



MAX-PLANCK-GESELLSCHAFT

**Max Planck Institute Magdeburg
Preprints**

Peter Benner¹ Matthias Heinkenschloss² Jens Saak¹
Heiko K. Weichelt¹

**Inexact Low-Rank Newton-ADI Method
for Large-Scale Algebraic Riccati Equations**



Affiliations:

¹ Research Group Computational Methods
in Systems and Control Theory (CSC),
Max Planck Institute for Dynamics of
Complex Technical Systems Magdeburg,
Sandtorstr. 1,
39106 Magdeburg, Germany
(`{benner,saak,weichelt}@mpi-magdeburg.mpg.de`)

² Department of Computational and
Applied Mathematics (CAAM),
Rice University,
MS-134, 6100 Main Street,
Houston, TX 77005-1892, USA
(`heinken@rice.edu`)

The research of this author was supported in part by grants AFOSR FA9550-12-1-0155
and NSF DMS-1115345

Corresponding author:

Heiko K. Weichelt¹
`weichelt@mpi-magdeburg.mpg.de`

Imprint:

Max Planck Institute for Dynamics of Complex Technical Systems, Magdeburg

Publisher:

Max Planck Institute for
Dynamics of Complex Technical Systems

Address:

Max Planck Institute for
Dynamics of Complex Technical Systems
Sandtorstr. 1
39106 Magdeburg

<http://www.mpi-magdeburg.mpg.de/preprints/>

Abstract

This paper improves the inexact Kleinman-Newton method by incorporating a line search and by systematically integrating the low-rank structure resulting from ADI methods for the approximate solution of the Lyapunov equation that needs to be solved to compute the Kleinman-Newton step. A convergence result is presented that tailors the convergence proof for general inexact Newton methods to the structure of Riccati equations and avoids positive semi-definiteness assumptions on the Lyapunov equation residual, which in general do not hold for low-rank approaches. On a test example, the improved inexact Kleinman-Newton method is seven to twelve times faster than the exact Kleinman-Newton method without line search; the addition of the line search to the inexact Kleinman-Newton method alone can reduce computation time by up to a factor of two.

1 Introduction

We present improvements of the inexact Kleinman-Newton method for the solution of large-scale continuous algebraic Riccati equations (CARE)

$$\mathcal{R}(X) = C^T C + A^T X + X A - X B B^T X = 0 \quad (1.1)$$

with $C \in \mathbb{R}^{p \times n}$, $A \in \mathbb{R}^{n \times n}$, $X = X^T \in \mathbb{R}^{n \times n}$, $B \in \mathbb{R}^{n \times r}$, and $p + r \ll n$. The algorithmic improvements consist of incorporating a line search and of systematically integrating the low-rank structure resulting from ADI methods for the approximate solution of the Lyapunov equation

$$(A^{(k)})^T X^{(k+1)} + X^{(k+1)} A^{(k)} = -C^T C - X^{(k)} B B^T X^{(k)}, \quad (1.2)$$

where

$$A^{(k)} = A - B B^T X^{(k)},$$

which has to be approximately solved in the k -th iteration. The paper is motivated by the recent work of Feitzinger et al. [10] who propose and analyze inexact Kleinman-Newton methods without line search, by Benner and Byers [3] who incorporate line search into the exact Kleinman-Newton method, and by the recent work of Benner et al. [4, 6] on algorithmic improvement of low-rank ADI methods. We improve on [10] by adding a line search and systematically integrating the low-rank structure resulting from ADI methods into the algorithm. The convergence result in [10] makes positive semi-definiteness assumptions on the Lyapunov equation residual that are in general not valid when the Lyapunov equation is solved using ADI. Our convergence result follows that of general inexact Newton methods, but uses the structure of Riccati equations. We add the inexact solution of the Lyapunov equation to [3] and incorporate the low-rank structure.

On a test example resulting from the finite difference approximation of an LQR problem governed by an advection diffusion equation, our improved inexact Kleinman-Newton method is seven to twelve times faster than the exact Kleinman-Newton method without line search. The addition of the line search to the inexact Kleinman-Newton method alone can reduce computation time by up to a factor of two, although,

as already observed in [3], in our experiments the line search is used at most in the first two iterations.

The paper is organized as follows. In the next section we recall a basic existence and uniqueness result for symmetric positive semi-definite stabilizing solution of the CARE (1.1). Section 3 introduces the inexact Kleinman-Newton method with line search and presents the basic convergence result. The basic ingredients of the ADI methods that are needed for this paper are reviewed in Section 4. Section 5 discusses the efficient computation of various quantities like Newton residual using the low-rank structure. As a result, computational cost of our overall algorithm is proportional to the total number of ADI iterations used; in comparison the cost of other components, such as execution of the line search, are negligible. Finally, we demonstrate the contributions of the various improvements on the overall performance gains in Section 6. As mentioned before, in our numerical tests, our improved inexact Kleinman-Newton method is seven to twelve times faster than the exact Kleinman-Newton method without line search.

Notation. Throughout the paper we consider the Hilbert space of matrices in $\mathbb{R}^{n \times n}$ endowed with the inner product $\langle M, N \rangle = \text{tr}(M^T N) = \sum_{i,j=1}^n M_{ij} N_{ij}$ and the corresponding (Frobenius) norm $\|M\|_F = (\langle M, M \rangle)^{1/2} = (\sum_{i,j=1}^n M_{ij}^2)^{1/2}$. Furthermore, given real symmetric matrices M, N we write $M \succeq N$ if and only if $M - N$ is positive semi-definite, and $M \succ N$ if and only if $M - N$ is positive definite.

2 The Riccati Equation

We recall an existence and uniqueness result for the continuous Riccati equation (1.1).

Definition 1 Let $A \in \mathbb{R}^{n \times n}$, $B \in \mathbb{R}^{n \times r}$, and $C \in \mathbb{R}^{p \times n}$. The pair (A, B) is called stabilizable if there exists a feedback matrix $K \in \mathbb{R}^{n \times r}$ such that $A - BK^T$ is stable, which means that $A - BK^T$ has only eigenvalues in the negative half complex plane \mathbb{C}^- . The pair (C, A) is called detectable if (A^T, C^T) is stabilizable.

Notice that (A, B) is stabilizable if and only if (A, BB^T) is stabilizable and (C, A) is detectable if and only if $(C^T C, A)$ is detectable.

Assumption 2 The matrices $A \in \mathbb{R}^{n \times n}$, $B \in \mathbb{R}^{n \times r}$, and $C \in \mathbb{R}^{p \times n}$ are given such that (A, B) is stabilizable and (C, A) is detectable.

If Assumption 2 holds, there exists a unique symmetric positive semi-definite solution $X^{(*)}$ of the CARE (1.1) which is also the unique stabilizing solution. This follows from Theorems 8.5.1 and 9.1.2 (see also p. 244) in [16].

Furthermore, it can be shown that all symmetric positive semi-definite solutions of the CARE (1.1) are stabilizing.

Theorem 3 If Assumption 2 holds, every symmetric solution $X^{(*)} \succeq 0$ of the CARE (1.1) is stabilizing.

Proof. Let $X = X^T \succeq 0$ solve the CARE (1.1). We show that $A - BB^T X$ is stable by contradiction.

Assume that μ is an eigenvalue of $A - BB^T X$ with $\text{Re}(\mu) > 0$ and let $v \in \mathbb{C}^n \setminus \{0\}$ be a corresponding eigenvector. The CARE (1.1) can be written as

$$(A - BB^T X)^T X + X(A - BB^T X) = -C^T C - XBB^T X. \quad (2.1)$$

Multiply (2.1) with v^T from left and v from the right. The left-hand side of (2.1) yields

$$2 \text{Re}(\mu) v^T X v \geq 0, \text{ since } X = X^T \succeq 0,$$

and the right-hand side of (2.1) yields

$$-v^T C^T C v - v^T XBB^T X v \leq 0, \text{ since } C^T C \succeq 0 \text{ and } XBB^T X \succeq 0.$$

Hence, left- and right-hand sides of (2.1) multiplied by v^T from left and v from the right are equal to zero that is $v^T X v = 0$ and $v^T C^T C v + v^T XBB^T X v = 0$ which yields $Cv = 0$ and $B^T X v = 0$. Since $(A - BB^T X)v = \mu v$, v is an eigenvector of A with eigenvalue μ and $\text{Re}(\mu) > 0$.

The Hautus-Popov Test for Detectability [13, p. 80-8] states that (C, A) is detectable if and only if $Ax = \lambda x$, $x \neq 0$ and $\text{Re}(\lambda) \geq 0$ implies $Cx \neq 0$. Thus, the existence of $v \neq 0$ and $\text{Re}(\mu) > 0$ with $Av = \mu v$ contradicts the detectability of (C, A) by the Hautus-Popov test. □

3 The Inexact Kleinman-Newton Method with Linear Search

This section introduces the inexact Kleinman-Newton method with line search and gives a convergence result. The fundamental ideas are identical to what is well known for inexact Newton methods, see, e.g., Kelley [14, Sec. 8.2], but are tailored to the structure of the Riccati equations. The presentation of the basic algorithm in Section 3.1 combines ideas from general inexact Newton methods, from Kleinman-Newton with inexactness, see, e.g., Feitzinger et al. [10], and Kleinman-Newton with line search, see, e.g., Benner and Byers [3]. In Section 3.3, we will show that the assumptions made in Feitzinger et al. [10], to ensure convergence of the inexact Kleinman Newton method, are in general not valid if low-rank Lyapunov solvers are used to compute the inexact Kleinman-Newton step, and we will present an alternative convergence result that follows more closely that of general inexact Newton methods.

