

Research Article

Nicholas Assimakis and Maria Adam*

Fast iterative solutions of Riccati and Lyapunov equations

<https://doi.org/10.1515/math-2022-0546>

received June 24, 2022; accepted December 8, 2022

Abstract: In this article, new iterative algorithms for solving the discrete Riccati and Lyapunov equations are derived in the case where the transition matrix is diagonalizable with real eigenvalues. It is shown that the proposed iterative algorithms are faster than the classical ones, even for a small number of iterations.

Keywords: Riccati equation, Lyapunov equation, Kalman filter, steady state

MSC: 15A24, 15B57, 93C55

1 Introduction

The discrete time Riccati equation arises by implementing the discrete time Kalman filter, which is the most famous estimation algorithm associated with time invariant models of the form [1]:

$$x(k+1) = Fx(k) + w(k), \quad (1)$$

$$z(k) = Hx(k) + v(k), \quad (2)$$

for $k \geq 0$ (discrete time).

In these models, $x(k)$ is the n dimensional state and $z(k)$ is the m dimensional measurement. The model describes the relation between two successive states through the $n \times n$ transition matrix F and the relation between the state and the measurement through the $m \times n$ output matrix H . In addition, $w(k)$ and $v(k)$ are the state and measurement noises, which are zero mean Gaussian processes with known covariances Q and R , respectively. It is worth to note that Q and R are symmetric and nonnegative definite of dimensions $n \times n$ and $m \times m$, respectively. So, the model parameters F , H , Q and R are time invariant (constant) and known.

Kalman filter produces iteratively the state prediction with the associated prediction error covariance as well as the state estimation with the associated estimation error covariance. The equations of the discrete time Kalman filter result in the discrete time *Riccati equation*:

$$P(k+1|k) = Q + FP(k|k-1)F^T - FP(k|k-1)H^T[HP(k|k-1)H^T + R]^{-1}HP(k|k-1)F^T. \quad (3)$$

Riccati equation relates two successive values of the prediction error covariance $P(k|k-1)$, which is symmetric and nonnegative definite of dimensions $n \times n$.

It is well known [1] that if the system is asymptotically stable, i.e., all eigenvalues of the transition matrix F lie inside the unit circle, then for any nonnegative symmetric initial condition $P_0 = P(0|-1)$, there exists a unique steady state or limiting solution P of the Riccati equation:

* **Corresponding author: Maria Adam**, Department of Computer Science and Biomedical Informatics, University of Thessaly, Lamia, Greece, e-mail: madam@dib.uth.gr

Nicholas Assimakis: Department of Digital Industry Technologies, National and Kapodistrian University of Athens, Psachna Evias, Greece, e-mail: nasimakis@uoa.gr

$$P = \lim_{k \rightarrow \infty} P(k | k - 1). \quad (4)$$

This limiting solution satisfies the algebraic Riccati equation (or steady-state Riccati equation):

$$P = Q + FPF^T - FPH^T[HPH^T + R]^{-1}HPF^T. \quad (5)$$

It is worth to note that inverse of the matrix $HP(k/k - 1)H^T + R$ in (3) exists when R is positive definite, which means that no measurement is accurate, or when the initial condition P_0 is positive definite, which seems not to be a problem, since we are able to choose a convenient initial condition.

The nonsingularity of Q and R ensures the nonsingularity of $P(k/k - 1)$, which then it is positive definite. In this (reasonable and general) case, by using the matrix inversion lemma,¹ the Riccati equation (3) can be transformed to the *transformed Riccati equation*:

$$P(k + 1 | k) = Q + F[P^{-1}(k | k - 1) + B]^{-1}F^T, \quad (6)$$

where

$$B = H^TR^{-1}H. \quad (7)$$

The limiting solution P of the transformed Riccati equation (6) satisfies the transformed algebraic Riccati equation:

$$P = Q + F[P^{-1} + B]^{-1}F^T. \quad (8)$$

In the infinite measurement noise case, the discrete time *Lyapunov equation* is derived:

$$P(k + 1 | k) = Q + FP(k | k - 1)F^T, \quad (9)$$

with a limiting solution P , that satisfies the algebraic Lyapunov equation:

$$P = Q + FPF^T. \quad (10)$$

The importance of the Riccati and Lyapunov equations is undoubtable: the Riccati equation plays a very important role in various problems of stochastic filtering, statistics, ladder networks, and dynamic programming [1,2], and has many applications in the process of obtaining optimal control and determining system stability [3,4]; Lyapunov equations play a very important role in the stability theory of discrete systems [1,5].

Significant bibliography exists concerning iterative as well as noniterative solutions of the Riccati and Lyapunov equations [1,2,6–13]. The existence of the unique solution of the Riccati equation and of the Lyapunov equation requires the knowledge of the eigenvalues of the transition matrix. The classical iterative solution of the Riccati equation implements the iterative Riccati equation (3) or the iterative transformed Riccati equation (6) till the steady-state solution is reached. Similarly, the classical iterative solution of the Lyapunov equation implements the iterative Lyapunov equation (7) till the steady-state solution is reached.

The aim of this article is to study the optimal algorithm for solving the discrete Riccati and Lyapunov equations, in the case where the transition matrix is diagonalizable with real eigenvalues. The required condition appears in various application areas, see the eigenvalues of the transition matrix in the radar tracking system [14], in the eye movement prediction model [15], and in the mobile position tracking model [16,17]; the state and parameter estimation has been derived using a transition matrix, which is a diagonal itself [18]. The novelty of this article concerns the exploitation of the knowledge of the eigenvalues and eigenvectors of the transition matrix, which leads to the development of the new algorithms for iterative solving of the Riccati and Lyapunov equations, which are superior to classical algorithms, since the proposed algorithms are in general faster than the classical ones.

