote a positive definite (positive semidefinite) matrix \(M\). We assume without loss of generality that \(D_0 + D_0^T > 0\) (i.e., the system is strictly PR or strictly passive [1]); otherwise, the algorithm in [23] is used to achieve so in a reduced-order embedded system. In addition, an impulse-free system in the descriptor form [4] with a singular \(E_0\) before \(\dot{x}\) can be put into the regular form in (1) [3]. The PR lemma [1], [4] states that the linear system (1) is passive if and only if there exists an \(X \in \mathbb{R}^{n \times n}\) \(> 0\) satisfying the linear matrix inequality

\[
\begin{bmatrix}
A_0^T X + X A_0 & X B_0 - C_0^T D_0^T \\
B_0^T X - C_0 & -(D_0 + D_0^T)\end{bmatrix} \leq 0.
\]

By Schur complement, (2) is equivalent to

\[
A_0^T X + X A_0 + (X B_0 - C_0^T) (D_0 + D_0^T)^{-1} (B_0^T X - C_0) \leq 0.
\]

Setting (3) to an equality results in an ARE. In the PRBT, the unique stabilizing solutions [1] \(X_>(> 0)\) and \(X_o(> 0)\) to the dual AREs

\[
\begin{align*}
A_0 X_c + X_c A_0^T + (X_c C_0^T B_o - B_o) \\
&\times (D_0 + D_0^T)^{-1} (C_0 X_c - B_o) = 0 \\
A_0^T X_o + X_o A_0 + (X_o B_0 - C_0^T) \\
&\times (D_0 + D_0^T)^{-1} (B_o^T X_o - C_0) = 0
\end{align*}
\]

are solved. Their existence is guaranteed by the passivity assumption. To simplify notations, we define

\[
\begin{align*}
B &= B_o (D_0 + D_0^T)^{-1/2}, & C &= (D_0 + D_0^T)^{-1/2} C_0 \\
A &= A_0 - B_o (D_0 + D_0^T)^{-1} C_0
\end{align*}
\]

where “:=” denotes assignment, such that (4) also reads

\[
\begin{align*}
A X_c + X_c A^T + X_c C^T C x + B B^T &= 0 \\
A^T X_o + X_o A + X_o B B^T X_o + C^T C &= 0
\end{align*}
\]

Following the standard BT terminology in Section II-B, we call \(X_c\) the controllability Riccatian and \(X_o\) the observability Riccatian (although they are also sometimes known as the PR Gramians). By factoring out \(X_c\) and \(X_o\) in (6) for their coefficient matrices, we define

\[
\begin{align*}
A_c &= A + X_c C^T C \\
A_o &= A + B B^T X_o
\end{align*}
\]

Stabilizability of solutions implies that \(\text{spec}(A_c) = \text{spec}(A_o) \subset \mathbb{C}_-\) [7]. Let \(X_c = L_c L_c^T\) and \(X_o = L_o L_o^T\), where \(L_c, L_o \in \mathbb{R}^{n \times n}\), be any matrix square-root decompositions. Compute the singular value decomposition

\[
L_o^T L_o = U \Sigma V^T, \quad \Sigma = \text{diag}(\sigma_1, \ldots, \sigma_n).
\]

Here, \(\Sigma\) is a diagonal matrix with descending singular values

\[
\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_r \gg \sigma_{r+1} \geq \cdots \geq \sigma_n.
\]