3.1 Derivation of the Method

We want to compute the symmetric, positive semi-definite, stabilizing solution $X^{(*)}$ of the CARE (1.1). The operator $\mathcal{R} : \mathbb{R}^{n \times n} \rightarrow \mathbb{R}^{n \times n}$ defined in (1.1) is twice F chet differentiable with derivatives given by

$$\mathcal{R}'(X)N = (A - BB^T X)^T N + N(A - BB^T X), \quad (3.1a)$$

$$\mathcal{R}''(X)(N_1, N_2) = -N_1 BB^T N_2 - N_2 BB^T N_1. \quad (3.1b)$$

Since \mathcal{R} is quadratic in X , the 2nd order F chet derivative is independent of X and $\mathcal{R}(Y)$ can be expressed via a Taylor series as

$$\mathcal{R}(Y) = \mathcal{R}(X) + \mathcal{R}'(X)(Y - X) + \frac{1}{2}\mathcal{R}''(X)(Y - X, Y - X). \quad (3.2)$$

The CARE (1.1) can be solved using Newton's method, which in this context is referred to as the Kleinman-Newton method [15]. Given an approximate symmetric solution $X^{(k)}$ of (1.1), the new Kleinman-Newton iterate is given by

$$\mathcal{R}'(X^{(k)})X^{(k+1)} = \mathcal{R}'(X^{(k)})X^{(k)} - \mathcal{R}(X^{(k)}). \quad (3.3)$$

Equation (3.3) is the Lyapunov equation (1.2). Instead of solving (3.3) for the new iterate, one could solve $\mathcal{R}'(X^{(k)})S^{(k)} = -\mathcal{R}(X^{(k)})$ for the step $S^{(k)} = X^{(k+1)} - X^{(k)}$. While the latter equation may be favorable in cases where the Lyapunov equation is solved using direct methods (see, e.g., [3, p. 101]), (3.3) is favorable when the Lyapunov equation is solved inexactly using iterative methods. The right hand side can be written as $-GG^T$, where $G = [C^T \mid X^{(k)}B] \in \mathbb{R}^{n \times (p+r)}$, which, as we will see later, is important when the Kleinman-Newton method is combined with low-rank approximation methods. Expressions of $\mathcal{R}(X^{(k)})$ which lend themselves to the application of low-rank approximation methods, and which are equal to $\mathcal{R}(X^{(k)})$ in the exact Kleinman-Newton method, fail when used in an inexact Kleinman-Newton method as shown in Feitzinger et al. [10].

If Assumption 2 holds, then the special structure of \mathcal{R} allows one to prove global convergence of the Kleinman-Newton method: If the initial iterate $X^{(0)}$ is symmetric and stabilizing, then the Kleinman-Newton method is well defined (i.e., (1.2) has a unique solution), and the iterates generated by the Kleinman-Newton method converge with a q-quadratic convergence rate, and satisfy $X^{(1)} \succeq X^{(2)} \succeq \dots \succeq X^{(*)} \succeq 0$; see, e.g., Kleinman [15] or Lancaster and Rodman [16, Sec. 9.2].

Even though the Kleinman-Newton method exhibits global convergence, it was observed by Benner and Byers [3] that a line search improves its efficiency. Especially in the first iteration of the Kleinman-Newton method the residual may increase dramatically if no line search is used. For large scale problems the Newton equation (the Lyapunov equation) (3.3) is solved iteratively, and the residual error in the Lyapunov equation has to be controlled appropriately to ensure convergence. We integrate the inexact solution of (3.3) and a line search into the Kleinman-Newton method. As we have mentioned before, the fundamental ideas are identical to what is well known for inexact Newton methods, see, e.g., Kelley [14, Sec. 8.2].

Given a symmetric $X^{(k)} \in \mathbb{R}^{n \times n}$ and $\alpha > 0$, $\eta_k \in (0, 1)$ we compute a symmetric step $S^{(k)} \in \mathbb{R}^{n \times n}$ with

$$\left\| \mathcal{R}'(X^{(k)})S^{(k)} + \mathcal{R}(X^{(k)}) \right\|_F \leq \eta_k \left\| \mathcal{R}(X^{(k)}) \right\|_F \quad (3.4)$$

and then compute the next iterate

$$X^{(k+1)} = X^{(k)} + \lambda_k S^{(k)},$$

where the step size $\lambda_k > 0$ such that the sufficient decrease condition

$$\left\| \mathcal{R} \left(X^{(k)} + \lambda_k S^{(k)} \right) \right\|_F \leq (1 - \lambda_k \alpha) \left\| \mathcal{R} \left(X^{(k)} \right) \right\|_F \quad (3.5)$$

is satisfied and the step size λ_k is not unnecessarily small.

If we define the Newton step residual

$$\mathcal{R}'(X^{(k)})S^{(k)} + \mathcal{R}(X^{(k)}) = L^{(k+1)}, \quad (3.6)$$

then (3.4) reads

$$\left\| L^{(k+1)} \right\|_F \leq \eta_k \left\| \mathcal{R}(X^{(k)}) \right\|_F. \quad (3.7)$$

Using the definition (1.1), (3.1a), and

$$\tilde{X}^{(k+1)} = X^{(k)} + S^{(k)},$$

the equation (3.6) is equivalent to

$$(A^{(k)})^T \tilde{X}^{(k+1)} + \tilde{X}^{(k+1)} A^{(k)} = -X^{(k)} B B^T X^{(k)} - C^T C + L^{(k+1)} \quad (3.8a)$$

and the new iterate is

$$X^{(k+1)} = (1 - \lambda_k) X^{(k)} + \lambda_k \tilde{X}^{(k+1)}. \quad (3.8b)$$

The Riccati residual at $X^{(k+1)} = X^{(k)} + \lambda_k S^{(k)}$ can be expressed using (3.2) and (3.6) as

$$\begin{aligned} \mathcal{R}(X^{(k)} + \lambda_k S^{(k)}) &= \mathcal{R}(X^{(k)}) + \lambda_k \mathcal{R}'(X^{(k)})S^{(k)} + \frac{\lambda_k^2}{2} \mathcal{R}''(X^{(k)})(S^{(k)}, S^{(k)}) \\ &= (1 - \lambda_k) \mathcal{R}(X^{(k)}) + \lambda_k L^{(k+1)} - \lambda_k^2 S^{(k)} B B^T S^{(k)}. \end{aligned} \quad (3.9)$$

Therefore, if $\eta_k \leq \bar{\eta} < 1$ and $\alpha \in (0, 1 - \bar{\eta})$, then (3.7) and (3.9) imply

$$\begin{aligned} &\left\| \mathcal{R}(X^{(k)} + \lambda S^{(k)}) \right\|_F \\ &\leq (1 - \lambda) \left\| \mathcal{R}(X^{(k)}) \right\|_F + \lambda \left\| L^{(k+1)} \right\|_F + \lambda^2 \left\| S^{(k)} B B^T S^{(k)} \right\|_F \\ &\leq (1 - \lambda + \lambda \bar{\eta}) \left\| \mathcal{R}(X^{(k)}) \right\|_F + \lambda^2 \frac{\left\| S^{(k)} B B^T S^{(k)} \right\|_F}{\left\| \mathcal{R}(X^{(k)}) \right\|_F} \left\| \mathcal{R}(X^{(k)}) \right\|_F \\ &\leq (1 - \alpha \lambda) \left\| \mathcal{R}(X^{(k)}) \right\|_F \end{aligned}$$

for all λ with

$$0 < \lambda \leq (1 - \alpha - \bar{\eta}) \frac{\left\| \mathcal{R}(X^{(k)}) \right\|_F}{\left\| S^{(k)} B B^T S^{(k)} \right\|_F}. \quad (3.10)$$

In particular the sufficient decrease condition (3.5) is satisfied for all λ_k with (3.10).

Algorithm 1 Inexact Kleinman-Newton Method with Line Search

Input: A, B, C , stable initial iterate $X^{(0)}$, $tol_{\text{Newton}} > 0$, $\bar{\eta} \in (0, 1)$ and $\alpha \in (0, 1 - \bar{\eta})$.

Output: Approximate solution of the CARE (1.1).

```
1: for  $k = 0, 1, \dots$  do
2:   if  $\|\mathcal{R}(X^{(k)})\| \leq tol_{\text{Newton}}$  then
3:     Return  $X^{(k)}$  as an approximate solution of the CARE (1.1).
4:   end if
5:   Set  $A^{(k)} = (A - BB^T X^{(k)})$ ,  $G = [C^T \mid X^{(k)} B]$ .
6:   Select  $\eta_k \in (0, \bar{\eta}]$ .
7:   Compute an approximate solution  $\tilde{X}^{(k+1)}$  of the Lyapunov equation such that
      
$$(A^{(k)})^T \tilde{X}^{(k+1)} + \tilde{X}^{(k+1)} A^{(k)} = -GG^T + L^{(k+1)}$$

      and  $\|L^{(k+1)}\|_F \leq \eta_k \|\mathcal{R}(X^{(k)})\|_F$ .
8:   Set  $S^{(k)} = \tilde{X}^{(k+1)} - X^{(k)}$ .
9:   Compute  $\lambda_k > 0$  such that  $\|\mathcal{R}(X^{(k)} + \lambda_k S^{(k)})\|_F \leq (1 - \lambda_k \alpha) \|\mathcal{R}(X^{(k)})\|_F$ .
10:  Set  $X^{(k+1)} = X^{(k)} + \lambda_k S^{(k)}$ .
11: end for
```