¹ Let A and C be $n \times n$ and $m \times m$ invertible matrices and B and D be matrices of size $n \times m$ and $m \times n$, respectively. Then, the matrix inversion lemma is formulated:

$$[A + BCD]^{-1} = A^{-1} - A^{-1}B[C^{-1} + DA^{-1}B]^{-1}DA^{-1}.$$

This article is organized as follows: In Section 2, new iterative solutions of the Riccati and Lyapunov equations are derived by the eigenvalues and eigenvectors of the transition matrix. In Section 3, the proposed algorithms are compared to the classical ones with respect to their computational burdens. Section 4 summarizes the conclusions.

2 New iterative solution algorithms for Riccati equation and Lyapunov equation

Recall that the model parameters F , H , Q and R are time invariant (constant) and known. The Riccati equation and the Lyapunov equation have an unique symmetric nonnegative definite solution P if and only if $|\lambda_i| < 1$ for all $i = 1, \dots, n$, where λ_i , $i = 1 \dots n$ are the eigenvalues of F .

In the following, we consider that the $n \times n$ transition matrix F has real eigenvalues with the algebraic multiplicity of every eigenvalue identified by its geometric; thus, the associated eigenvectors are linearly independent vectors.

Consider the diagonalization formula of the transition matrix F given by

$$F = WLW^{-1}, \quad (11)$$

where L is the $n \times n$ diagonal matrix containing the eigenvalues of matrix F and the $n \times n$ matrix; W consists of columns of the associated linearly independent eigenvectors of F .

It is worth noting that equation (11) is essential for the derivation of the new iterative algorithms for solving the discrete Riccati and Lyapunov equations, since they use the eigenvalues and eigenvectors of the transition matrix instead of the transition matrix itself, reducing the computations due to the similarity of the matrices F and L .

Concerning the *Riccati equation*, by multiplying the left-hand side of equation (3) by W^{-1} and the right-hand side by W^{-T} and using (11) and the property of the diagonal matrix L , $L = L^T$, as well as the equalities $W^T W^{-T} = W^{-T} W^T = I$, where I denotes the identity matrix, we derive

$$\begin{aligned} W^{-1}P(k+1|k)W^{-T} &= W^{-1}QW^{-T} + LW^{-1}P(k|k-1)W^{-T}L - LW^{-1}P(k|k-1)W^{-T}W^TH^T \\ &\quad [HWW^{-1}P(k|k-1)W^{-T}W^TH^T + R]^{-1} \\ &\quad HWW^{-1}P(k|k-1)W^{-T}L. \end{aligned}$$

Setting in the latter equality

$$\tilde{P}(k|k-1) = W^{-1}P(k|k-1)W^{-T}, \quad (12)$$

$$\tilde{Q} = W^{-1}QW^{-T}, \quad (13)$$

$$\tilde{H} = HW \quad (14)$$

arises the modified Riccati equation.

$$\tilde{P}(k+1|k) = \tilde{Q} + L\tilde{P}(k|k-1)L - L\tilde{P}(k|k-1)\tilde{H}^T[\tilde{H}\tilde{P}(k|k-1)\tilde{H}^T + R]^{-1}\tilde{H}\tilde{P}(k|k-1)L. \quad (15)$$

It is important to note that the Riccati equation (3) with parameters F , H , Q and R has been modified to equation (15), which is a Riccati equation with well-defined parameters L , $\tilde{H}$, $\tilde{Q}$ and R .

Then, the limiting solution of (15) is

$$\tilde{P} = \lim_{k \rightarrow \infty} \tilde{P}(k+1|k) = \lim_{k \rightarrow \infty} \tilde{P}(k|k-1) \quad (16)$$

and satisfies the modified algebraic Riccati equation:

$$\tilde{P} = \tilde{Q} + L\tilde{P}L - L\tilde{P}\tilde{H}^T[\tilde{H}\tilde{P}\tilde{H}^T + R]^{-1}\tilde{H}\tilde{P}L. \quad (17)$$

It is clear that the solution of the Riccati equation (3) can be computed by the solution of the modified Riccati Equation (15) due to (16) and (12):

$$P = W\tilde{P}W^T. \quad (18)$$

Concerning the *transformed Riccati equation*, by multiplying the left-hand side of equation (6) by W^{-1} and the right-hand side by W^{-T} , using (11) and the property of the diagonal matrix L , $L = L^T$, as well as the equalities $W^TW^{-T} = W^{-T}W^T = I$, we derive

$$\begin{aligned} W^{-1}P(k+1|k)W^{-T} &= W^{-1}QW^{-T} + LW^{-1}[P^{-1}(k|k-1) + B]^{-1}W^{-T}L^TW^TW^{-T} \\ &= W^{-1}QW^{-T} + L[W^TP^{-1}(k|k-1)W + W^TBW]^{-1}L. \end{aligned}$$

Setting in the latter equality

$$\tilde{P}(k|k-1) = W^{-1}P(k|k-1)W^{-T}, \quad (19)$$

$$\tilde{Q} = W^{-1}QW^{-T},$$

$$\tilde{B} = W^TBW \quad (20)$$

Q arises the modified transformed Riccati equation:

$$\tilde{P}(k+1|k) = \tilde{Q} + L[\tilde{P}^{-1}(k|k-1) + \tilde{B}]^{-1}L. \quad (21)$$

It is important to note that the transformed Riccati equation (6) with parameters F , H , Q and R has been modified to Equation (21), which is a transformed Riccati equation with well-defined parameters L , $\tilde{H}$, $\tilde{Q}$ and R .

