

Article

The Iterative Solution to Discrete-Time H_∞ Control Problems for Periodic Systems

Ivan G. Ivanov * and Boryana C. Bogdanova

Faculty of Economics and Business Administration, Sofia University “St. Kliment Ohridski”,
125 Tsarigradsko Shosse Blvd., bl.3, 1113 Sofia, Bulgaria; bpeleva@gmail.com

* Correspondence: i_ivanov@feb.uni-sofia.bg; Tel.: +359-898-636-339

Academic Editors: Alicia Cordero, Juan R. Torregrosa and Francisco I. Chicharro

Received: 22 October 2015; Accepted: 9 March 2016; Published: 14 March 2016

Abstract: This paper addresses the problem of solving discrete-time H_∞ control problems for periodic systems. The approach for solving such a type of equations is well known in the literature. However, the focus of our research is set on the numerical computation of the stabilizing solution. In particular, two effective methods for practical realization of the known iterative processes are described. Furthermore, a new iterative approach is investigated and applied. On the basis of numerical experiments, we compare the presented methods. A major conclusion is that the new iterative approach is faster than rest of the methods and it uses less RAM memory than other methods.

Keywords: H_∞ optimal control problem; generalized riccati equation; periodic systems; stabilizing solution

1. Introduction

Periodic systems are an important sub-class of time-varying systems therefore their investigation has attracted significant attention during the past years. The theory of stability, optimal control and applications of periodic systems is available in several recent monographs [1–4] as well as in the references therein. The so called stabilizing solution of discrete-time periodic systems has a crucial role in solving a linear linear quadratic optimal control problem on infinite time horizon. There are different methods to solve the periodic Riccati equations with a definite quadratic part [5,6]. Recently, [7,8] two iterative procedures have been developed for numerical computation of the stabilizing solution of a class of periodic discrete-time Riccati equation with an indefinite sign of it's quadratic term.

Our aim is to find the stabilizing solution of a set of periodic discrete-time Riccati equations with indefinite sign in the quadratic part:

$$X(t) = \mathcal{R}(t, X) := A^T(t)X(t+1)A(t) - [A^T(t)X(t+1)B(t) + C^T(t)D(t)] + C^T(t)C(t) \times [R_\gamma(t) + B^T(t)X(t+1)B(t)]^{-1} [A^T(t)X(t+1)B(t) + C^T(t)D(t)]^T \quad (1)$$

where $t \in \mathbb{Z}$, $A(t) \in \mathbb{R}^{n \times n}$, $B(t) = (B_1(t) \quad B_2(t)), B_k(t) \in \mathbb{R}^{n \times m_k}, k = 1, 2$, $C(t) \in \mathbb{R}^{p \times n}$, $D(t) = (D_1(t) \quad D_2(t)), D_k(t) \in \mathbb{R}^{p \times m_k}, k = 1, 2$, $R_\gamma(t) = D^T(t)D(t) + \begin{pmatrix} -\gamma^2 I_{m_1} & 0 \\ 0 & 0 \end{pmatrix} \in \mathbb{R}^{m \times m}$, $m = m_1 + m_2$, $\gamma > 0$ is a given scalar. Our investigation is based on the theory of periodic discrete-time Riccati equations as well as on the iterative methods described in [7]. Dragan *et al.* [7,8] study two iterative processes for the computation of the stabilizing solution to Equation (1) and derive their convergence properties. However, the proposed iterative methods are not investigated numerically. In this paper we consider two iterative methods (Block Successive Approximations and A Block Stein Iteration) so as to develop numerical procedures (derived in [7,8]) for computing the stabilizing solution of a set of periodic discrete-time Riccati equations.

We further propose a new iterative procedure for the practical realization of the main iterative processes. The performed numerical experiments are supportive to the fact that the new procedure is faster than the others, furthermore, it uses less RAM memory. We should note that our study is based on ideas developed in [9–11], where new iterative procedures for the numerical computation of a stabilizing solution in the time-invariant case has been proposed.