By defining the matrices

\[
T_R := L_o U \Sigma^{-\frac{1}{2}} \quad \text{and} \quad T_L := T_R^{-1} = \Sigma^{-\frac{1}{2}} V^T L_o^T
\]
and using ‘→’ to denote the corresponding similarity transform, we get the PR-balanced model \((A_0, B_0, C_0, D)\), namely
\[
(A_0, B_0, C_0, D_0) \rightarrow (\hat{A}_0, \hat{B}_0, \hat{C}_0, D)
\]
\[
= (T_I A_0 T_R, T_I B_0, C_0 T_R, D_0). \tag{11}
\]
Moreover, in this PR-balanced state space, the controllability and observability Riccian matrices are simultaneously diagonalized
\[
X_c → T_I X_c T_R^T = \Sigma \quad \text{and} \quad X_o → T_R^T X_o T_R = \Sigma \tag{12}
\]
which satisfy
\[
\hat{A}_0 \Sigma + \Sigma \hat{A}_0^T + \left( \Sigma \hat{C}_0^T - \hat{B}_0 \right) (D_0 + D_0^T)^{-1} \left( \hat{C}_0 \Sigma - \hat{B}_0 \right) = 0
\]
\[
\left( 13a \right)
\]
\[
\hat{A}_0^T \Sigma + \Sigma \hat{A}_0 + \left( \Sigma \hat{B}_0 - \hat{C}_0^T \right) (D_0 + D_0^T)^{-1} \left( \hat{B}_0^T \Sigma - \hat{C}_0 \right) = 0.
\]
\[
\left( 13b \right)
\]
The state variables \((\hat{A}_0, \hat{B}_0, \hat{C}_0, D_0)\) are aligned in descending importance in energy transfer [3]. Express \(\Sigma = \text{diag}(\Sigma_0, \Sigma_n)\), where \(\Sigma_0 \in \mathbb{R}^{n \times r}\) holds the “bigger” singular values, and \(\Sigma_n \in \mathbb{R}^{(n-r) \times (n-r)}\) holds the “smaller” ones. Partition columns of \(T_R\) and rows of \(T_I\) accordingly so that with respect to (12)
\[
X_c = T_R \Sigma T_R^T = [T_{Rb} \quad T_{Rs}] \begin{bmatrix} \Sigma_b & \Sigma_s \end{bmatrix} \begin{bmatrix} T_{Rb}^T \\ T_{Rs}^T \end{bmatrix} \tag{14a}
\]
\[
X_o = T_I^T \Sigma T_I = [T_{Ib} \quad T_{Is}] \begin{bmatrix} \Sigma_b & \Sigma_s \end{bmatrix} \begin{bmatrix} T_{Ib}^T \\ T_{Is}^T \end{bmatrix}. \tag{14b}
\]
The PRBT-reduced model is then obtained from the rank-r subspace projection
\[
(\hat{A}_r, \hat{B}_r, \hat{C}_r, D) = (T_{Ib} A_0 T_{Rb}, T_{Ib} B_0, C_0 T_{Rb}, D_0) \tag{15}
\]
where \(\hat{A}_r \in \mathbb{R}^{r \times r}\) and \(\hat{B}_r, \hat{C}_r \in \mathbb{R}^{r \times m}\). It is easily checked that the system in (15) is passive and stable. In addition, the transfer matrix \(G_r(s) = D_0 + \hat{C}_r(sI - \hat{A}_r)^{-1} \hat{B}_r\) has an \(H_\infty\) norm approximation error bound determined by the truncated subsystem [5].

**B. Sylvester Equation in Symmetric-System Standard BT**

The standard BT involves a pair of dual Lyapunov equations
\[
A_0 W_c + W_c A_0^T + B_0 B_0^T = 0 \tag{16a}
\]
\[
A_0^T W_o + W_o A_0 + C_0^T C_0 = 0 \tag{16b}
\]
whose solutions \(W_c\) and \(W_o\) are the controllability and observability Gramians, respectively. The rest of the standard BT procedure is the same as in the PRBT except that \(X_c\) and \(X_o\) are replaced with \(W_c\) and \(W_o\), respectively. Previous work has studied the use of the cross Gramian in the standard BT of the symmetric MIMO systems (which trivially include the SISO systems) [11], [12], [14]. Specifically, the cross Gramian of a symmetric system \(W_{co}\) is solved from the Sylvester equation
\[
A_0 W_{co} + W_{co} A_0 + B_0 C_0 = 0 \tag{17}
\]
where \(W_{co}\) has the important property \(W_{co}^2 = W_c W_o\). Subsequently, the range or image of \(W_{co}(W_{co}^2)\) also spans the reachable (observable) space of (1). Moreover, it can be easily shown that the eigenvalues of \(W_{co}\) are invariant under similarity transforms and that their magnitudes are exactly the Hankel singular values of (1). In other words, the dominant left and right eigenspaces of \(W_{co}\) capture most of the input–output energy transfer defined in the sense of the Hankel norm [1] (with the input and output energy measures defined by \(\int_{-\infty}^{0} u^T u dt\) and \(\int_{0}^{\infty} y^T y dt\), respectively). This fact allows the standard BT to proceed directly by utilizing the eigenvector bases of \(W_{co}\) without solving the two Lyapunov equations [6].

**III. HAMILTONIAN-BASED SYMMETRIC-SYSTEM PRBT**

We first define system symmetry. The MIMO system \((A_0, B_0, C_0, D_0)\) in (1) is symmetric if \(G(s) = G(s)^T\) or
\[
D_0 + C_0(sI - A_0) = D_0^T + B_0^T (sI - A_0^T)^{-1} C_0^T \tag{18}
\]
for all \(s \in \mathbb{C} \setminus \text{spec}(A_0)\), where ‘−’ denotes set subtraction. This necessitates \(D_0 = D_0^T\). An alternative but equivalent definition for a symmetric system is that \(D_0 = D_0^T\) and all of the system Markov parameters are symmetric, namely
\[
C_0 A_0^i B_0 = (C_0 A_0^0 B_0)^T = B_0^T (A_0^T)^i C_0^T \tag{19}
\]
for all integers \(i \geq 0\). The symmetry and minimality of the system means that \((A_0, B_0, C_0)\) is similar to \((A_0^T, C_0^T, B_0^T)\), which is denoted by \((A_0, B_0, C_0) \sim (A_0^T, C_0^T, B_0^T)\), through a nonsingular and symmetric \(T = T^T \in \mathbb{R}^{n \times n}\) (e.g., [14] and [15], Th. 6.2–4 in [24]) such that
\[
A_0^T = T^{-1} A_0 T, \quad C_0^T = T^{-1} C_0, \quad \text{and} \quad B_0^T = C_0 T. \tag{20}
\]
It can be checked that the second and third equalities in (20) imply one another. On the other hand, for the matrices defined in (5), we readily have \(T = T^T\) and
\[
A^T = T^{-1} A T, \quad C^T = T^{-1} C, \quad \text{and} \quad B^T = C T. \tag{21}
\]
In other words, \((A, B, C) \sim (A^T, C^T, B^T)\). Analogous to (16) and (17), we formulate, with respect to (6), the XRE
\[
A X_{co} + X_{co} A + X_{co} B C X_{co} + BC = 0. \tag{22}
\]
Here, \(X_{co}\) is called the cross Riccatian as in [16] and [17]. The XRE-based PRBT of symmetric systems consists of two steps, namely, solving the XRE and constructing the projection matrices, which are described in the following.