Then, using (16)

$$\tilde{P} = \lim_{k \rightarrow \infty} \tilde{P}(k|k-1) = \lim_{k \rightarrow \infty} \tilde{P}(k+1/k)$$

arises the limiting solution of (21), which satisfies the algebraic Riccati equation:

$$\tilde{P} = \tilde{Q} + L[\tilde{P}^{-1} + \tilde{B}]^{-1}L. \quad (22)$$

It is clear that the solution of the transformed Riccati equation (6) can be computed by the solution of the Riccati equation (22) due to (16) and (19):

$$P = W\tilde{P}W^T. \quad (23)$$

Concerning the *Lyapunov equation*, by multiplying the left-hand side of (9) by W^{-1} and the right-hand side by W^{-T} , using (11) and the property of the diagonal matrix L , $L = L^T$, as well as the equalities $W^TW^{-T} = W^{-T}W^T = I$, we derive

$$P(k+1|k)W^{-T} = W^{-1}QW^{-T} + LW^{-1}P(k|k-1)W^{-T}L.$$

Setting in the latter equality

$$\tilde{P}(k|k-1) = W^{-1}P(k|k-1)W^{-T}, \quad (24)$$

$$\tilde{Q} = W^{-1}QW^{-T}$$

arises

$$\tilde{P}(k+1|k) = \tilde{Q} + L\tilde{P}(k|k-1)L. \quad (25)$$

It is important to note that the Lyapunov equation (9) with parameters F and Q has been modified to equation (25), which is a Lyapunov equation with well-defined parameters L and $\tilde{Q}$.

Then, using (16)

$$\tilde{P} = \lim_{k \rightarrow \infty} \tilde{P}(k|k-1) = \lim_{k \rightarrow \infty} \tilde{P}(k+1/k)$$

arises the limiting solution of (25), which satisfies the modified algebraic Lyapunov equation:

$$\tilde{P} = \tilde{Q} + L\tilde{P}L. \quad (26)$$

It is clear that the solution of the Lyapunov equation (9) can be computed by the solution of the modified Lyapunov equation (25) due to (16) and (24):

$$P = W\tilde{P}W^T. \quad (27)$$

Table 1 summarizes the classical and the derived iterative algorithms for solving the Riccati and Lyapunov equations.

Table 1: Riccati and Lyapunov equations iterative solution algorithms

Equation		Algorithm
Riccati equation	Classical	input: F, H, Q, R initialization: $L = \text{eigenvalues of } F$ iteration $P(k+1 k) = Q + FP(k k-1)F^T - FP(k k-1)H^T[HP(k k-1)H^T + R]^{-1}HP(k k-1)F^T$ convergence: $P = \lim_{k \rightarrow \infty} P(k k-1)$ output: P
	Proposed	input: F, H, Q, R initialization: $L = \text{eigenvalues of } F, W = \text{eigenvectors of } F, \tilde{Q} = W^{-1}QW^{-T}, \tilde{H} = HW$ iteration $\tilde{P}(k+1 k) = \tilde{Q} + L\tilde{P}(k k-1)L - L\tilde{P}(k k-1)\tilde{H}^T[\tilde{H}\tilde{P}(k k-1)\tilde{H}^T + R]^{-1}\tilde{H}\tilde{P}(k k-1)L$ convergence: $\tilde{P} = \lim_{k \rightarrow \infty} \tilde{P}(k k-1)$ finalization: $P = W\tilde{P}W^T$ output: P
Transformed Riccati equation	Classical	input: F, H, Q, R initialization: $L = \text{eigenvalues of } F, B = H^TR^{-1}H$ iteration $P(k+1 k) = Q + F[P^{-1}(k k-1) + B]^{-1}F^T$ convergence: $P = \lim_{k \rightarrow \infty} P(k k-1)$ output: P
	Proposed	input: F, H, Q, R initialization: $L = \text{eigenvalues of } F, W = \text{eigenvectors of } F, B = H^TR^{-1}H, \tilde{Q} = W^{-1}QW^{-T}, \tilde{B} = W^TBW$ iteration $\tilde{P}(k+1 k) = \tilde{Q} + L[\tilde{P}^{-1}(k k-1) + \tilde{B}]^{-1}L$ convergence: $\tilde{P} = \lim_{k \rightarrow \infty} \tilde{P}(k k-1)$ finalization: $P = W\tilde{P}W^T$ output: P
Lyapunov equation	Classical	input: F, Q initialization: $L = \text{eigenvalues of } F$ iteration $P(k+1 k) = Q + FP(k k-1)F^T$ convergence: $P = \lim_{k \rightarrow \infty} P(k k-1)$ output: P
	Proposed	input: F, Q initialization: $L = \text{eigenvalues of } F, W = \text{eigenvectors of } F, \tilde{Q} = W^{-1}QW^{-T}$ iteration $\tilde{P}(k+1 k) = \tilde{Q} + L\tilde{P}(k k-1)L$ convergence: $\tilde{P} = \lim_{k \rightarrow \infty} \tilde{P}(k k-1)$ finalization: $P = W\tilde{P}W^T$ output: P

3 Comparison of classical and proposed algorithms

It is established that the proposed iterative algorithms for solving the Riccati and Lyapunov equations have been derived by the classical iterative algorithms. Thus, the classical and the derived algorithms are equivalent with respect to their behavior since they calculate theoretically the same solutions executing the same number of iterations s .

We are going to compare the classical and the derived algorithms with respect to their calculation burdens. Scalar operations are involved in matrix manipulation operations, which are needed for the implementation of the algorithms. Table 2 summarizes the calculation burdens of needed matrix operations. The computation of the eigenvalues is achieved by solving the characteristic equation using the Newton-Raphson method, Laguerre's method, Regula-Falsi method or Bernoulli's method, because each method requires different computations for solving the characteristic equation. In the following, we consider that all the algorithms apply the same method; thus the computation of the eigenvalues has the same calculation burden, which is denoted as CB_L . The eigenvectors computation is achieved using Gauss elimination and backward substitutions. In fact, Gauss elimination requires $\frac{1}{6}(4n^3 + 9n^2 - 7n)$ scalar operations; analytically, $\frac{1}{2}(n^2 + n)$ divisions, $\frac{1}{6}(2n^3 + 3n^2 - 5n)$ multiplications, and $\frac{1}{6}(2n^3 + 3n^2 - 5n)$ subtractions; the details are given in [19]. Backward substitutions require n^2 scalar operations: analytically, $\sum_{i=1}^{n-1} i = \frac{(n-1)n}{2}$ multiplications and $\sum_{i=1}^{n-1} i = \frac{(n-1)n}{2}$ and n divisions. Thus, the eigenvalues computation requires $\frac{1}{6}(4n^3 + 15n^2 - 7n)$ scalar operations. The details for the matrix operations are given in [20].