The main assumption for the existence of such a type set of periodic discrete-time Riccati equations is:

H1: An integer exists $\theta \geq 1$ such that $A(t + \theta) = A(t), B(t + \theta) = B(t), C(t + \theta) = C(t), D(t + \theta) = D(t)$ for $t \in \mathbb{Z}$.

In our approach the class of admissible solutions consists of all bounded sequences $\{X(t)\}_{t \in \mathbb{Z}} \subset \mathcal{S}_n$ satisfying Equation (1) and the following two sign conditions:

$$D_1^T(t)D_1(t) + B_1^T(t)X(t+1)B_1(t) - \left[B_1^T(t)X(t+1)B_2(t) + D_1^T(t)D_2(t) \right] \\ \times \left[B_2^T(t)X(t+1)B_2(t) + D_2^T(t)D_2(t) \right]^{-1} \left[B_1^T(t)X(t+1)B_2(t) + D_1^T(t)D_2(t) \right]^T - \gamma^2 I_{m_1} < 0 \quad (2)$$

$$D_2^T(t)D_2(t) + B_2^T(t)X(t+1)B_2(t) > 0 \quad (3)$$

$t \in \mathbb{Z}$. Therefore, we are interested in the global and bounded on $\mathbb{Z}$ solutions for which the matrices $R_\gamma(t) + B^T(t)X(t+1)B(t), t \in \mathbb{Z}$ have not defined sign.

Definition 1. [7] An admissible solution $\{\tilde{X}(t)\}_{t \in \mathbb{Z}}$ is called **stabilizing solution**, if the zero state equilibrium of the discrete-time linear system on $\mathbb{R}^n$:

$$x(t+1) = [A(t) + B(t)\tilde{F}(t)]x(t) \quad (4)$$

is exponentially stable, where

$$\tilde{F}(t) = -[R_\gamma(t) + B^T(t)\tilde{X}(t+1)B(t)]^{-1} [B^T(t)\tilde{X}(t+1)A(t) + D^T(t)C(t)] \quad (5)$$

2. Iterative Methods for Stabilizing Solution

2.1. The Iterative Procedure with Two Matrix Sequences

We consider the iterative procedure for numerical computation of the unique stabilizing and θ -periodic solution $\tilde{X}(\cdot)$ of Equation (1), which is described in Section 3.1 of the paper of Dragan and co-authors [7]. In fact, the iterative procedure is based on the solution to θ -periodic discrete-time Riccati equation with definite quadratic part:

$$Z^{(k)}(t) = \mathbb{W}(Z^{(k)}) := (\hat{A}(t))^T Z^{(k)}(t+1) \hat{A}(t) - \left[(\hat{A}(t))^T Z^{(k)}(t+1) B_2(t) \right] \\ \times \left[R_2^{(k)}(t) + B_2(t) Z^{(k)}(t+1) B_2(t) \right]^{-1} \left[(\hat{A}(t))^T Z^{(k)}(t+1) B_2(t) \right]^T + M^{(k)}(t) \quad (6)$$

with $\hat{A}(t) = A(t) + B(t)F^{(k)}(t)$ where

$$F^{(k)}(t) = -[R_\gamma(t) + B^T(t)X^{(k)}(t+1)B(t)]^{-1} [B^T(t)X^{(k)}(t+1)A(t) + D^T(t)C(t)] \\ R_2^{(k)}(t) = D_2^T(t)D_2(t) + B_2^T(t)X^{(k)}(t+1)B_2(t) \\ M^{(k)}(t) = A^T(t)X^{(k)}(t+1)A(t) - [A^T(t)X^{(k)}(t+1)B(t) + C^T(t)D(t)] + C^T(t)C(t) \\ \times [R_\gamma(t) + B^T(t)X^{(k)}(t+1)B(t)]^{-1} [A^T(t)X^{(k)}(t+1)B(t) + C^T(t)D(t)]^T - X^{(k)}(t)$$