In the actual computation of the step size λ_k we use (3.9) which implies that

$$\begin{aligned} f(\lambda) &= \|\mathcal{R}(X^{(k)} + \lambda S^{(k)})\|_F^2 \\ &= (1 - \lambda)^2 \alpha^{(k)} + \lambda^2 \beta^{(k)} + \lambda^4 \delta^{(k)} + 2\lambda(1 - \lambda)\gamma^{(k)} - 2\lambda^2(1 - \lambda)\varepsilon^{(k)} - 2\lambda^3 \zeta^{(k)} \end{aligned} \quad (3.11)$$

is a quartic polynomial with

$$\begin{aligned} \alpha^{(k)} &= \|\mathcal{R}(X^{(k)})\|_F^2, & \delta^{(k)} &= \|S^{(k)} BB^T S^{(k)}\|_F^2, \\ \beta^{(k)} &= \|L^{(k+1)}\|_F^2, & \varepsilon^{(k)} &= \langle \mathcal{R}(X^{(k)}), S^{(k)} BB^T S^{(k)} \rangle, \\ \gamma^{(k)} &= \langle \mathcal{R}(X^{(k)}), L^{(k+1)} \rangle, & \zeta^{(k)} &= \langle L^{(k+1)}, S^{(k)} BB^T S^{(k)} \rangle. \end{aligned} \quad (3.12)$$

The derivative is $f'(\lambda) = \langle \mathcal{R}(X^{(k)} + \lambda S^{(k)}), -\mathcal{R}(X^{(k)}) + L^{(k+1)} - 2\lambda S^{(k)} BB^T S^{(k)} \rangle$. In particular, using the Cauchy–Schwarz inequality and (3.7), we find $f'(0) < 0$, which again confirms that $S^{(k)}$ is a decent direction.

Remark 4 *If the current iterate $X^{(k)}$ is symmetric positive semi-definite, if the solution $\tilde{X}^{(k+1)}$ of (3.8a) is symmetric positive semi-definite, and if $\lambda_k \in (0, 1]$, then $X^{(k+1)} = X^{(k)} + \lambda_k(\tilde{X}^{(k+1)} - X^{(k)})$ is also symmetric positive semi-definite.*

The basic inexact Kleinman-Newton method with line search is summarized in Algorithm 1.

3.2 Line Search

There are many possibilities to compute a step size λ_k that satisfies the sufficient decrease condition (3.5). We review two. In both cases, the representation (3.11) of

the Riccati residual as a quartic polynomial can be used for the efficient implementation of the respective procedure.

3.2.1 Armijo Rule

Given $\beta \in (0, 1)$, the Armijo rule in its simplest form selects $\lambda_k = \beta^\ell$, where ℓ is the smallest integer such that the sufficient decrease condition (3.5) is satisfied. See Kelley [14, Sec. 8.2] for more details. Since the sufficient decrease condition (3.5) is satisfied for all step sizes with (3.10) and ℓ is the smallest integer such that $\lambda_k = \beta^\ell$ satisfies (3.5), the step size λ_k generated by the Armijo rule satisfies

$$\lambda_k > \beta(1 - \alpha - \bar{\eta}) \frac{\|\mathcal{R}(X^{(k)})\|_F}{\|S^{(k)}BB^TS^{(k)}\|_F}.$$

3.2.2 Exact Line Search

Equation (3.11) shows that $\mathcal{R}(X^{(k)} + \lambda S^{(k)})$ is quadratic in λ . Hence, $\min_{\lambda > 0} \|\mathcal{R}(X^{(k)} + \lambda S^{(k)})\|_F^2$ corresponds to the minimization of a the quartic polynomial (3.11). For the Kleinman-Newton method with exact Lyapunov equation solves, $L^{(k+1)} = 0$, the exact line search is analyzed by Benner and Byers [3]. In particular, they show that there is a local minimum $\lambda_k \in (0, 2]$, and that if $A^{(k)}$ is stable and $X^{(k+1)}$ is computed with a step length $\lambda_k \in (0, 2]$, then $A^{(k+1)}$ is also stable. The proof in [3] that $\lambda \mapsto \|\mathcal{R}(X^{(k)} + \lambda S^{(k)})\|_F$ has a local minimum in $(0, 2]$ can be extended under additional conditions on $L^{(k+1)}$. Furthermore, it seems that the proof in [3] that $A^{(k+1)}$ is stable can be extended under some definiteness assumptions on $L^{(k+1)}$, similar to those made by Feitzinger et al. [10]. However, as we will show below, these definiteness assumptions typically do not hold in the low-rank case. Since there is always a step size $\lambda_k \in (0, 1]$ that satisfies the sufficient decrease condition, see (3.10), we minimize $\lambda \mapsto \|\mathcal{R}(X^{(k)} + \lambda S^{(k)})\|_F$ over $[0, 1]$.

3.3 Convergence

Feitzinger et al. [10] extend the convergence results for the Kleiman-Newton method with step size $\lambda_k = 1$ to the inexact case, provided the Lyapunov residual $L^{(k+1)}$ satisfies certain positive semi-definiteness assumptions. We summarize these results next. The first result establishes the well-posedness of the inexact Kleiman-Newton method.

Theorem 5 ([10, Thm. 4.3]) *Let $X^{(k)}$ be a symmetric and positive semi-definite such that $A - BB^T X^{(k)}$ is stable and*

$$L^{(k+1)} \preceq C^T C \tag{3.13}$$

holds. Then

- (i) *the iterate $X^{(k+1)} = \tilde{X}^{(k+1)}$ of the inexact Kleiman-Newton method with step-size $\lambda_k = 1$ is well defined, symmetric and positive semi-definite,*

(ii) and the matrix $A - BB^T X^{(k+1)}$ is stable.

The next theorem presents a convergence result for the inexact Kleiman-Newton method.

Theorem 6 ([10, Thm. 4.4]) *Let Assumption 2 be satisfied and let $X^{(0)}$, symmetric and positive semi-definite, be such that $A - BB^T X^{(0)}$ is stable. Let (3.13) hold for all $k \in \mathbb{N}$, and let $X^{(k)}$ be the iterates of the inexact Kleiman-Newton method with step size $\lambda_k = 1$. If*

$$0 \preceq L^{(k+1)} \preceq (X^{(k+1)} - X^{(k)})BB^T(X^{(k+1)} - X^{(k)}) \quad (3.14)$$

hold for all $k \in \mathbb{N}$, then the iterates of inexact Kleinman-Newton method (3.8) with step size $\lambda_k = 1$ satisfy

- (i) $\lim_{k \rightarrow \infty} X^{(k)} = X^\infty$ and $0 \preceq X^\infty \preceq \dots \preceq X^{(k+1)} \preceq X^{(k)} \preceq \dots \preceq X^{(1)}$,
- (ii) $(A - BB^T X^\infty)$ is stable and X^∞ is maximal solution of $\mathcal{R}(X^\infty) = 0$,
- (iii) $\|X^{(k+1)} - X^\infty\|_F \leq c\|X^{(k)} - X^\infty\|_F^2, k \in \mathbb{N}$.

We will use the low-rank ADI method to approximately solve the Lyapunov equation. This means that in our algorithm $L^{(k+1)}$, $X^{(k)}$, and other matrices are low-rank. We will show that in this case the definiteness assumptions (3.13) and (3.14) typically do not hold.

If the ADI iteration is used to solve the Lyapunov equation (3.8a), then $L^{(k+1)} = WW^T = \mathfrak{F}C^T C \mathfrak{F}^T$, where \mathfrak{F} is a matrix with spectrum inside the unit ball. More precisely, we will see in Section 4 that \mathfrak{F} is a real invertible analytic matrix function depending on A^T and on $\{q\}_{i=1}^{n_{ADI}}$ as defined in (4.6). The next Theorem shows that the assumption (3.13) is not fulfilled, in general, if one uses an ADI method to solve the Lyapunov equation (3.8a).

Theorem 7 *Let $L^{(k+1)} = \mathfrak{F}C^T C \mathfrak{F}^T$ with \mathfrak{F} , where \mathfrak{F} is a matrix with spectrum inside the unit ball. The condition $L^{(k+1)} \preceq C^T C$ implies $\text{range}(\mathfrak{F}C^T) \subset \text{range}(C^T)$.*

Proof. First we show by contradiction that if M, N are symmetric positive semi-definite matrices with $M \succeq N$, then $\ker(M) \subset \ker(N)$ and $\text{range}(N) \subset \text{range}(M)$. In fact, if there exists $v \in \ker(M)$ with $v \notin \ker(N)$, then $v^T M v - v^T N v = -v^T N v < 0$. Hence, $\text{range}(M)^\perp = \ker(M) \subset \ker(N) = \text{range}(N)^\perp$ and, consequently, $\text{range}(N) \subset \text{range}(M)$.

Application of this general result with $M = C^T C$ and $N = L^{(k+1)}$ gives the desired statement. □

Our convergence proof follows that of inexact Newton methods, see, e.g., Kelley [14, Sec. 8.2]. First, we prove $\|\mathcal{R}(X^{(k)})\|_F \rightarrow 0$ and then we use the structure of the Riccati equations to argue convergence of $\{X^{(k)}\}$. In particular, Benner and Byers [3,

Lem. 6] prove that if (A, B) is controllable and $\{\mathcal{R}(X^{(k)})\}$ is bounded, then $\{X^{(k)}\}$ is also bounded. Since controllability of (A, B) implies stabilizability of (A, B) , the assumption of controllability is stronger than Assumption 2. Guo and Laub [11] removed the controllability assumption and showed that if (A, B) is stabilizable, $\{\mathcal{R}(X^{(k)})\}$ is bounded, and the matrices $A^{(k)}$ are stable, then $\{X^{(k)}\}$ is also bounded.