**A. XRE Solution Via Stable Invariant Subspace Identification**

We first present a standard way, based on (21) and the Hamiltonian-like approach in [15], for solving (22) that arises from a symmetric system. We also give some new analytical derivations and results regarding the spectral properties of the XRE solution thus obtained [20]. First, define two
Hamiltonian matrices [1] $H_c$ and $H_o$ corresponding to (6a) and (6b), respectively,

$$H_c := \begin{bmatrix} A^T & C^T C \\ -BB^T & -A \end{bmatrix}, \quad H_o := \begin{bmatrix} A & BB^T \\ -C^T C & -A^T \end{bmatrix}. \tag{23}$$

The spectral structure of a Hamiltonian matrix and the solutions of (6a) and (6b) by identifying the stable invariant subspaces of $H_c$ and $H_o$, respectively, are well studied [1], [7]. We note that $H_c$ and $H_o$ share the same spectrum since

$$H_c = \begin{bmatrix} 0 & I \\ I & 0 \end{bmatrix} (-H_o) \begin{bmatrix} 0 & I \\ I & 0 \end{bmatrix}.$$  

In [7], it has been shown that when the stable and unstable subspaces of $H_o$ are separated (passivity implies no purely imaginary eigenvalues), i.e.,

$$H_o \begin{bmatrix} X_{11} & X_{12} \\ X_{21} & X_{22} \end{bmatrix} = \begin{bmatrix} X_{11} & X_{12} \\ X_{21} & X_{22} \end{bmatrix} \begin{bmatrix} \Phi & 0 \\ 0 & \Phi \end{bmatrix}, \tag{24}$$

with $\Phi_- \in \mathbb{R}^{n \times n}$ and $\Phi_+ \in \mathbb{R}^{n \times n}$ corresponding to the stable and unstable eigenvalues, respectively, of spec($H_o$), then the stabilizing solution to (6b) is $X_o = X_{21}X_{11}^{-1}$ and that to (6a) is $X_c = X_{12}X_{22}^{-1}$. Now, we show that this invariant subspace approach is also applicable to solving (22). First, define

$$H_{co} := \begin{bmatrix} A & BC \\ -BC & -A \end{bmatrix}. \tag{25}$$

It can be seen that spec($H_{co}$) = spec($H_o$) = spec($H_c$) since

$$H_{co} = \begin{bmatrix} 0 & I \\ I & 0 \end{bmatrix} H_o \begin{bmatrix} 0 & I \\ I & 0 \end{bmatrix}^{-1}. \tag{26}$$

By applying (26) to (24), we get

$$H_{co} \begin{bmatrix} X_{11} \\ TX_{21} \end{bmatrix} = \begin{bmatrix} X_{11} \\ TX_{21} \end{bmatrix} \begin{bmatrix} \Phi & 0 \\ 0 & \Phi \end{bmatrix}. \tag{27}$$

Then, by noting

$$\begin{bmatrix} 0 & I \\ I & 0 \end{bmatrix} H_{co} \begin{bmatrix} 0 & I \\ I & 0 \end{bmatrix} = -H_{co}, \tag{27}$$

(27) is broken into two equations

$$H_{co} \begin{bmatrix} X_{11} \\ TX_{21} \end{bmatrix} = \begin{bmatrix} X_{11} \\ TX_{21} \end{bmatrix} \Phi_-, \tag{28a}$$

$$H_{co} \begin{bmatrix} TX_{22} \\ X_{12} \end{bmatrix} = \begin{bmatrix} TX_{22} \\ X_{12} \end{bmatrix} \Phi_+. \tag{28b}$$

Simple algebra shows that $TX_{21}X_{11}^{-1} = TX_o$ and $X_{12}X_{22}^{-1}X_{11}^{-1} = X_cT^{-1}$ both solve (22). From the Hamiltonian structure, we further have spec($\Phi_-$) = spec($\Phi_+$). Thus, using span(•) to denote the span (image) of a matrix

$$\text{span} \left( \begin{bmatrix} X_{11} \\ TX_{21} \end{bmatrix} \right) = \text{span} \left( \begin{bmatrix} TX_{22} \\ X_{12} \end{bmatrix} \right). \tag{29}$$

which implies that $TX_o = X_cT^{-1} =$: $X_{co}$ is a solution to (22) found by identifying the stable invariant subspace of $H_{co}$. The choice of the actual basis is immaterial as long as the spectrum restricted to it is the same, and from this, the uniqueness of $X_{co}$ follows. Such invariant subspace identification for solving $X_{co}$ is usually done by the Schur decomposition which, in general, requires $O((2n)^3)$ work and $O((2n)^2)$ memory space.