Table 2: Calculation burdens of matrix operations

Matrix operation	Matrix dimensions	Calculation burden
$L = \text{eigenvalues of } F$	$n \times n$	CB_L
$W = \text{eigenvectors of } F$	$n \times n$	$\frac{1}{6}(4n^3 + 15n^2 - 7n)$
$C = A + B$	$(n \times m) + (n \times m)$	nm
⁽¹⁾ $S = A + B$	$(n \times n) + (n \times n)$	$\frac{1}{2}n^2 + \frac{1}{2}n$
$C = A \cdot B$	$(n \times m) \cdot (m \times \ell)$	$2nm\ell - n\ell$
⁽¹⁾ $S = A \cdot B$	$(n \times m) \cdot (m \times n)$	$n^2m + nm - \frac{1}{2}n^2 - \frac{1}{2}n$
⁽²⁾ $A \cdot D$	$(n \times n) \cdot (n \times n)$	n^2
⁽²⁾ $D \cdot A$	$(n \times n) \cdot (n \times n)$	n^2
⁽³⁾ $B = A^{-1}$	$n \times n$	$\frac{1}{6}(16n^3 - 3n^2 - n)$

⁽¹⁾ S is a symmetric matrix.

⁽²⁾ D is a diagonal matrix.

⁽³⁾ For the general multi-dimensional case, where $n \geq 2$.

Table 3 summarizes the calculation burdens of the classical and the proposed algorithms, for the general multidimensional case, where $n \geq 2$ and $m \geq 2$. The details are given in the Appendix. In this table,

Table 3: Calculation burdens of algorithms

Equation	Algorithm	Calculation burden
Riccati equation	Classical	$CB_{CRE} = CB_L + \left(\frac{1}{6}(16m^3 - 3m^2 - m) + 3n^3 + 3n^2m + 3nm^2\right)s$
	Proposed	$CB_{PRE} = CB_L + \frac{1}{6}(56n^3 + 6n^2 - 14n) + 2n^2m - nm + \left(\frac{1}{6}(16m^3 - 3m^2 - m) + \frac{1}{2}(5n^2 + n) + 3n^2m + 3nm^2\right)s$
Transformed Riccati equation	Classical	$CB_{CTRE} = CB_L + \frac{1}{6}(16m^3 - 3m^2 - m) + 2nm^2 + n^2m - \frac{1}{2}n^2 - \frac{1}{2}n + \frac{1}{6}(50n^3 - 3n^2 + n)s$
	Proposed	$CB_{PTRE} = CB_L + \frac{1}{6}(16m^3 - 3m^2 - m) + 2nm^2 + n^2m + \frac{1}{6}(74n^3 - 20n) + \frac{1}{6}(32n^3 + 12n^2 + 4n)s$
Lyapunov equation	Classical	$CB_{CLE} = CB_L + 3n^3s$
	Proposed	$CB_{PLE} = CB_L + \frac{1}{6}(56n^3 + 6n^2 - 14n) + \frac{1}{2}(5n^2 + n)s$

the calculation burden CB_L for the computation of the eigenvalues is the same for all algorithms, since we consider that the computation is achieved by solving the characteristic equation using the same method.

Concerning the *Riccati equation* solution algorithms, from Table 3, we can write:

$$CB_{CRE} - CB_{PRE} = \frac{n}{6}\{(18n^2 - 15n - 3)s - (56n^2 + 6n - 14) - (12n - 6)m\}. \quad (28)$$

Note that in the case

$$s > \frac{56n^2 + 6n - 14}{18n^2 - 15n - 3} + \frac{12n - 6}{18n^2 - 15n - 3}m = 3 + \frac{2n^2 + 51n - 5}{18n^2 - 15n - 3} + \frac{12n - 6}{18n^2 - 15n - 3}m,$$

since $18n^2 - 15n - 3 > 0$, for every $n \geq 2$, equation (28) yields

$$CB_{CRE} - CB_{PRE} > 0.$$

The aforementioned notation leads to the conclusion that the proposed algorithm is faster than the classical algorithm for a small number of iterations, especially when the state and measurement dimensions are large.

Figure 1 depicts the faster algorithm with respect to the state dimension n , the measurement dimension m , and the number of iterations s the algorithms execute. The proposed algorithm is faster than the classical algorithm in the region under the curves and is slower than the classical algorithm in the region above the curves, confirming the aforementioned note.

Concerning the *transformed Riccati equation* solution algorithms, from Table 3, we obtain:

$$CB_{CTRE} - CB_{PTRE} = \frac{n}{6}\{(18n^2 - 15n - 3)s - (74n^2 + 3n - 17)\}. \quad (29)$$

Note that in the case

$$s > \frac{74n^2 + 3n - 17}{18n^2 - 15n - 3} = 4 + \frac{2n^2 + 63n - 5}{18n^2 - 15n - 3},$$

since $18n^2 - 15n - 3 > 0$, for every $n \geq 2$, equation (29) yields

$$CB_{CTRE} - CB_{PTRE} > 0.$$

The aforementioned notation leads to the conclusion that the proposed algorithm is faster than the classical algorithm for a small number of iterations. In fact, the proposed algorithm is always faster than the classical algorithm, when $s > 7$. In addition, the proposed algorithm outperforms the classical one, when the number of iterations is greater than 4 as the state dimension increases.