The papers [15] on exact Kleinman-Newton, [3] on Kleinman-Newton with line search and [10] on inexact Kleinman-Newton contain proofs that the matrices $A^{(k)}$ corresponding to the iterates $X^{(k)}$ are stable, provides that $A^{(0)}$ is stable. This implies the unique solution of the Lyapunov equation (1.2) and, therefore, the well-posedness of the respective method. Since the definiteness assumption in [10, Thm. 4.3] typically does not hold in the low-rank case, there is no result yet on the well-posedness of the inexact Kleinman-Newton method and we have to assume existence of $\tilde{X}^{(k+1)}$ such that (3.8a) and (3.7) are satisfied.

Theorem 8 *Let Assumption 2 be satisfied and assume that for all k there exists a symmetric positive semi-definite $\tilde{X}^{(k+1)}$ such that (3.8a) and (3.7) are satisfied.*

(i) *If the step sizes are bounded away from zero, $\lambda_k \geq \lambda_{\min} > 0$ for all k , then $\|\mathcal{R}(X^{(k)})\|_F \rightarrow 0$.*

(ii) *If, in addition to (i), the pair (A, B) is controllable, and $X^{(k)} \succeq 0$ for all $k \geq K$, then $X^{(k)} \rightarrow X^{(*)}$, where $X^{(*)} \succeq 0$ is the unique stabilizing solution of the CARE.*

(iii) *If, in addition to (i), the matrices $A^{(k)}$ are stable for $k \geq K$, and $X^{(k)} \succeq 0$ for all $k \geq K$, then $X^{(k)} \rightarrow X^{(*)}$, where $X^{(*)} \succeq 0$ is the unique stabilizing solution of the CARE.*

Proof. (i) The first part is a standard line search argument. The sufficient decrease condition (3.5) implies

$$\begin{aligned} \|\mathcal{R}(X^{(0)})\|_F &\geq \|\mathcal{R}(X^{(0)})\|_F - \|\mathcal{R}(X^{(K+1)})\|_F \\ &= \sum_{k=0}^K \|\mathcal{R}(X^{(k)})\|_F - \|\mathcal{R}(X^{(k+1)})\|_F \geq \sum_{k=0}^K \lambda_k \alpha \|\mathcal{R}(X^{(k)})\|_F \geq 0. \end{aligned}$$

Taking the limit $K \rightarrow \infty$ and using $\lambda_k \geq \lambda_{\min} > 0$ implies $\|\mathcal{R}(X^{(k)})\|_F \rightarrow 0$.

(ii) If (A, B) is controllable and since $\{\mathcal{R}(X^{(k)})\}$ is bounded, then, by [3, Lem. 6], $\{X^{(k)}\}$ is also bounded. Hence, $\{X^{(k)}\}$ has a converging subsequence. For any converging subsequence $\lim_j X^{(k_j)} \succeq 0$ and $0 = \lim_j \|\mathcal{R}(X^{(k_j)})\|_F = \|\mathcal{R}(\lim_j X^{(k_j)})\|_F$. Since the symmetric positive semi-definite solution to the CARE (1.1) is unique and stabilizing, every converging subsequence of $\{X^{(k)}\}$ has the same limit $X^{(*)}$. Therefore, the entire sequence converges.

(iii) If the matrices $A^{(k)}$ are stable for $k \geq K$ and $\{\mathcal{R}(X^{(k)})\}$ is bounded, [11, Lem. 2.3] guarantees that $\{X^{(k)}\}$ is bounded. Thus, we can proceed as in (ii). \square

Remark 9 *If the step size $\lambda_k \in (0, 1]$, then $X^{(k)} \succeq 0$ for all k , see Remark 4.*

As it is well known for inexact Newton methods, see, e.g., Kelley [14, Sec. 8.2], the specific choice of the forcing parameter η_k in (3.4) determines the rate of convergence. In particular, if $\eta_k \rightarrow 0$ the inexact Kleinman Newton method converges superlinearly (under the assumptions of Theorem 8) and if $\eta_k = O(\|\mathcal{R}(X^{(k_j)})\|_F)$ the convergence is quadratic.

4 ADI Method

To compute the new iterate $X^{(k+1)}$ within the Kleinman-Newton method one has to solve the Lyapunov equation (1.2). In this section we review the basic ingredients of the ADI method [8, 17] and recent [6], [5] algorithmic improvements. To simplify the notation we drop the index k and write this equation in a more general form as

$$F^T X + X F = G G^T \quad (4.1)$$

with $G = [C^T \mid X^{(k)} B] \in \mathbb{R}^{n \times (p+r)}$ and $F = (A - B B^T X^{(k)})^T \in \mathbb{R}^{n \times n}$.

The basic ADI iteration computes a low-rank solution factor Z such that $Z Z^H \approx X$ is the approximated solution of the Lyapunov equation (4.1); see, e.g., [8]. There, one uses a certain number of ADI shifts $\{q_1, \dots, q_\ell\} \in \mathbb{C}^-$ to iteratively build up Z . In each step a new block V_ℓ of the solution is computed via

$$V_1 = (F + q_1 I)^{-1} G, \quad (4.2a)$$

$$V_\ell = (V_{\ell-1} - (q_\ell + \bar{q}_{\ell-1})(F + q_\ell I)^{-1} V_{\ell-1}) \quad \forall \ell \geq 2 \quad (4.2b)$$

such that after ℓ steps Z can be defined as

$$Z = \left[\sqrt{-2 \operatorname{Re}(q_1)} V_1, \dots, \sqrt{-2 \operatorname{Re}(q_\ell)} V_\ell \right] \in \mathbb{C}^{n \times (\ell \cdot (p+r))}. \quad (4.3)$$

The equation (4.2b) can be written as

$$\begin{aligned} V_\ell &= (I - (q_\ell + \bar{q}_{\ell-1})(F + q_\ell I)^{-1}) V_{\ell-1} \\ &= (F - \bar{q}_{\ell-1} I)(F + q_\ell I)^{-1} V_{\ell-1} \end{aligned} \quad \forall \ell \geq 2.$$

Considering the fact that $(F \pm qI)$ and $(F + \hat{q}I)^{-1}$ commute for all $q, \hat{q} \in \mathbb{C} \setminus \Lambda(F)$, V_ℓ can be defined explicitly as

$$V_\ell = \left(\prod_{j=2}^{\ell} (F - \bar{q}_{j-1} I)(F + q_j I)^{-1} \right) (F + q_1 I)^{-1} G.$$

By regrouping these products, Benner et al. introduced a novel low-rank residual

formulation for ADI [5] as follows:

$$V_\ell = (F + q_\ell I)^{-1} \underbrace{\left(\prod_{j=1}^{\ell-1} (F - \bar{q}_j I)(F + q_j I)^{-1} \right)}_{=: W_{\ell-1}} G, \quad (4.4a)$$

$$W_\ell = (F - \bar{q}_\ell I)V_\ell = \prod_{j=1}^{\ell} (F - \bar{q}_j I)(F + q_j I)^{-1} G \quad (4.4b)$$

that can be written implicitly with $W_0 = G$ as

$$\left. \begin{aligned} V_\ell &= (F + q_\ell I)^{-1} W_{\ell-1}, \\ W_\ell &= (F - \bar{q}_\ell I)(F + q_\ell I)^{-1} W_{\ell-1} \\ &= (I - 2 \operatorname{Re}(q_\ell) (F + q_\ell I)^{-1}) W_{\ell-1} \\ &= W_{\ell-1} - 2 \operatorname{Re}(q_\ell) V_\ell, \end{aligned} \right\} \forall \ell \geq 1$$

such that Z can be build up using the definition (4.3) and $X \approx ZZ^H \in \mathbb{R}^{n \times n}$ is the approximate solution of (1.2).

Using this slightly changed formulation, that is mathematically equivalent to the original algorithm in [8, 17], Benner et al. show in [5] that the Lyapunov residual after ADI step ℓ can be written as

$$L_\ell = FZZ^T + ZZ^T F + GG^T = W_\ell W_\ell^T \in \mathbb{R}^{n \times n} \quad (4.5)$$

with $W_\ell \in \mathbb{C}^{n \times (p+r)}$. Furthermore, Benner and colleagues showed that

$$\operatorname{rank}(L) = \operatorname{rank}(W) \leq \operatorname{rank}(G) \ll n.$$

Using the low-rank structure $L_\ell = W_\ell W_\ell^T$ together with the commonly known result that the eigenvalues $\sigma(W_\ell W_\ell^T) \setminus \{0\} = \sigma(W_\ell^T W_\ell) \setminus \{0\}$, see, e.g., [12, Theorem 1.32], leads to an efficient way to compute and accumulate the Lyapunov residual and its norm to control the accuracy of the ADI iteration [5].

Exploiting the fact that the ADI shifts usually occur either as a real number $q_\ell \in \mathbb{R}^-$ or as a pair of complex conjugate numbers $q_\ell \in \mathbb{C}^-$, $q_{\ell+1} = \bar{q}_\ell$, Benner et al. introduced a reformulated low-rank ADI iteration in [4, 6], where the low-rank solution factor $Z \in \mathbb{R}^{n \times (\ell \cdot (p+r))}$ and the low-rank Lyapunov residual factor $W_\ell \in \mathbb{R}^{n \times (p+r)}$ are real valued by construction such that the residual can be written as

$$W_\ell = \mathfrak{F}G \in \mathbb{R}^{n \times (p+r)} \quad (4.6)$$

with $\mathfrak{F}(F, q_1, \dots, q_\ell)$ a real and analytical matrix function depending on F and the ADI shifts q_ℓ . For more details regarding these ADI modifications we refer to [4, 5, 6] and the references therein. Summarizing, these innovations yield the real low-rank ADI method Alg. 2 [6].