Recalling (6) and (7), we generalize “stabilizability” in this symmetric-system cross-Riccatian sense. By factoring out $X_{co}$ from both sides in (22) for the coefficient matrices and by noting $X_{co} = X_cT^{-1} = TX_o$ and (20), we get

$$A + X_{co}BC = A_c \quad \text{and} \quad A + BCX_{co} = A_o. \tag{30}$$

It is readily seen that spec($A_c$) = spec($A_o$) = spec($\Phi_-$) \subset \mathbb{C}_-. \quad \text{and therefore, we call } X_{co} \quad \text{the stabilizing solution to (22)} \quad \text{analogously. Similar to the results in Section II-B, we have}

$$X_{co}^2 = X_cX_o \tag{31}$$

and the actual $T$ in (20) or (21) is immaterial. With the same notion as in Section II-A, the effect of the similarity transform on $X_{co}$ in fact, for arbitrary $T_R$ and $T_L = T_R^{-1}$ is

$$X_{co} \rightarrow T_LX_{co}T_R \tag{32}$$

thus, spec($X_{co}$) is invariant. Since there exists a similar system in which $X_c$ and $X_o$ are both diagonal [cf. (12)], we have

$$|\lambda_i(X_{co})| = \sigma_i, \quad i = 1, 2, \ldots, n \tag{33}$$

where $\sigma_i$’s are the singular values in (9), and $\lambda_i(\cdot)$ denotes the $i$th eigenvalue in descending magnitude without loss of generality. By contrasting with Section II-B, it can be concluded that $X_{co}$ plays the same role in the PRBT as $W_{co}$ does in the standard BT. A difference is that the input and output energy measures in the PRBT, by noting that $u$ and $y$ are power conjugated, are $\int_0^\infty u^Tydt$ and $-\int_0^\infty u^Tydt$, respectively (see [25] for the physical interpretation).

B. Constructing the Projection Matrices

The system balancing in the PRBT, corresponding to (8), (9), (10), and (11), can sometimes be numerically ill conditioned. This is particularly so for large-scale systems with nearly singular $X_c$ and/or $X_o$, i.e., some states are nearly uncontrollable and/or unobservable [6], [11], [14]. To avoid ill conditions, we borrow results from [6] on the standard BT and give an alternative way for obtaining a reduced-order model with the same transfer function as the PRBT-reduced model. From (31)

$$\text{span}(X_{co}) = \text{span}(X_c), \quad \text{span} \left( X_{co}^T \right) = \text{span}(X_o). \tag{34}$$

Therefore, the cross Riccatian $X_{co}$ carries both the controllable and observable subspaces. By (14) and (31)

$$X_{co}^2 = [T_{Rb} \quad T_{Rs}] \begin{bmatrix} \Sigma_{c2}^2 & 0 \\ 0 & \Sigma_{c2}^2 \end{bmatrix} [T_{Lb} \quad T_{Ls}] \tag{35}$$

Knowing (33), $X_{co}$ is block diagonalized into $V^{-1}X_{co}V = \text{diag} \left( X_{co}^{(1)} \quad X_{co}^{(2)} \right)$, where $V \in \mathbb{R}^{n \times n}$ is nonsingular [this $V$ should not be confused with the one in (8)]. Here, $X_{co}^{(1)} \in \mathbb{R}^{r \times r}$ and $X_{co}^{(2)} \in \mathbb{R}^{(n-r) \times (n-r)}$ hold the bigger and smaller eigenvalues, in terms of magnitude, respectively. This block
diagonalization generally takes the $O(n^3)$ work. Next, partition
the columns of $V$ and the rows of $W = V^{-1}$ compatibly
\[ X_{co} = [V_b \ V_s \begin{bmatrix} X_b^T & 0 \\ 0 & X_s^T \end{bmatrix}] [W_b \ W_s]. \] (36)

We have that $T_{Rb}$ and $V_b$ span the same (right) eigenspace
corresponding to $X_{co}$, whereas $T_{Rs}$ and $V_s$ span the same
(right) eigenspace corresponding to $X_{so}$. Therefore, there exist
nonsingular $M_1 \in \mathbb{R}^{r \times r}$ and $M_2 \in \mathbb{R}^{(n-r)\times(n-r)}$ such that
\[ [T_{Rb} \ T_{Rs}] = [V_b \ V_s \begin{bmatrix} M_1 & 0 \\ 0 & M_2 \end{bmatrix}]. \] (37)