Figure 2 depicts the faster algorithm with respect to the state dimension n and the number of iterations s the algorithms execute. The proposed algorithm is faster than the classical algorithm in the region under the curve and is slower than the classical algorithm in the region above the curve, confirming the aforementioned note.

Concerning the *Lyapunov equation* solution algorithms, from Table 3, we obtain:

$$CB_{CLE} - CB_{PLE} = \frac{n}{6}\{(18n^2 - 15n - 3)s - (56n^2 + 6n - 14)\}. \quad (30)$$

Note that in the case

$$s > \frac{56n^2 + 6n - 14}{18n^2 - 15n - 3} = 3 + \frac{2n^2 + 51n - 5}{18n^2 - 15n - 3},$$

since $18n^2 - 15n - 3 > 0$, for every $n \geq 2$, equation (30) yields

$$CB_{CLE} - CB_{PLE} > 0$$

The aforementioned notation leads to the conclusion that the proposed algorithm is faster than the classical algorithm for a small number of iterations. In fact, the proposed algorithm is always faster than the classical algorithm, when $s > 5$. In addition, the proposed algorithm outperforms the classical one, when the number of iterations is greater than 3, as the state dimension increases.

Figure 3 depicts the faster algorithm with respect to the state dimension n and the number of iterations s the algorithms execute. The proposed algorithm is faster than the classical algorithm in the region under the curve and is slower than the classical algorithm in the region above the curve, confirming the aforementioned note.

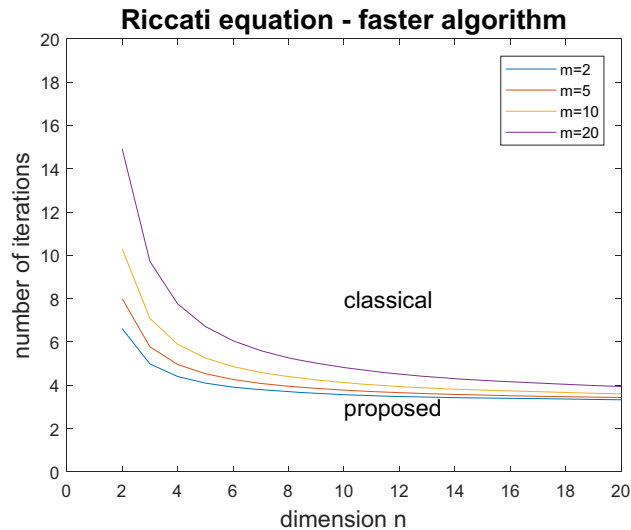


Figure 1: Comparison of the Riccati equation solution algorithms.

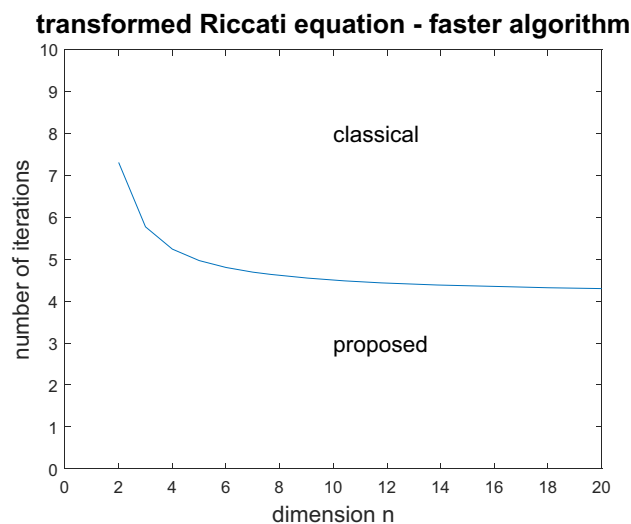


Figure 2: Comparison of transformed Riccati equation solution algorithms.

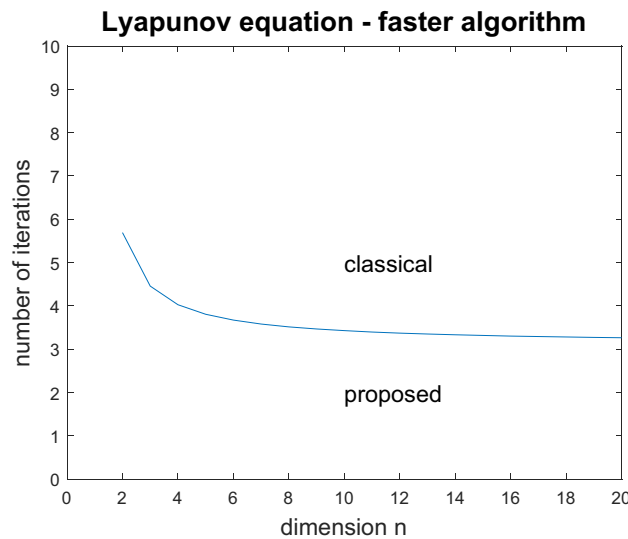


Figure 3: Comparison of Lyapunov equation solution algorithms.

4 Conclusions

Riccati and Lyapunov equations are very important in many fields of science. The existence of their solutions involves the knowledge of the eigenvalues of the transition matrix. We have developed new fast iterative algorithms for solving the Riccati and Lyapunov equations. The proposed algorithms take advantage of the knowledge of the eigenvalues and eigenvectors of the transition matrix. The proposed algorithms hold in the case where the transition matrix is diagonalizable in the form $=WLW^{-1}$, where the diagonal matrix L contains the real eigenvalues of F and the matrix W consists of columns containing the linearly independent eigenvectors of F . It is shown that the proposed algorithms may be faster than the classical ones depending on the state dimension, the measurement dimension, and the number of iterations executed, even for a small number of iterations. The proposed algorithms are faster than the classical ones for such number of iterations that decreases as the state dimension increases. This is rational because the operations that involve the eigenvalue matrix L are of the order of $O(n^2)$, while operations that involve the eigenvalue matrix F are of the order of $O(n^3)$.