Remark 10 For unstable matrices A an initial feedback has to be found to ensure the convergence of the ADI method (see, e.g., [2]) such that condition (3.7) can be fulfilled as it is described in, e.g., [1] and the references therein.

Algorithm 2 real low-rank ADI method [6]

Input: F, G, tol_{ADI} , shifts $q_\ell \in \mathbb{C}^-$.

Output: Z such that $ZZ^T \approx X$ solves Eq. (4.1).

```
1: Set  $\ell = 1$ ,  $Z = []$ ,  $W_0 = G$ .
2: while  $\|W_{\ell-1}^T W_{\ell-1}\|_F > tol_{\text{ADI}}$  do
3:   Solve  $V = (F + q_\ell I)^{-1} W_{\ell-1}$ .
4:   if  $\text{Im}(q_\ell) = 0$  then
5:      $W_\ell = W_{\ell-1} - 2q_\ell V$ 
6:      $\tilde{V} = \sqrt{-2q_\ell} V$ 
7:   else
8:      $\gamma_\ell = 2\sqrt{-\text{Re}(q_\ell)}$ ,  $\delta_\ell = \text{Re}(q_\ell) / \text{Im}(q_\ell)$ 
9:      $W_{\ell+1} = W_{\ell-1} + \gamma_\ell^2 (\text{Re}(V) + \delta_\ell \text{Im}(V))$ 
10:     $\tilde{V} = \left[ \gamma_\ell (\text{Re}(V) + \delta_\ell \text{Im}(V)) \mid \gamma_\ell \sqrt{(\delta_\ell^2 + 1)} \text{Im}(V) \right]$ 
11:     $\ell = \ell + 1$ 
12:   end if
13:    $Z = \left[ Z \mid \tilde{V} \right]$ 
14:    $\ell = \ell + 1$ 
15: end while
```

5 Low-Rank Residual Newton-ADI Method

Using Algorithm 2 as the inner loop to solve the Lyapunov equations in Line 5 of Algorithm 1, we arrive at an algorithm for the Kleinman-Newton method, where the low-rank structure can be used to efficiently compute residuals and the quartic function (3.11) that arises in the line search computation.

As we have seen in the previous section, the Lyapunov residual is

$$L_\ell^{(k+1)} = W_\ell^{(k+1)} (W_\ell^{(k+1)})^T,$$

where ℓ is the iteration counter in the inner ADI iteration and $W_\ell^{(k+1)} \in \mathbb{R}^{n \times (p+r)}$. Since $\|L_\ell^{(k+1)}\|_F^2$ is the sum of the squares of the eigenvalues of $L_\ell^{(k+1)}$ and $\sigma(W_\ell^{(k+1)} (W_\ell^{(k+1)})^T) \setminus \{0\} = \sigma((W_\ell^{(k+1)})^T W_\ell^{(k+1)}) \setminus \{0\}$, the norm $\|L_\ell^{(k+1)}\|_F^2$ can be efficiently computed by solving a small $(p+r) \times (p+r)$ eigenvalue problem.

5.1 Norm of the Difference of Outer Products

Let $W \in \mathbb{R}^{n \times m}$ and $K \in \mathbb{R}^{n \times p}$ with $m+p \ll n$ be generic matrices. We frequently need to compute Frobenius or 2-norms of the difference $WW^T - KK^T$. This can be done efficiently using the indefinite low-rank factorization $WW^T - KK^T = UDU^T \in \mathbb{R}^{n \times n}$, where

$$U = \begin{bmatrix} W & K \end{bmatrix} \quad \text{and} \quad D = \begin{bmatrix} I_m & 0 \\ 0 & -I_p \end{bmatrix}.$$

For the spectrum we have $\sigma(UDU^T) \setminus \{0\} = \sigma(U^TUD) \setminus \{0\}$ (see, e.g., [12, Theorem 1.32]). Since U^TUD is a small $(m+p) \times (m+p)$ matrix, its spectrum can be computed efficiently and we can use

$$\begin{aligned} \|WW^T - KK^T\|_2 &= \max\{|\lambda| : \lambda \in \sigma(WW^T - KK^T)\} = \max\{|\lambda| : \lambda \in \sigma(U^TUD)\}, \\ \|WW^T - KK^T\|_F^2 &= \sum_{\lambda_i \in \sigma(WW^T - KK^T)} \lambda_i^2 = \sum_{\lambda_i \in \sigma(U^TUD)} \lambda_i^2. \end{aligned}$$

Notice that since U^TUD is not symmetric, $\max\{|\lambda| : \lambda \in \sigma(U^TUD)\} \neq \|U^TUD\|_2$ and $\sum_{\lambda_i \in \sigma(U^TUD)} \lambda_i^2 \neq \|U^TUD\|_F^2$.

5.2 Low-Rank Riccati Residual and Feedback Accumulation

Recall that $\tilde{X}^{(k+1)} = X^{(k)} + S^{(k)}$. Consider

$$S^{(k)}B = \tilde{X}^{(k+1)}B - X^{(k)}B =: \tilde{K}^{(k+1)} - K^{(k)} =: \Delta\tilde{K}^{(k+1)} \in \mathbb{R}^{n \times r}, \quad (5.1)$$

which defines the change of the feedback K corresponding to the trial solution $\tilde{X}^{(k+1)}$ of (3.8a).

The key ingredient to use the line search idea efficiently for large-scale problems are the low-rank formulations of the Lyapunov and Riccati residuals. Recall from (4.5) that

$$L^{(k+1)} = W^{(k+1)}(W^{(k+1)})^T \quad (5.2a)$$

and assume that

$$\mathcal{R}(X^{(k)}) = W^{(k)}(W^{(k)})^T - \Delta K^{(k)}(\Delta K^{(k)})^T = U^{(k)}D(U^{(k)})^T \quad (5.2b)$$

with

$$D = \begin{bmatrix} I_{r+p} & 0 \\ 0 & -I_r \end{bmatrix} \text{ and } U^{(k)} = [W^{(k)} \mid \Delta K^{(k)}] \in \mathbb{R}^{n \times (2r+p)}. \quad (5.2c)$$

For $k = 0$ and $X^{(0)} = 0$ (5.2) holds with $W^{(0)} = C^T$ and $\Delta K^{(0)} = 0$. We call a factorization of the form (5.2b) an indefinite low-rank factorization (compare Section 5.1).

If one uses (5.2) and the feedback change (5.1), than (3.9) implies

$$\begin{aligned}
\mathcal{R}(X^{(k+1)}) &= \mathcal{R}(X^{(k)} + \lambda_k S^{(k)}) \\
&= (1 - \lambda_k) U^{(k)} D(U^{(k)})^T + \lambda_k W^{(k+1)} (W^{(k+1)})^T - \lambda_k^2 \Delta \tilde{K}^{(k+1)} \left(\Delta \tilde{K}^{(k+1)} \right)^T \\
&= (1 - \lambda_k) \left(W^{(k)} (W^{(k)})^T - \Delta K^{(k)} (\Delta K^{(k)})^T \right) + \lambda_k W^{(k+1)} (W^{(k+1)})^T \\
&\quad - \lambda_k^2 \Delta \tilde{K}^{(k+1)} \left(\Delta \tilde{K}^{(k+1)} \right)^T \\
&= \begin{bmatrix} \left[\sqrt{(1 - \lambda_k)} W^{(k)} \mid \sqrt{\lambda_k} W^{(k+1)} \right] \left[\sqrt{(1 - \lambda_k)} \Delta K^{(k)} \mid \lambda_k \Delta \tilde{K}^{(k+1)} \right] \\ \times \begin{bmatrix} I_{(s+1)(p+r)} & 0 \\ 0 & -I_{(s+1)r} \end{bmatrix} \\ \times \left[\left[\sqrt{(1 - \lambda_k)} W^{(k)} \mid \sqrt{\lambda_k} W^{(k+1)} \right] \left[\sqrt{(1 - \lambda_k)} \Delta K^{(k)} \mid \lambda_k \Delta \tilde{K}^{(k+1)} \right] \right]^T \end{bmatrix}^T, \quad (5.3)
\end{aligned}$$

where $s \in \{0, 1, \dots\}$ is the number of times immediately before the k st iteration the step size was less than one, $\lambda_{k-s-1} = 1, \lambda_{k-s} < 1, \lambda_k < 1$. See below for more details.

If $\lambda_k = 1$, then $X^{(k+1)} = \tilde{X}^{(k+1)}$, $\Delta K^{(k+1)} = \Delta \tilde{K}^{(k+1)}$ and (5.3) simplifies to

$$\begin{aligned}
\mathcal{R}(X^{(k+1)}) &= \mathcal{R}(\tilde{X}^{(k+1)}) \\
&= W^{(k+1)} (W^{(k+1)})^T - \Delta K^{(k+1)} (\Delta K^{(k+1)})^T =: U^{(k+1)} D(U^{(k+1)})^T \quad (5.4)
\end{aligned}$$

with $U^{(k+1)} = [W^{(k+1)} \mid \Delta K^{(k+1)}]$, which is of the form (5.2b). If $\lambda_k \in (0, 1)$, we can redefine

$$\begin{aligned}
W^{(k+1)} &\leftarrow \left[\sqrt{(1 - \lambda_k)} W^{(k)} \mid \sqrt{\lambda_k} W^{(k+1)} \right] \in \mathbb{R}^{n \times (s+1)(p+r)}, \\
\Delta K^{(k+1)} &\leftarrow \left[\sqrt{(1 - \lambda_k)} \Delta K^{(k)} \mid \lambda_k \Delta \tilde{K}^{(k+1)} \right] \in \mathbb{R}^{n \times (s+1)r}, \\
D &\leftarrow \begin{bmatrix} I_{(s+1)(p+r)} & 0 \\ 0 & -I_{(s+1)r} \end{bmatrix}.
\end{aligned}$$

After this redefinition, (5.4) holds. Notice that if $\lambda_k < 1$ the sizes of $W^{(k+1)}$ and $\Delta K^{(k+1)}$ grow. Their sizes depend on the number s of times immediately before the k st iteration the step size was less than one, $\lambda_{k-s-1} = 1, \lambda_{k-s} < 1, \lambda_k < 1$.