By taking inverse on both sides, we also get
\[ \begin{bmatrix} T_{Lb} \\ T_{Ls} \end{bmatrix} = \begin{bmatrix} M_1^{-1} & 0 \\ 0 & M_2^{-1} \end{bmatrix} \begin{bmatrix} W_b \\ W_s \end{bmatrix}. \] (38)

By using (37) and (38), the reduced-order model by projection
onto the “more significant” subspaces of (36)
\[ (\tilde{A}, \tilde{B}, \tilde{C}, D) = (W_b A_0 V_b, W_b B_0, C_0 V_b, D_0) \] (39)
is easily shown to be similar to (15), thereby producing the same
transfer function as the PRBT-reduced model. In other
words, an equivalent PRBT-reduced model can be obtained
from the block diagonalization of $X_{co}$ without actually bal-
ancing the system. To make this paper self-contained, such
diagonalization process is included in Appendix I.

IV. QADI-BASED SYMMETRIC-SYSTEM PRBT

The recent work in [8]–[10] generalizes the ADI iteration
[26]–[28], which is a highly efficient solver for the Lyapunov
equations, to a QADI algorithm for the fast solution of AREs.
Consider the ARE
\[ A^T X + X A + X B B^T X + C^T C = 0 \] (40)
where $A$ is assumed stable, and a stabilizing solution $X \geq 0$
exists such that $\text{spec}(A + B B^T X) \subset \mathbb{C}_-$. The bounded real
lemma [1] states that such an $X$ exists if and only if
$\sup \sigma(C(j\omega - A)^{-1} B) < 1, \forall \omega \in \mathbb{R}$, where $\sigma(\cdot)$
denotes the largest singular value. In its original form, the QADI solves (40)
by iterating on two half-steps
\[ \begin{align*}
(A^T + X_{j-1}^T B B^T + p_j I) X_{j-\frac{1}{2}} &= -C^T C - X_{j-1}^T (A - p_j I) \\
\left( A^T + X_{j-\frac{1}{2}} B B^T + p_j I \right) X_j &= -C^T C - X_{j-\frac{1}{2}} (A - p_j I)
\end{align*} \] (41a)
where $X_0 = 0$, and the shifts $p_j \in \mathbb{C}_-, j = 1, 2, \ldots$, are either
real or complex conjugate pairs. For complex conjugate $p_j$’s,
real-value arithmetic, and thus higher numerical efficiency and
robustness, is maintained by combining two runs of (41) into
one [9]. Furthermore, it has been shown that $X_j$ converges to the
stabilizing $X$ superlinearly [8]–[10].

A. XRE Solution Via XQADI

By assuming the existence of a stabilizing solution to (22)
and by comparing it to (40), we modify QADI to XQADI that
solves an XRE with the following two half-steps:
\[ \begin{align*}
(A^T + X_{j-1}^T (BC) + p_j I) X_{j-\frac{1}{2}} &= -(BC) X_{j}^T - X_{j-1}^T (A - p_j I) \\
(A + X_{j-\frac{1}{2}} BC + p_j I) X_j &= -BC - X_{j-\frac{1}{2}} (A - p_j I)
\end{align*} \] (42a)
where $X_0 = 0$, and $p_j \in \mathbb{C}_-, j = 1, 2, \ldots$, are either real or
complex conjugate pairs. For complex conjugate pairs, the
XQADI reduces the storage requirements to $O(n^2)$ memory space.

The convergence of $X_j$ to $X_{co}$ in XQADI can be drawn
from that of QADI [8], [10]. To see this, we note that the
QADI computes the stabilizing solution, which is denoted by

---

Authorized licensed use limited to: The University of Hong Kong. Downloaded on June 5, 2009 at 00:54 from IEEE Xplore. Restrictions apply.
where $K_c$ is a positive constant, and $\| \cdot \|$ denotes the Frobenius norm. To see this, basic matrix algebra dictates that $\text{spec}(Z_R(j)) \cup \{0\} = \text{spec}(Z_L(j))$, and with reference to (36), the QXADI should be terminated once the eigenvalue update is smaller than a preset tolerance. By using (47) and similar techniques as in [29], it is straightforward to show that $Z_R(j)$ and $Z_L(j)$ usually converge in just several to tens of runs of (49) (see numerical examples later); thus, these low-rank factors can provide further computational and memory savings [14].

The matrix inverse operation in $S_j$, assuming that no structures are exploited, still costs $O(n^3)$ work and constitutes the most expensive step in the LRXQADI, but other than that, the remaining operations are just matrix vector multiplications and small size matrix inverses. Moreover, the factorized iterates consume only an $O(mn)$ memory space, thus representing a significant reduction over the invariant subspace and XQADI approaches.