More specifically,

- (a) Concerning the Riccati equation, the proposed solution is faster than the classical one for a small number of iterations, especially when the state and measurement dimensions are large.
- (b) Concerning the transformed Riccati equation, the proposed solution is always faster than the classical one when the number of iterations is greater than 7; the proposed algorithm outperforms the classical one when the number of iterations is greater than 4, as the state dimension increases.
- (c) Concerning the Lyapunov equation, the proposed solution is always faster than the classical one when the number of iterations is greater than 5; the proposed algorithm outperforms the classical one when the number of iterations is greater than 3, as the state dimension increases.

It is evident that the proposed iterative algorithms outperform the classical ones even for a small number of iterations.

Acknowledgements: The authors thank the anonymous reviewers for their helpful comments.

Funding information: The authors state that no funding was involved.

Author contributions: The authors applied the SDC approach for the sequence of authors. NA: conceptualization and performed the simulation. MA: performed the formal analysis and prepared the manuscript.

Conflict of interest: The authors state no conflict of interest.

Data availability statement: All data generated or analyzed during this study are included in this published article.

References

- [1] B. D. O. Anderson and J. B. Moore, *Optimal Filtering*, Dover Publications, New York, 2005.
- [2] N. Assimakis and M. Adam, *Fast doubling algorithm for the solution of the riccati equation using cyclic reduction method*, 2020 International Conference on Mathematics and Computers in Science and Engineering (MACISE), 2020, pp. 1–5, <https://doi.org/10.1109/MACISE49704.2020.00007>.
- [3] D. S. Bernstein, *Matrix Mathematics: Theory Facts and Formulas with Application to Linear Systems Theory*, Princeton University Press, Princeton, NJ, 2005.
- [4] C. Kojima, K. Takaba, O. Kaneko, and P. Rapisarda, *A characterization of solutions of the discrete-time algebraic riccati equation based on quadratic difference forms*, Linear Algebra Appl. **416** (2006), 1060–1082.
- [5] A. Nakhmani, *Modern Control: State-Space Analysis and Design Methods*, McGraw Hill, New York, 2020.
- [6] N. D. Assimakis, D. G. Lainiotis, S. K. Katsikas, and F. L. Sanida, *A survey of recursive algorithms for the solution of the discrete time Riccati equation*, Nonlinear Anal. (1997), no. 4, 2409–2420.
- [7] D. G. Lainiotis, N. D. Assimakis, and S. K. Katsikas, *A new computationally effective algorithm for solving the discrete Riccati equation*, J. Math. Anal. Appl. **186** (1994), no. 3, 868–895.
- [8] D. G. Lainiotis, N. D. Assimakis, and S. K. Katsikas, *Fast and numerically robust recursive algorithms for solving the discrete time Riccati equation: The case of nonsingular plant noise covariance matrix*, Neural, Parallel Sci. Comput. **3** (1995), 565–584.
- [9] N. Assimakis, S. Roulis, and D. Lainiotis, *Recursive solutions of the discrete time riccati equation*, Neural, Parallel Sci. Comput. **11** (2003), 343–350.
- [10] V. Dragan, *The linear quadratic optimization problem for a class of singularly stochastic systems*, Int. J. Innov. Comput. Inf. Control **1** (2005), 53–63.
- [11] N. Komaroff, *Iterative matrix bounds and computational solutions to the discrete algebraic Riccati equation*, IEEE Trans. Autom. Control **39** (1994), 1676–1678.
- [12] J. Zhang and J. Liu, *New upper and lower bounds, the iteration algorithm for the solution of the discrete algebraic Riccati equation*, Adv. Differ. Equ. **313** (2015), 1–17, <https://doi.org/10.1186/s13662-015-0649-6>.
- [13] N. Assimakis and M. Adam, *Closed form solutions of Lyapunov equations using the vech and veck operators*, WSEAS Trans. Math. **20** (2021), 276–282, <https://doi.org/10.37394/23206.2021.20.28>.
- [14] M. S. Grewal and A. P. Andrews, *Kalman Filtering: Theory and Practice Using MATLAB*, John Wiley and Sons, Inc, New York, 2001.
- [15] T. Grindinger, *Eye movement analysis & prediction with the Kalman filter*, (MS thesis), Clemson University, 2006.
- [16] J. P. Dubois, J. S. Daba, M. Nader, and C. El Ferkh, *GSM position tracking using a Kalman filter*, Int. J. Electron. Commun. Eng. **6** (2012), no. 8, 867–876, <https://publications.waset.org/vol/68>.
- [17] N. Assimakis and M. Adam, *Mobile position tracking in three dimensions using Kalman and Lainiotis filters*, Open Math. J. **8** (2015), 1–6, <http://doi.org/10.2174/1874117701508010001>.
- [18] R. Zanetti and C. D'Souza, *Recursive implementations of the Schmidt-Kalman 'consider' filter*, J. of Astronaut. Sci. **60** (2013), no. 3–4, 672–685, <http://doi.org/10.1007/s40295-015-0068-7>.
- [19] R. W. Farebrother, *Linear Least Squares Computations, STATISTICS: Textbooks and Monographs*, Marcel Dekker, New York, 1988.
- [20] N. Assimakis and M. Adam, *Discrete time Kalman and Lainiotis filters comparison*, Internat. J. Math. Anal. **1** (2007), no. 13, 635–659.

Appendix

The calculation burdens of the classical and the proposed algorithms for solving the Riccati and Lyapunov equations, for the general multi-dimensional case (where $n \geq 2$, $m \geq 2$), are analytically presented.