The representation (5.4) can be used to compute the Riccati residual $\|\mathcal{R}(X^{(k)} + \lambda_k S^{(k)})\|_F$ in dependence of λ_k efficiently (see Section 5.1). It is important to mention that we need to keep $U^{(k)} \in \mathbb{R}^{n \times ((s+1)(2r+p))}$ to perform the line search; it is not sufficient to just keep $\|\mathcal{R}(X^{(k)})\|_F$.

The trial iterate $\tilde{X}^{(k+1)}$ is computed by Algorithm 2 iteratively and, consequently, $\tilde{K}^{(k+1)} = \tilde{X}^{(k+1)} B$ can already be computed during the execution of Algorithm 2. Let

ℓ be the iteration counter in Algorithm 2. We have

$$\begin{aligned}\tilde{K}_\ell^{(k+1)} &= \tilde{X}_\ell^{(k+1)} B = \begin{bmatrix} \tilde{V}_1 & \dots & \tilde{V}_\ell \end{bmatrix} \begin{pmatrix} \begin{bmatrix} \tilde{V}_1^T \\ \vdots \\ \tilde{V}_\ell^T \end{bmatrix} B \end{pmatrix} = \sum_{j=1}^{\ell} \tilde{V}_j (\tilde{V}_j^T B) \\ &= \tilde{K}_{\ell-1} + \tilde{V}_\ell (\tilde{V}_\ell^T B) \quad (\tilde{K}_0 = 0).\end{aligned}$$

If we define $\Delta \tilde{K}_0^{(k+1)} = -K^{(k)}$, then

$$\Delta \tilde{K}_\ell^{(k+1)} = \tilde{K}_\ell^{(k+1)} - K^{(k)} = \tilde{K}_{\ell-1}^{(k+1)} + \tilde{V}_\ell (\tilde{V}_\ell^T B) - K^{(k)} = \Delta \tilde{K}_{\ell-1}^{(k+1)} + \tilde{V}_\ell (\tilde{V}_\ell^T B).$$

Thus, the feedback change can be assembled efficiently during the ADI. The low-rank Riccati residual factor for the $k+1$ th Riccati step after ℓ ADI steps can be written as $U_\ell^{(k+1)} = [W_\ell \mid \Delta \tilde{K}_\ell^{(k+1)}] \in \mathbb{R}^{n \times (2r+p)}$. The Riccati residual norm $\|\mathcal{R}(X_\ell^{(k+1)})\|_F$ can be computed easily during the ADI iteration by computing the eigenvalues of the small matrix $(U_\ell^{(k+1)})^T U_\ell^{(k+1)} D$, see Section 5.1.

5.3 Low-Rank Line Search Implementation

To compute the step size as discussed in Section 3.2 to large-scale problems, we need to compute the quartic polynomial (3.11). We can compute the coefficients defined in (3.12) efficiently.

The coefficient $\alpha^{(k)} = \|\mathcal{R}(X^{(k)})\|_F^2$ can be computed using (5.2b) (see Section 5.1). Similarly, $\beta^{(k)} = \|L^{(k+1)}\|_F^2 = \|W^{(k+1)}(W^{(k+1)})^T\|_F^2$ can be computed efficiently as show at the beginning of this section. Instead of using eigenvalues, we can use the property $\text{tr}(MN) = \text{tr}(NM)$ and, for symmetric matrices M , $\text{tr}(M^2) = \sum_{i,j} (M_{ij})^2$, and compute

$$\begin{aligned}\beta^{(k)} &= \|W^{(k+1)}(W^{(k+1)})^T\|_F^2 = \text{tr} \left(W^{(k+1)}(W^{(k+1)})^T W^{(k+1)}(W^{(k+1)})^T \right) \\ &= \text{tr} \left((W^{(k+1)})^T W^{(k+1)} (W^{(k+1)})^T W^{(k+1)} \right) = \|(W^{(k+1)})^T W^{(k+1)}\|_F^2.\end{aligned}$$

Similarly, with $\Delta \tilde{K}^{(k+1)} = S^{(k)} B \in \mathbb{R}^{n \times r}$,

$$\delta^{(k)} = \|\Delta \tilde{K}^{(k+1)} (\Delta \tilde{K}^{(k+1)})^T\|_F^2 = \|(\Delta \tilde{K}^{(k+1)})^T \Delta \tilde{K}^{(k+1)}\|_F^2.$$

Application of trace identities give

$$\begin{aligned}
\gamma^{(k)} &= \langle \mathcal{R}(X^{(k)}), L^{(k+1)} \rangle = \text{tr} \left(U^{(k)} D(U^{(k)})^T W^{(k+1)} (W^{(k+1)})^T \right) \\
&= \text{tr} \left((U^{(k)})^T W^{(k+1)} (W^{(k+1)})^T U^{(k)} D \right) \\
&= \text{tr} \left(\begin{bmatrix} (W^{(k)})^T W^{(k+1)} \\ (\Delta K^{(k)})^T W^{(k+1)} \end{bmatrix} \left[(W^{(k+1)})^T W^{(k)} \mid - (W^{(k+1)})^T \Delta K^{(k)} \right] \right) \\
&= \sum_{i,j} \begin{bmatrix} (W^{(k)})^T W^{(k+1)} \\ (\Delta K^{(k)})^T W^{(k+1)} \end{bmatrix}_{ij} \begin{bmatrix} (W^{(k)})^T (W^{(k+1)}) \\ -(\Delta K^{(k)})^T W^{(k+1)} \end{bmatrix}_{ij} \\
&= \sum_{i,j} \left(((W^{(k)})^T W^{(k+1)})_{ij} \right)^2 - \sum_{i,j} \left((\Delta K^{(k)})^T W^{(k+1)} \right)_{ij}^2
\end{aligned}$$

and, analogously,

$$\begin{aligned}
\varepsilon^{(k)} &= \langle \mathcal{R}(X^{(k)}), S^{(k)} B B^T S^{(k)} \rangle = \text{tr} \left(U^{(k)} D(U^{(k)})^T \Delta \tilde{K}^{(k+1)} (\Delta \tilde{K}^{(k+1)})^T \right) \\
&= \sum_{i,j} \left(((W^{(k)})^T \Delta \tilde{K}^{(k+1)})_{ij} \right)^2 - \sum_{i,j} \left(((\Delta K^{(k)})^T \Delta \tilde{K}^{(k+1)})_{ij} \right)^2.
\end{aligned}$$

Finally,

$$\begin{aligned}
\zeta^{(k)} &= \langle L^{(k+1)}, S^{(k)} B B^T S^{(k)} \rangle = \text{tr} \left(W^{(k+1)} (W^{(k+1)})^T \Delta \tilde{K}^{(k+1)} (\Delta \tilde{K}^{(k+1)})^T \right) \\
&= \sum_{i,j} \left(((W^{(k+1)})^T \Delta \tilde{K}^{(k+1)})_{ij} \right)^2.
\end{aligned}$$

After choosing λ_k appropriately, the next iterate $X^{(k+1)}$ (3.8b) and the feedback $K^{(k+1)}$ can be computed. Using a low-rank ADI method (see Section 4), the low-rank approximations of the previous iterate $X^{(k)} = Z^{(k)} (Z^{(k)})^T$ and $\tilde{X}^{(k+1)} = \tilde{Z}^{(k+1)} (\tilde{Z}^{(k+1)})^T$ as low-rank approximation of the solution of (3.8a) are used.

$$\begin{aligned}
X^{(k+1)} &= X^{(k)} + \lambda_k S^{(k)} = (1 - \lambda_k) X^{(k)} + \lambda_k \tilde{X}^{(k+1)} \\
&= (1 - \lambda_k) Z^{(k)} (Z^{(k)})^T + \lambda_k \tilde{Z}^{(k+1)} (\tilde{Z}^{(k+1)})^T \\
&= \left[\sqrt{1 - \lambda_k} Z^{(k)} \mid \sqrt{\lambda_k} \tilde{Z}^{(k+1)} \right] \left[\sqrt{1 - \lambda_k} Z^{(k)} \mid \sqrt{\lambda_k} \tilde{Z}^{(k+1)} \right]^T. \quad (5.5)
\end{aligned}$$

$$\begin{aligned}
K^{(k+1)} &= X^{(k+1)} B = (1 - \lambda_k) X^{(k)} B + \lambda_k \tilde{X}^{(k+1)} B \\
&= (1 - \lambda_k) K^{(k)} + \lambda_k \tilde{K}^{(k+1)}. \quad (5.6)
\end{aligned}$$

Notice that the size of $Z^{(k)}$ and $\tilde{Z}^{(k+1)}$ depends on the number of ADI steps that are needed to solve (3.8a). Although (5.5) might be very large, it is important to mention that it only needs to be computed at the end of the Newton iteration, because

the previous iterate $X^{(k)}$ enters the right-hand side of (3.8a) only as product with the input matrix B from the right. This means one only needs the inexpensively accumulated feedback $K^{(k+1)} = X^{(k+1)}B$ in Eq. (5.6) to proceed with the Newton iteration. Furthermore, typically, a line search will only be necessary in the first one or two Newton steps, so that (5.5) might never be used after the first few iterations and instead simply $X^{(k+1)} = \tilde{Z}^{(k+1)}(\tilde{Z}^{(k+1)})^T$.