### C. Projection Matrices From Low-Rank Factors

Suppose that $Z_L(j)$ and $Z_R(j)$ in LRXQADI converge to $Z_L$ and $Z_R$, respectively. Then, $X_{co}$ can be numerically equated to, or closely approximated by, the product $Z_L Z_R$, where $Z_L \in \mathbb{R}^{n \times k}$ and $Z_R \in \mathbb{R}^{k \times n}$ and, often, $k \ll n$. In other words

$$\text{span}(X_{co}) = \text{span}(Z_L), \quad \text{span}(X_{co}^T) = \text{span}(Z_R^T).$$

This section presents an effective way, by exploiting the low-rank nature of $Z_L$ and $Z_R$, to construct the projection matrices without actual balancing. From the results in Section III-B, the projection bases are, in fact, the dominant (left and right) eigenspaces of $X_{co}$, or, equivalently, those of $X_{co}^T$. Suppose that the cross product $Z_R Z_L \in \mathbb{R}^{k \times k}$ is block diagonalized, using the same mechanics as in Section III-B and Appendix I (but with the much less $O(k^3)$ work), such that

$$Z_R Z_L = \tilde{V} \text{diag} \left( \hat{X}_{co}^b, \hat{X}_{co}^b \right) \tilde{W},$$

where $\tilde{W} = \tilde{V}^{-1}$, and $\tilde{V}_b \hat{X}_{co}^b \tilde{W}_b$ represents the “more significant” eigendecomposition of $Z_R Z_L$. Then

$$X_{co}^2 = Z_L Z_R Z_L Z_R \approx Z_L \tilde{V}_b \left( \hat{X}_{co}^b \right)^{-1} \left( \hat{X}_{co}^b \right)^2 \tilde{W}_b Z_R.$$
Note that since \( \text{spec}(Z_{R}^{(j)} Z_{L}^{(j)} Z_{R}^{(j)}) \cup \{0\} = \text{spec}(Z_{L}^{(j)} Z_{R}^{(j)}) \), \((X_{R}^{(j)})^{2}\) is holding the “more significant” eigenvalues of \(X_{R}^{(j)}\). By defining \(W_{R}^{(j)} := \tilde{W}_{R} Z_{R}\) and \(V_{R}^{(j)} := Z_{L} \tilde{V}_{L} (X_{R}^{(j)})^{-1}\), it is straightforward to verify that \(W_{R}^{(j)} V_{R}^{(j)} = I\). Thus, the projection matrices corresponding to the dominant eigenspaces of \(X_{R}^{(j)}\) are captured in \(W_{R}^{(j)}\) and \(V_{R}^{(j)}\), respectively. Moreover, similar to (39), the PRBT-reduced model can be obtained by

\[
(\hat{A}, \hat{B}, \hat{C}, D) = (W_{R}^{(j)} A_{0} V_{R}^{(j)}, W_{R}^{(j)} B_{0}, C_{0} V_{R}^{(j)}, D_{0}).
\]  

(53)

Therefore, besides the reduced computation and storage due to the low-rank iterates in the LRXQADI, further savings can be achieved in the projection phase because usually, only a low-dimensional matrix decomposition is required in (51) in contrast to the full decomposition in (36).

V. Numerical Examples

In the context of symmetric systems, the proposed XRE-based PRBT approaches described in Sections III and IV are implemented and compared with the conventional PRBT realizations wherein two AREs are solved. All experiments are done in the Matlab R2006a environment and executed on a 1.8-GHz notebook with 2-GB memory. The AREs are solved with the Matlab routine \texttt{aresolv} with \texttt{eigen} flag enabled (generally faster than its \texttt{schur} option) and also the SLICOT Fortran routine \texttt{slcares} (which is faster than its generalized Schur counterpart \texttt{slcaregs}) invoked through a Matlab gateway. Moreover, the recent PRBT/CFQADI flow [8], [10], in which the dual AREs are solved by the CFQADI (coded with Matlab m-script files), is contrasted. For the proposed XRE-based PRBT flow, we first solve (22) by identifying the transmission-line model on the order of 256, the second is nontrivial algorithms [31].

We try out five real-life examples [7], [8]. The first is a transmission-line model on the order of 256, the second is a spiral inductor on the order of 500, and the rest are \(RLC\) ladders on the orders of 800, 2000, and 3000. The CPU times for various PRBT implementations are shown in Table I. We remark that ARE or XRE solutions are unavailable in some high-order cases due to memory exhaustion, and in the XQADI approach, the large-size Matlab matrix multiplications and inversions are too slow that convergence cannot be sought in practical time. Since the ARE solvers \texttt{slcares} and \texttt{aresolv} and the invariant subspace method are all based on the Hamiltonian matrix or alike, it is not surprising that the latter is the fastest among the three as only one matrix equation needs to be solved. In fact, when the time measurements are available, the PRBT/XRE/invar time is approximately half that of the standard PRBT/ARE/\texttt{aresolv} flow. Nonetheless, these Hamiltonian-based schemes still suffer from a prohibitive computation compared with PRIMA, particularly for large systems. Generally, faster PRBT times are observed in the PRBT/XRE/XQADI method, but they are still relatively expensive and do not scale well with model orders.