Riccati equation – classical iterative algorithm

Initialization

$L = \text{eigenvalues of } F$

CB_L

Iteration (s iterations)

$M_1 = HP(k k - 1)$	$(m \times n) \cdot (n \times n)$	$2n^2m - nm$
$M_2 = M_1 H^T$	$(m \times n) \cdot (n \times m)$	$nm^2 + nm - \frac{1}{2}m^2 - \frac{1}{2}m$
$M_3 = M_2 + R$	$(m \times m) + (m \times m)$	$\frac{1}{2}m^2 + \frac{1}{2}m$
$M_4 = M_3^{-1}$	$(m \times m)$	$\frac{1}{6}(16m^3 - 3m^2 - m)$
$M_5 = M_4 M_1$	$(m \times m) \cdot (m \times n)$	$2nm^2 - nm$
$M_6 = M_1^T M_5$	$(n \times m) \cdot (m \times n)$	$n^2m + nm - \frac{1}{2}n^2 - \frac{1}{2}n$
$M_7 = P(k/k - 1) - M_6$	$(n \times n) + (n \times n)$	$\frac{1}{2}n^2 + \frac{1}{2}n$
$M_8 = FM_7$	$(n \times n) \cdot (n \times n)$	$2n^3 - n^2$
$M_9 = M_8 F^T$	$(n \times n) \cdot (n \times n)$	$n^3 + \frac{1}{2}n^2 - \frac{1}{2}n$
$P(k + 1/k) = Q + M_9$	$(n \times n) + (n \times n)$	$\frac{1}{2}n^2 + \frac{1}{2}n$
$CB_{CRE} = CB_L + \left(\frac{1}{6}(16m^3 - 3m^2 - m) + 3n^3 + 3n^2m + 3nm^2\right)s$		

Riccati equation – proposed iterative algorithm

Initialization

$L = \text{eigenvalues of } F$

CB_L

$W = \text{eigenvectors of } F$

W^{-1}	$(n \times n)$	$\frac{1}{6}(4n^3 + 15n^2 - 7n)$
$W^{-1}Q$	$(n \times n) \cdot (n \times n)$	$2n^3 - n^2$
$\tilde{Q} = W^{-1}QW^{-T}$	$(n \times n) \cdot (n \times n)$	$n^3 + \frac{1}{2}n^2 - \frac{1}{2}n$
$\tilde{H} = HW$	$(m \times n) \cdot (n \times n)$	$2n^2m - nm$
Iteration (s iterations)		
$M_1 = \tilde{H}\tilde{P}(k k - 1)$	$(m \times n) \cdot (n \times n)$	$2n^2m - nm$
$M_2 = M_1 \tilde{H}^T$	$(m \times n) \cdot (n \times m)$	$nm^2 + nm - \frac{1}{2}m^2 - \frac{1}{2}m$
$M_3 = M_2 + R$	$(m \times m) + (m \times m)$	$\frac{1}{2}m^2 + \frac{1}{2}m$
$M_4 = M_3^{-1}$	$(m \times m)$	$\frac{1}{6}(16m^3 - 3m^2 - m)$
$M_5 = M_4 M_1$	$(m \times m) \cdot (m \times n)$	$2nm^2 - nm$
$M_6 = M_1^T M_5$	$(n \times m) \cdot (m \times n)$	$n^2m + nm - \frac{1}{2}n^2 - \frac{1}{2}n$
$M_7 = \tilde{P}(k/k - 1) - M_6$	$(n \times n) + (n \times n)$	$\frac{1}{2}n^2 + \frac{1}{2}n$
$M_8 = LM_7$	$(n \times n) \cdot (n \times n)$	n^2
$M_9 = M_8 L$	$(n \times n) \cdot (n \times n)$	n^2
$\tilde{P}(k + 1/k) = \tilde{Q} + M_9$	$(n \times n) + (n \times n)$	$\frac{1}{2}n^2 + \frac{1}{2}n$

Finalization

$$\begin{array}{lll}
 W\tilde{P} & (n \times n) \cdot (n \times n) & 2n^3 - n^2 \\
 P = W\tilde{P}W^T & (n \times n) \cdot (n \times n) & n^3 + \frac{1}{2}n^2 - \frac{1}{2}n \\
 CB_{PRE} = CB_L + \frac{1}{6}(56n^3 + 6n^2 - 14n) + 2n^2m - nm + \left(\frac{1}{6}(16m^3 - 3m^2 - m) + \frac{1}{2}(5n^2 + n) + 3n^2m + 3nm^2\right)s & &
 \end{array}$$

Transformed Riccati equation – classical iterative algorithm

Initialization

$$\begin{array}{lll}
 L = \text{eigenvalues of } F & & CB_L \\
 R^{-1} & (m \times m) & \frac{1}{6}(16m^3 - 3m^2 - m) \\
 H^TR^{-1} & (n \times m) \cdot (m \times m) & 2nm^2 - nm \\
 B = H^TR^{-1}H & (n \times m) \cdot (m \times n) & n^2m + nm - \frac{1}{2}n^2 - \frac{1}{2}n \\
 \text{Iteration (s iterations)} & & \\
 P^{-1}(k | k - 1) & (n \times n) & \frac{1}{6}(16n^3 - 3n^2 - n) \\
 P^{-1}(k | k - 1) + B & (n \times n) + (n \times n) & \frac{1}{2}n^2 + \frac{1}{2}n \\
 [P^{-1}(k | k - 1) + B]^{-1} & (n \times n) & \frac{1}{6}(16n^3 - 3n^2 - n) \\
 F[P^{-1}(k | k - 1) + B]^{-1} & (n \times n) \cdot (n \times n) & 2n^3 - n^2 \\
 F[P^{-1}(k | k - 1) + B]^{-1}F^T & (n \times n) \cdot (n \times n) & n^3 + \frac{1}{2}n^2 - \frac{1}{2}n \\
 P(k + 1 | k) = Q + F[P^{-1}(k | k - 1) + B]^{-1}F^T & (n \times n) + (n \times n) & \frac{1}{2}n^2 + \frac{1}{2}n \\
 CB_{CTRE} = CB_L + \frac{1}{6}(16m^3 - 3m^2 - m) + 2nm^2 + n^2m - \frac{1}{2}n^2 - \frac{1}{2}n + \frac{1}{6}(50n^3 - 3n^2 + n)s & &
 \end{array}$$