5.4 Complete Implementation

We conclude this section with a summary of the resulting algorithm and some comments on the line search and the convergence of the inexact Kleinman-Newton method with line search.

Remark 11 1. The step size $\lambda_k = 1$ is accepted if after reaching the condition (3.7) at ADI step ℓ it holds that $\|\mathcal{R}(X_\ell^{(k+1)})\| \leq (1 - \alpha)\|\mathcal{R}(X^{(k)})\|$.

One could perform line search in every Newton step. We have observed that in some examples this can slightly reduce the number of Newton steps, but the cost of the line search computation increases, since the Riccati low-rank residual factor grows significantly as shown in (5.3), and computational cost associated with this growth can destroy the gains due to saving of a Newton iteration.

2. We also perform a line search when

a) Before reaching the condition (3.7) the actual step $\ell \geq 2$ yields

$$\|L_\ell\|_F > \|L_1\|_F,$$

i.e., the norm of the Lyapunov residual exceeds the norm of the initial Lyapunov residual.

b) The number of ADI steps ℓ exceeds the maximal number of allowed ADI steps without reaching the condition (3.7).

If conditions 2a or 2b are observed, it indicates that the ADI methods does not converge, e.g., because the matrix $A^{(k)}$ is not stable. Although condition (3.7) is violated, we perform a line search, since the cost of its execution is small, and accept $X^{(k+1)} = X^{(k)} + \lambda_k S^{(k)}$ if condition (3.5) is fulfilled.

3. If the line search method determines a λ_k that is too small (currently we use $\lambda_k \alpha \leq 10^{-16}$) we switch to an ‘exact’ Kleinman-Newton method, i.e., we use Algorithm 1 with ADI Algorithm 2 with tolerance $\text{tol}_{\text{ADI}} = 10^{-1} \text{tol}_{\text{Newt}}$ as the inner solver. Since we cannot guarantee stability of $A^{(k)}$ it is not guaranteed that the ADI Algorithm 2 converges. If we observe that ADI Algorithm 2 does not converge, we restart the entire process using the ‘exact’ Kleinman-Newton method as described above. During the ‘exact’ Kleinman-Newton scheme, the algorithm switches back to the inexact scheme as soon as the Riccati residual shows the expected convergence behavior.

Algorithm 3 inexact Kleinman-Newton-ADI method with line search

Input: A, B, C , initial feedback $K^{(0)}$, tol_{Newt} , $\bar{\eta} > 0$.

Output: $K^{(k+1)}$ (optional: $Z^{(k+1)}$ such that $Z^{(k+1)}(Z^{(k+1)})^T$ is a stabilizing approximate solution of the CARE (1.1)).

```
1: Set  $k = 0$ ,  $\text{res}_{\text{Newt}}^{(0)} = \|C^T C + K^{(0)}(K^{(0)})^T\|$ .
2: while  $(\text{res}_{\text{Newt}}^{(k)} > \text{tol}_{\text{Newt}} \cdot \text{res}_{\text{Newt}}^{(0)})$  do
3:   Set  $A^{(k)} = (A^T - K^{(k)}B^T)$ ,  $G = [C^T \mid K^{(k)}]$ .
4:   Compute ADI shifts  $\{q_\ell\}_{\ell=1}^{n_{\text{max,ADI}}} \in \mathbb{C}^-$  and choose  $\eta_k \in (0, \bar{\eta}]$ .
5:   Set  $\ell = 1$ ,  $W_0 = G$ ,  $\Delta K_0 = -K^{(k)}$  (optional  $Z = []$ ).
6:   while  $(\|W_\ell W_\ell^T\|_F > \eta_k \text{res}_{\text{Newt}}^{(k)})$  do
7:      $V = (A^{(k)} + q_\ell I)^{-1} W_{\ell-1}$ 
8:     if  $\text{Im}(q_\ell) = 0$  then
9:        $W_\ell = W_{\ell-1} - 2q_\ell V$ 
10:       $\tilde{V} = \sqrt{-2q_\ell} V$ 
11:       $\Delta K_\ell = \Delta K_{\ell-1} + \tilde{V}(\tilde{V}^T B)$ 
12:     else
13:        $\gamma = 2\sqrt{-\text{Re}(q_\ell)}$ ,  $\delta = \text{Re}(q_\ell) / \text{Im}(q_\ell)$ 
14:        $W_{\ell+1} = W_{\ell-1} + \gamma^2(\text{Re}(V) + \delta \text{Im}(V))$ 
15:        $\tilde{V} = [\gamma(\text{Re}(V) + \delta \text{Im}(V)) \mid \gamma\sqrt{(\delta^2 + 1)}\text{Im}(V)]$ 
16:        $\ell = \ell + 1$ 
17:        $\Delta K_\ell = \Delta K_{\ell-2} + \tilde{V}(\tilde{V}^T B)$ 
18:     end if
19:     (optional  $Z = [Z \mid \tilde{V}]$ )
20:      $U_\ell = [W_\ell \mid \Delta K_\ell]$ 
21:      $\ell = \ell + 1$ 
22:   end while
23:   if  $\|U_\ell D U_\ell^T\|_F > (1 - \alpha)\text{res}_{\text{Newt}}^{(k)}$  then
24:     Choose  $\lambda_k \in (0, 1)$  using Armijo or exact line search.
25:      $\Delta K_{\ell-1} = \lambda_k \Delta K_{\ell-1}$ 
26:      $W^{(k+1)} = [\sqrt{1 - \lambda_k} W^{(k)} \mid \sqrt{\lambda_k} W^{(k+1)}]$ 
27:      $\Delta K^{(k+1)} = [\sqrt{1 - \lambda_k} \Delta K^{(k)} \mid \Delta K_{\ell-1}]$ 
28:      $U^{(k+1)} = [W^{(k+1)} \mid \Delta K^{(k+1)}]$ 
29:     (optional  $Z^{(k+1)} = [\sqrt{1 - \lambda_k} Z^{(k)} \mid \sqrt{\lambda_k} Z]$ )
30:   else
31:      $W^{(k+1)} = W_\ell$ ,  $\Delta K^{(k+1)} = \Delta K_{\ell-1}$ 
32:      $U^{(k+1)} = U_\ell$ 
33:     (optional  $Z^{(k+1)} = Z$ )
34:   end if
35:    $K^{(k+1)} = K^{(k)} + \Delta K_{\ell-1}$ 
36:    $\text{res}_{\text{Newt}}^{(k+1)} = \|U^{(k+1)} D (U^{(k+1)})^T\|_F$ 
37:    $k = k + 1$ 
38: end while
```

We note that in the numerical example studies in the next section, the ADI Algorithm 2 always reached the required tolerance, i.e., condition (3.7) was always achieved, and the line search was always successful.

The inexact Kleinman-Newton method with line search and a real low-rank ADI method as inner solve is summarized in Algorithm 3. The residual $\mathcal{R}(\tilde{X}^{(k+1)}) = U_\ell^{(k+1)} D(U_\ell^{(k+1)})^T$ is accumulated during the ADI iteration. In practice the factor U of the indefinite low-rank decomposition of the Riccati residual in lines 20, 28, and 32 is never assembled explicitly since norm computation and line search directly use W and ΔK .

6 Numerical experiments

We demonstrate the performance of our method on a numerical examples that was also used, e.g., by Feitzinger et al. [10]. Consider the infinite dimensional optimal control problem

$$\begin{aligned} & \text{Minimize } \frac{1}{2} \int_0^\infty \left(\tilde{\gamma} \int_\Omega \tilde{x}(\xi, t) d\xi \right)^2 + u^2(t) dt, \\ & \text{subject to } \frac{\partial \tilde{x}}{\partial t}(\xi, t) = \Delta \tilde{x}(\xi, t) + 20 \frac{\partial z}{\partial \xi_2}(\xi, t) + 100 \tilde{x}(\xi, t) + f(\xi)u(t), \quad \xi \in \Omega, t > 0, \\ & \quad \tilde{x}(\xi, t) = 0, \quad \xi \in \partial\Omega, t > 0, \end{aligned}$$

with $\Omega = (0, 1) \times (0, 1)$, $\tilde{\gamma} > 0$, and

$$f(\xi) := \begin{cases} 100 & 0.1 < \xi_1 < 0.3, \quad 0.4 < \xi_2 < 0.6 \\ 0 & \text{else.} \end{cases}$$

We use a finite difference method with upwinding on a uniform grid with mesh size $h = 1/(n+1)$. The spatial discretization of the objective function is

$$\tilde{\gamma} \int_\Omega \tilde{x}(\xi, t) d\xi \approx \tilde{\gamma} h^2 \sum_{i,j=1}^n x_{ij}(t) = \tilde{\gamma} h^2 \mathbf{e}_n^T x(t) = \gamma C x(t),$$

where $x_{ij}(t) \approx \tilde{x}(ih, jh, t)$, $i, j = 1, \dots, n$, $\gamma = 10\tilde{\gamma}h^2$, and $C = [0.1, \dots, 0.1]$. The rescaling of $\tilde{\gamma}$ to γ is used to arrive at the same matrix C used in [10]. After this spatial discretization we obtain the linear quadratic control problem

$$\text{Minimize } \frac{1}{2} \int_0^\infty y(t)^T y(t) + u^2(t) dt, \quad (6.1a)$$

$$\text{subject to } \dot{x}(t) = Ax(t) + Bu(t), \quad t > 0, \quad (6.1b)$$

$$y(t) = \gamma C x(t), \quad t > 0, \quad (6.1c)$$

with $A \in \mathbb{R}^{n^2 \times n^2}$, $B, C^T \in \mathbb{R}^{n^2 \times 1}$, $x(t) \in \mathbb{R}^{n^2}$, $u(t) \in \mathbb{R}$, and $y(t) \in \mathbb{R}$.