The real breakthrough of the PRBT complexity barrier actually comes from the low-rank implementations, namely, CFQADI and LRXQADI, primarily due to their utilization of the low-rank iterates and the fast-decaying singular values inherent in many physical systems. From Table I, it can be seen that the speed gain of PRBT/XRE/LRXQADI is more significant for large systems, and in the third example, its speed is even faster than that of PRIMA. This is inferred by Fig. 1 which shows the quick attenuating cross-Riccation singular values [cf. (33)] in all the test examples. In our experiments, we terminate CFQADI, XQADI, and LRXQADI once the update of singular values (or eigenvalue magnitudes) drops below \(10^{-5} \sim 10^{-10}\); thus, the number of iterations and ranks of the low-rank factors is consistently on the order of tens. Fig. 2 shows \(|X_{j} - X_{\infty}|_{F}/|X_{\infty}|_{F}\) [see (47)] for the XRE solution by XQADI or, equivalently, LRXQADI in some test examples.

![Table I](image)

**Table I**

<table>
<thead>
<tr>
<th>Implementation</th>
<th>Time (s)</th>
<th>speedup (times)</th>
<th>rank (AI)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1×ARE</td>
<td>3.20</td>
<td>1.84</td>
<td>6.16</td>
</tr>
<tr>
<td>2×ARE</td>
<td>4.36</td>
<td>2.02</td>
<td>4.84</td>
</tr>
<tr>
<td>3×ARE</td>
<td>6.47</td>
<td>2.01</td>
<td>2.33</td>
</tr>
<tr>
<td>1×XRE</td>
<td>1.86</td>
<td>2.01</td>
<td>0.47</td>
</tr>
<tr>
<td>2×XRE</td>
<td>3.78</td>
<td>2.01</td>
<td>0.78</td>
</tr>
<tr>
<td>3×XRE</td>
<td>5.80</td>
<td>2.01</td>
<td>1.50</td>
</tr>
<tr>
<td>PRIMA</td>
<td>0.13</td>
<td>2.01</td>
<td>0.64</td>
</tr>
</tbody>
</table>

† The brackets are time breakdowns: (top) time for solving 2×AREs or 1×XRE; (middle) time for computing the PRBT projection matrices. N/A means out of memory or impractically slow convergence.
issue, however, remains with every numerical implementation of iterative algorithms for the PRBT. Indeed, even with “noniterative” algorithms like the Hamiltonian-based approaches, there are inner iterations which may terminate early in case of ill-conditioned data, resulting in potential problems. This is why passivity check and enforcement, if necessary [32], [33], are still required in practical implementations. To this end, we note the following.

a) The passivity “check” is usually fast because only the low-order reduced models are involved. The conventional checking procedure is to form a Hamiltonian matrix and then search for imaginary axis eigenvalues [32], [33].

b) The passivity “enforcement” is much more expensive as it involves finding regions of passivity violation and performing passivity compensation, for example, by first-order perturbation or convex programming [32]–[34]. Moreover, the compensation process will always introduce errors to the approximation accuracy of the reduced-order model and, therefore, should be kept to be as sparing as possible.

Consequently, for algorithms that do not tend to guarantee passivity [like standard BT, Padé via Lanczos, etc.], there can be multiple frequency bands of passivity violation. The “enforcement” step must be invoked iteratively to ensure a passive model at least for the frequencies of interest, and then, the approximation error thus incurred can be high. The PRBT theoretically guarantees a passive reduced model and tends to achieve so in practice. In fact, in our numerical examples, all the reduced-order models are found to be strictly passive. Therefore, in most PRBT implementations, even when nonpassivity is present, passivity enforcement requires only minor correction, and the disturbance on approximation accuracy would be small.

2) For simplicity, only a single shift has been used in CFQADI, XQADI, and LRXQADI in our examples. Referring to (48), we have chosen $p = -\sqrt{\rho(H_{co})(\rho(H_{co})^{-1})}$ [7], [9], where $\rho(\cdot)$ denotes spectral radius. When computing the shifts $p_i$’s, one often makes use of the power iteration, for example, by $(2n)^2$. We can also make use of the matrix inversion lemma to verify that

$$H_{co}^{-1} = \begin{bmatrix} I & -A^{-1}BC \end{bmatrix} \begin{bmatrix} A^{-1}S^{-1} & 0 \\ 0 & I \end{bmatrix} \begin{bmatrix} I & 0 \\ BCA^{-1} & I \end{bmatrix}$$

$$S^{-1} = (-A + BCA^{-1}BC)^{-1} = -A^{-1} - A^{-1}BCA^{-1}B \times (I - CA^{-1}BCA^{-1}B)^{-1}CA^{-1}$$

thus allowing the power iteration of $H_{co}^{-1}$ to be done efficiently with only an $O(n^2)$ work.
3) In practice, the time for PRBT/LRXQADI is approximately of the same order as that for PRIMA, but the reduced-order model from the PRBT always has superior accuracy. Although a two-stage reduction based on PRIMA and then PRBT is possible [3], [7], any first-stage reduction by PRIMA would cause irremediable loss of approximation accuracy (because PRBT can then at most preserve the fidelity of the intermediate PRIMA-reduced model), and it is hard to determine the intermediate order to switch from PRIMA to PRBT. Given that PRBT/LRXQADI largely reduces the PRBT computation to an extent comparable to PRIMA, it is justifiable to deploy PRBT/LRXQADI directly on the initial model.