We rewrite the above equations ($t = 0, \dots, \theta$) using a block structure. We denote $\check{Z} = \text{diag}(Z(0), Z(1), \dots, Z(\theta))$ and J is a permutation matrix such that

$$J^{-1} \check{Z} J = \text{diag}(Z(1), Z(2), \dots, Z(\theta), Z(0)), \quad (J^{-1} = J^T)$$

Introducing the following notations

$$\begin{cases} \check{A} &= \text{diag}(\hat{A}(0), \hat{A}(1), \dots, \hat{A}(\theta)), \quad \check{B}_2 = \text{diag}(B_2(0), B_2(1), \dots, B_2(\theta)) \\ \check{R}_2^{(k)} &= \text{diag}(R_2^{(k)}(0), R_2^{(k)}(1), \dots, R_2^{(k)}(\theta)) \\ \check{M}^{(k)} &= \text{diag}(M^{(k)}(0), M^{(k)}(1), \dots, M^{(k)}(\theta)) \\ \check{F}_{\check{Z}^{(k)}} &= - \left[\check{R}_2^{(k)} + \check{B}_2^T J^{-1} \check{Z}^{(k)} J \check{B}_2 \right]^{-1} (\check{A}^T J^{-1} \check{Z}^{(k)} J \check{B}_2)^T \end{cases}$$

Equation (6) has a form:

$$\check{Z}^{(k)} = \check{A}^T J^{-1} \check{Z}^{(k)} J \check{A} - (\check{A}^T J^{-1} \check{Z}^{(k)} J \check{B}_2) \times \left[\check{R}_2^{(k)} + \check{B}_2^T J^{-1} \check{Z}^{(k)} J \check{B}_2 \right]^{-1} (\check{A}^T J^{-1} \check{Z}^{(k)} J \check{B}_2)^T + \check{M}^{(k)}$$

or, in the equivalent form

$$\check{Z}^{(k)} = (J \check{A} + J \check{B}_2 \check{F}_{\check{Z}^{(k)}})^T \check{Z}^{(k)} (J \check{A} + J \check{B}_2 \check{F}_{\check{Z}^{(k)}}) + (\check{F}_{\check{Z}^{(k)}})^T \check{R}_2^{(k)} \check{F}_{\check{Z}^{(k)}} + \check{M}^{(k)} \quad (7)$$

A matter of practical importance is how to solve the last equation. There are several approaches for this purpose, which are briefly outlined in the text that follows.

1. Block Successive Approximations. The method of Block Successive Approximations solves Equation (7). It is a modification of the method derived in [12] and applied in [13]. We take $\check{Y}^{(0)} = \text{diag}(2I, \dots, 2I)$ and use the following recurrence equation:

$$\begin{aligned} \check{Y}^{(i-1)} = \check{\mathbb{W}}(\check{Y}^{(i)}) &:= (J \check{A} + J \check{B}_2 \check{F}_{\check{Y}^{(i-1)}})^T \check{Y}^{(i-1)} (J \check{A} + J \check{B}_2 \check{F}_{\check{Y}^{(i-1)}}) \\ &+ (\check{F}_{\check{Y}^{(i-1)}})^T \check{R}_2^{(k)} \check{F}_{\check{Y}^{(i-1)}} + \check{M}^{(k)}, \quad i = 1, 2, \dots \end{aligned} \quad (8)$$

The iteration process stops with $\|\check{Y}^{(i_0)} - \check{\mathbb{W}}(\check{Y}^{(i_0)})\| \leq \text{eps}$.

2. A Block Stein Iteration. The second method for solving Equation (7) is a method of A Block Stein Iteration. We take $\check{Y}^{(0)} = \text{diag}(2I, \dots, 2I)$ and use the following recurrence equation:

$$\begin{aligned} \check{Y}^{(i)} = \check{\mathbb{W}}(\check{Y}^{(i)}) &:= (J \check{A} + J \check{B}_2 \check{F}_{\check{Y}^{(i-1)}})^T \check{Y}^{(i)} (J \check{A} + J \check{B}_2 \check{F}_{\check{Y}^{(i-1)}}) \\ &+ (\check{F}_{\check{Y}^{(i-1)}})^T \check{R}_2^{(k)} \check{F}_{\check{Y}^{(i-1)}} + \check{M}^{(k)}, \quad i = 1, 2, \dots \end{aligned} \quad (9)$$

The iteration process stops with $\|\check{Y}^{(i_0)} - \check{\mathbb{W}}(\check{Y}^{(i_0)})\| \leq \text{eps}$. We use MATLAB command dlyap for solving Equation (9). This approach is adapted in our investigation [13].