Table 1: Performance of the various Kleinman-Newton-ADI methods. The inexact Kleinman-Newton-ADI with forcing parameter $\eta_k = 0.9\|\mathcal{R}(X^{(k-1)})\|_F$ (quadratic) outperforms the other algorithm choices, especially for $\gamma > 1$. The line search can lead to significant savings, especially for $\gamma > 1$, although the step size is less than one only in at most the first two iterations.

(a) Comparison for $\gamma = 10^0$				
Method		# Newt.	# ADI	# LS
exact	$\text{tol}_{\text{ADI}} = 10^{-13}$	6	556	no LS
		5	524	1
inexact	superlinear	5	58	no LS
		5	58	0
	quadratic	6	54	no LS
		6	54	0

(b) Comparison for $\gamma = 10^2$				
Method		# Newt.	# ADI	# LS
exact	$\text{tol}_{\text{ADI}} = 10^{-13}$	15	825	no LS
		10	677	2
inexact	superlinear	13	230	no LS
		9	137	1
	quadratic	8	83	no LS
		5	66	1

(c) Comparison for $\gamma = 10^4$				
Method		# Newt.	# ADI	# LS
exact	$\text{tol}_{\text{ADI}} = 10^{-13}$	28	987	no LS
		16	720	2
inexact	superlinear	25	505	no LS
		15	287	2
	quadratic	24	286	no LS
		9	140	1

The matrix A is stable and, therefore, for this LQR problem (6.1) Assumption 2 is satisfied. It is a basic result, see, e.g., [18], that $u_*(t) = -B^T X x_*(t)$ minimizes the cost functional (6.1a) with X as solution of the CARE (1.1).

For $\gamma = 1$ the LQR problem (6.1) is equivalent to the example in [10]. A $\gamma \gg 1$ increases the effect that $\|\mathcal{R}(X^{(1)})\|_F \gg \|\mathcal{R}(X^{(0)})\|_F$. We use the mesh size $h = 1/(n+1)$ with $n = 23$, as in [10], and we consider $\gamma = 1, 10^2, 10^4$. The ADI shifts are computed following the V -shifts idea in [7].

We apply the Kleinman-Newton-ADI method either ‘exactly’ or inexactly. In the latter case we either use the forcing parameter η_k in (3.4) given by $\eta_k = 1/(k^3 + 1)$ or by $\eta_k = 0.9\|\mathcal{R}(X^{(k)})\|_F$. The first choice leads to superlinear convergence, while the second results in quadratic convergence (under the assumptions of Theorem 8).

In all cases the Kleinman-Newton-ADI is stopped when the normalized residual $\|\mathcal{R}(X^{(k)})\|/\|C^T C\|$ drops below $\text{tol}_{\text{Newt}} = 10^{-12}$. In the ‘exact’ Kleinman-Newton-ADI method, the ADI tolerance is set to $\text{tol}_{\text{ADI}} = \text{tol}_{\text{Newt}}/10$. We apply all methods without line search (‘no LS’), i.e., set $\lambda_k = 1$ in all iterations, and with line search. If the sufficient decrease condition (3.5) is not satisfied for $\lambda_k = 1$, then we compute a steps-size by minimizing (3.11) over $[0, 1]$ using the MATLAB[®] routine `fminbnd`.

The inexact Kleinman-Newton-ADI with superlinear convergence, $\eta_k = 1/(k^3 + 1)$, and no line search is comparable to the method proposed in [10], although the ADI implementation and shift selections are not the same.

The performances of the various Kleinman-Newton-ADI methods are summarized in Table 1. In Table 1, # Newt. is the total number of (inexact) Newton steps executed before the stopping criteria $\|\mathcal{R}(X^{(k)})\|/\|C^T C\| < \text{tol}_{\text{Newt}} = 10^{-12}$ is satisfied, # ADI is the total number of ADI iterations executed, and # LS is the total number of times the step size λ_k was chosen to be less than one. The entry ‘no LS’ indicated that $\lambda_k = 1$ was set in all iterations.

In all variations of the Kleinman-Newton-ADI method, the execution times are essentially proportional to the total number of ADI steps performed. Due to the low-rank structure, the execution times for other algorithm components, such as line search are negligible compared the execution of one ADI iteration.

Table 1 shows that the inexact Kleinman-Newton method significantly outperforms the exact version. In our test, the forcing parameter $\eta_k = 0.9\|\mathcal{R}(X^{(k)})\|_F$ (quadratic) led to significantly better performance than the forcing parameter $\eta_k = 1/(k^3 + 1)$ (superlinear). The differences in performance were the bigger the larger the problem parameter γ (the weight on the output y). In addition, Table 1 shows that the addition of a line search can lead to substantial performance gains. Although the steps size λ_k was less than one in at most two iterations, the savings in outer Newton and inner ADI iterations can be significant. Especially when $\gamma = 10^4$, the line search can lead to computational saving of up to 50%. The line search is active, i.e., $\lambda_k \neq 1$, in at most the first two iterations.

7 Conclusion

We have presented an efficient implementation of the inexact Kleinman-Newton method with a low-rank ADI subproblem solver. In our numerical example, the modifications lead to substantial improvements. On the theoretical side, we presented a convergence proof which is based on convergence proofs for general inexact Newton methods. Because of the low-rank case and lack of positive semi-definiteness conditions, like the one in Theorem 5 [10, Thm. 4.3], it is not possible to ensure that all iterates are stabilizing if the initial iterate is stabilizing. This was not an issue in our numerical example. We have begun numerical experiments with the computation of feedback controls for incompressible Navier-Stokes flows, similar to [1], where stability of iterates can be an issue. A detailed report of these tests, and comparisons with other large-scale Riccati solvers, like [9, 19], is part of future research.

References

- [1] E. BÄNSCH, P. BENNER, J. SAAK, AND H. K. WEICHELT, *Riccati-based boundary feedback stabilization of incompressible Navier-Stokes flows*, SIAM J. Sci. Comput., 37 (2015), pp. A832–A858.
- [2] P. BENNER, *Solving large-scale control problems*, IEEE Contr. Syst. Mag., 14 (2004), pp. 44–59.
- [3] P. BENNER AND R. BYERS, *An exact line search method for solving generalized continuous-time algebraic Riccati equations*, IEEE Trans. Automat. Control, 43 (1998), pp. 101–107.
- [4] P. BENNER, P. KÜRSCHNER, AND J. SAAK, *Efficient handling of complex shift parameters in the low-rank Cholesky factor ADI method*, Numer. Algorithms, 62 (2013), pp. 225–251.
- [5] ———, *An improved numerical method for balanced truncation for symmetric second order systems*, Math. Comp. Model. Dyn., 19 (2013), pp. 593–615.
- [6] ———, *A reformulated low-rank ADI iteration with explicit residual factors*, Proc. Appl. Math. Mech., 13 (2013), pp. 585–586.
- [7] ———, *Self-generating and efficient shift parameters in ADI methods for large Lyapunov and Sylvester equations*, Electron. Trans. Numer. Anal., 43 (2014), pp. 142–162.
- [8] P. BENNER, J.-R. LI, AND T. PENZL, *Numerical solution of large Lyapunov equations, Riccati equations, and linear-quadratic control problems*, Numer. Lin. Alg. Appl., 15 (2008), pp. 755–777.
- [9] P. BENNER AND J. SAAK, *A Galerkin-Newton-ADI Method for Solving Large-Scale Algebraic Riccati Equations*, Preprint SPP1253-090, DFG-SPP1253, 2010. Available from <http://www.am.uni-erlangen.de/home/spp1253/wiki/images/2/28/Preprint-SPP1253-090.pdf>.
- [10] F. FEITZINGER, T. HYLLA, AND E. W. SACHS, *Inexact Kleinman-Newton method for Riccati equations*, SIAM J. Matrix Anal. Appl., 31 (2009), pp. 272–288.
- [11] C.-H. GUO AND A. J. LAUB, *On a Newton-like method for solving algebraic Riccati equations*, SIAM J. Matrix Anal. Appl., 21 (1999), pp. 694–698 (electronic).
- [12] N. J. HIGHAM, *Functions of Matrices: Theory and Computation*, Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 2008.
- [13] L. HOGBEN, ed., *Handbook of Linear Algebra*, Chapman & Hall/CRC, Boca Raton, London, New York, 2nd edition ed., 2014.
- [14] C. T. KELLEY, *Iterative Methods for Linear and Nonlinear Equations*, SIAM, Philadelphia, 1995.

- [15] D. L. KLEINMAN, *On an iterative technique for Riccati equation computations*, IEEE Trans. Automat. Control, 13 (1968), pp. 114–115.
- [16] P. LANCASTER AND L. RODMAN, *The algebraic Riccati equation*, Oxford University Press, Oxford, 1995.
- [17] J.-R. LI AND J. WHITE, *Low rank solution of Lyapunov equations*, SIAM J. Matrix Anal. Appl., 24 (2002), pp. 260–280.
- [18] A. LOCATELLI, *Optimal control: An introduction*, Birkhäuser Verlag, Basel, Boston, Berlin, 2001.
- [19] V. SIMONCINI, D. B. SZYLD, AND M. MONSALVE, *On two numerical methods for the solution of large-scale algebraic Riccati equations*, IMA J. Numer. Anal., 34 (2014), pp. 904–920.