Transformed Riccati equation – proposed iterative algorithm

Initialization

$$\begin{array}{lll}
 L = \text{eigenvalues of } F & & CB_L \\
 W = \text{eigenvectors of } F & & \frac{1}{6}(4n^3 + 15n^2 - 7n) \\
 W^{-1} & (n \times n) & \frac{1}{6}(16n^3 - 3n^2 - n) \\
 W^{-1}Q & (n \times n) \cdot (n \times n) & 2n^3 - n^2 \\
 \tilde{Q} = W^{-1}QW^{-T} & (n \times n) \cdot (n \times n) & n^3 + \frac{1}{2}n^2 - \frac{1}{2}n \\
 R^{-1} & (m \times m) & \frac{1}{6}(16m^3 - 3m^2 - m) \\
 H^TR^{-1} & (n \times m) \cdot (m \times m) & 2nm^2 - nm \\
 B = H^TR^{-1}H & (n \times m) \cdot (m \times n) & n^2m + nm - \frac{1}{2}n^2 - \frac{1}{2}n \\
 W^TB & (n \times n) \cdot (n \times n) & 2n^3 - n^2 \\
 \tilde{B} = W^TBW & (n \times n) \cdot (n \times n) & n^3 + \frac{1}{2}n^2 - \frac{1}{2}n \\
 \text{Iteration (s iterations)} & & \\
 \tilde{P}^{-1}(k | k - 1) & (n \times n) & \frac{1}{6}(16n^3 - 3n^2 - n) \\
 \tilde{P}^{-1}(k | k - 1) + \tilde{B} & (n \times n) + (n \times n) & \frac{1}{2}n^2 + \frac{1}{2}n \\
 [\tilde{P}^{-1}(k | k - 1) + \tilde{B}]^{-1} & (n \times n) & \frac{1}{6}(16n^3 - 3n^2 - n) \\
 L[\tilde{P}^{-1}(k | k - 1) + \tilde{B}]^{-1} & (n \times n) \cdot (n \times n) & n^2 \\
 L[\tilde{P}^{-1}(k | k - 1) + \tilde{B}]^{-1}L & (n \times n) \cdot (n \times n) & n^2 \\
 \tilde{P}(k + 1 | k) = \tilde{Q} + L[\tilde{P}^{-1}(k | k - 1) + \tilde{B}]^{-1}L & (n \times n) + (n \times n) & \frac{1}{2}n^2 + \frac{1}{2}n
 \end{array}$$

Finalization

$$\begin{array}{lll}
 W\tilde{P} & (n \times n) \cdot (n \times n) & 2n^3 - n^2 \\
 P = W\tilde{P}W^T & (n \times n) \cdot (n \times n) & n^3 + \frac{1}{2}n^2 - \frac{1}{2}n \\
 CB_{PTRE} = CB_L + \frac{1}{6}(16m^3 - 3m^2 - m) + 2nm^2 + n^2m + \frac{1}{6}(74n^3 - 20n) + \frac{1}{6}(32n^3 + 12n^2 + 4n)s
 \end{array}$$

Lyapunov equation – classical iterative algorithm

Initialization

$$L = \text{eigenvalues of } F \quad CB_L$$

iteration (s iterations)

$$\begin{array}{lll}
 FP(k | k - 1) & (n \times n) \cdot (n \times n) & 2n^3 - n^2 \\
 FP(k | k - 1)F^T & (n \times n) \cdot (n \times n) & n^3 + \frac{1}{2}n^2 - \frac{1}{2}n \\
 P(k + 1 | k) = Q + FP(k | k - 1)F^T & (n \times n) + (n \times n) & \frac{1}{2}n^2 + \frac{1}{2}n
 \end{array}$$

$$CB_{CLE} = CB_L + 3n^3s$$

Lyapunov equation – proposed iterative algorithm

Initialization

$$L = \text{eigenvalues of } F \quad CB_L$$

$$W = \text{eigenvectors of } F \quad \frac{1}{6}(4n^3 + 15n^2 - 7n)$$

$$W^{-1} \quad (n \times n) \quad \frac{1}{6}(16n^3 - 3n^2 - n)$$

$$W^{-1}Q \quad (n \times n) \cdot (n \times n) \quad 2n^3 - n^2$$

$$\tilde{Q} = W^{-1}QW^{-T} \quad (n \times n) \cdot (n \times n) \quad n^3 + \frac{1}{2}n^2 - \frac{1}{2}n$$

Iteration (s iterations)

$$L\tilde{P}(k + 1 | k) \quad (n \times n) \cdot (n \times n) \quad n^2$$

$$L\tilde{P}(k + 1 | k)L \quad (n \times n) \cdot (n \times n) \quad n^2$$

$$\tilde{P}(k + 1 | k) = \tilde{Q} + L\tilde{P}(k | k - 1)L \quad (n \times n) + (n \times n) \quad \frac{1}{2}n^2 + \frac{1}{2}n$$

Finalization

$$W\tilde{P} \quad (n \times n) \cdot (n \times n) \quad 2n^3 - n^2$$

$$P = W\tilde{P}W^T \quad (n \times n) \cdot (n \times n) \quad n^3 + \frac{1}{2}n^2 - \frac{1}{2}n$$

$$CB_{PLE} = CB_L + \frac{1}{6}(56n^3 + 6n^2 - 14n) + \frac{1}{2}(5n^2 + n)s$$