4) PRBT/LRXQADI scales well with the problem size as the termination of LRXQADI is determined by the drop of singular values (eigenvalue magnitudes of the cross Riccatian) pertinent to a physical system, and it is fairly independent of the modeling order. In the same vein, the XRE and the LRXQADI formulations also provide a theoretical tool for analyzing the eigenvalue decay in the cross Riccatian, whose physical importance parallels its Lyapunov counterpart [35].
This paper has presented a computationally efficient cross-Riccatian PRBT implementation applicable to symmetric MIMO systems such as linear passive electrical networks. Instead of solving two AREs in the conventional PRBT, only one XRE has to be solved, which results in good numerical robustness and reduced computation and storage. Theoretical connection between the XRE and the PRBT has been elaborated. An invariant subspace approach, an iterative XQADI algorithm, and an LRXQADI scheme have been described to efficiently solve the XRE. A simple construction of the PRBT projection matrices through the Schur block diagonalization has been adapted to avoid the possibly ill-conditional PR balancing. Application examples have demonstrated the remarkable efficacy of the PRBT/XRE/LRXQADI integration over the traditional PRBT realizations.

VI. CONCLUSION

It can be easily verified that

\[
X_{co} = [Q_b \quad Q_s] \begin{bmatrix}
X_{co}^b & \Omega \\
0 & X_{co}^s
\end{bmatrix} \begin{bmatrix}
Q_b^T \\
Q_s^T
\end{bmatrix}
\]

Therefore, with respect to (36), the projection matrices are \( V_b = Q_b \) and \( W_b = Q_b^T - \Gamma Q_s^T \), respectively. Moreover, by using the notations as in Section III-B and supposing that \( X_c, X_o \in \mathbb{R}^{r \times r} \) are the controllability and observability Riccatians of \( (A, B, C, D) \) in (39), some interesting properties in the standard BT [6] can be generalized to this cross-Riccatian framework

\[
\begin{align*}
\tilde{X}_c &= M_1 \Sigma_b M_1^T = W_b X_c W_b^T \\
\tilde{X}_o &= M_1^{-T} \Sigma_b M_1^{-1} = V_b^T X_o V_b.
\end{align*}
\]

The proof is similar to that in [6] and is therefore omitted.

APPENDIX II

POSITIVE-SEMIDEFINITENESS PROOF IN LRXQADI

Referring to the LRXQADI steps in Section IV-B and (49), we now show that \( Z_{L}(j^{-1}) M_{11L}^{(j)} M_{11R}^{(j)} Z_{L}(j^{-1}) \geq 0 \), \( j = 2, 3, \ldots \). Obviously, it is sufficient to prove that \( (M_{11L}^{(j)})^T Z_{L}(j^{-1}) = M_{11R}^{(j)} Z_{L}(j^{-1}) \). Again, using (21) on (49), we have for all \( j \)’s

\[
\begin{align*}
(M_{11L}^{(j)})^T &= \sqrt{-2p_j} (I - CS_j B B^T S_j^T C^T)^{-\frac{1}{2}} B^T S_j T \\
&= \sqrt{-2p_j} (I - CS_j B B^T S_j^T C^T)^{-\frac{1}{2}} CTT^{-1} S_j T \\
&= M_{11R}^{(j)} T
\end{align*}
\]
and similarly
\[
\left( M_{12}^{(j)} \right)^T = T^{-1} M_{12}^{(j)} T. \tag{59}
\]
Since \( Z_L^{(1)} = M_{11L}^{(1)} \) and \( Z_R^{(1)} = M_{11R}^{(1)} \), we have \( (Z_L^{(1)})^T = T^{-1} Z_L^{(1)} \).

By using the aforementioned results, it is easy to prove that for \( j = 2, (M_{12}^{(1)} Z_L^{(1) T} = M_{11R}^{(2)} Z_L^{(1)}) \). Subsequently, we have \( (Z_L^{(1)})^T M_{12}^{(1)} M_{11R}^{(2)} Z_L^{(1)} \geq 0 \) and
\[
\left( Z_L^{(2)} \right)^2 = 
\left( M_{12}^{(1)} \right)^T \left( Z_L^{(1) T} \right)^2 \left( I - Z_L^{(1)} M_{11L}^{(2)} M_{11R}^{(2)} Z_L^{(1)} \right)^{-\frac{1}{2}}
\]
which, in turn, implies that \( (M_{11L}^{(3)} Z_L^{(2)^T} = M_{11R}^{(3)} Z_L^{(2)^T} \) and \( Z_L^{(2) M_{11L}^{(3)} M_{11R}^{(3)} Z_L^{(2)^T} \geq 0 \). The proof continues this way and extends to all \( j \).

References