3. Successive Approximations: Equation by Equation. On the basis of performed experiments we introduce a new (third) method for solving Equation (6). It belongs to the class of successive approximations methods and we call it Equation by Equation. We apply this approach to Equation (1) with $Y^{(0)}(0) = 2I$. Therefore, the set of new iterative equations is: ($i = 1, 2, \dots$)

$$Y^{(i)}(\theta) = \mathbb{W}(Y^{(i-1)}(0)), \quad Y^{(i)}(\theta - 1) = \mathbb{W}(Y^{(i)}(\theta)), \quad \dots, \quad Y^{(i)}(0) = \mathbb{W}(Y^{(i)}(1)) \quad (10)$$

The iteration process stops with $\max_t \|\check{Y}^{(i_0)}(t) - \mathbb{W}(\check{Y}^{(i_0)}(t))\| \leq \text{eps}$.

In order to execute the iterative procedure Equation (6), the following algorithm is applied:

Algorithm TMS (Two Matrix Sequences)

1. We take $X^{(0)}(t) = 0, t \in \mathbb{Z}$, as well as small positive numbers ε and ϵps .
2. We compute $Z^{(0)}(\cdot)$ as θ -periodic and stabilizing solution of the Riccati equation:

$$Z^{(0)}(t) = A^T(t)Z^{(0)}(t+1)A(t) - \left[A^T(t)Z^{(0)}(t+1)B_2(t) + C^T(t)D_2(t) \right] + C^T(t)C(t) \\ \times \left[D_2^T(t)D_2(t) + B_2^T(t)Z^{(0)}(t+1)B_2(t) \right]^{-1} \left[A^T(t)Z^{(0)}(t+1)B_2(t) + C^T(t)D_2(t) \right]^T$$

3. For $k = 1, 2, 3, \dots$ we carry out the following steps:

3.1. We take $X^{(k)}(t) = Z^{(k-1)}(t) + X^{(k-1)}(t)$.

3.2. We compute $Z^{(k)}(\cdot)$ as the stabilizing solution of the set of Riccati equations with a defined sign Equation (6). This step might be executed with one of the following approaches (with the tolerance ϵps):

3.2.1. Block Successive Approximations presented via Equation (8).

3.2.2. A Block Stein Iteration presented via Equation (9).

3.2.3. Successive Approximations: Equation by Equation presented via Equation (10).

3.3. The algorithm stops if for some integer $k_0 > 0$, the inequality $\max_t \|X^{(k_0)}(t) - \mathcal{R}(t, X^{(k_0)})\| \leq \varepsilon$ holds.

4. The stabilizing solution is $\tilde{X}(t) = X^{(k_0)}(t), t = 0, \dots, \theta$.

It should be noted that the approaches involved in the algorithm via points in 3.2.1, 3.2.2 and 3.2.3 preserve the convergence properties of the iterative procedure derived in Theorem 3.1 proved in [7].

2.2. The Iterative Procedure with One Matrix Sequences

An alternative globally convergent iterative method for the computation of the unique stabilizing and θ -periodic solution $\tilde{X}(\cdot)$ of Equation (1) is derived in [8]. The method preserves similar desirable characteristics as global convergence and local quadratic rate of convergence. The main advantage of this method is that the matrix sequence of approximations of the stabilizing solution are computed directly without exploiting an additional sequence as it is done in the Algorithm TMS. The procedure is described in Section III in [8].

Compute $X^{(k)}(\cdot)$ as the stabilizing solution of the discrete-time Riccati equations with a defined sign [8]:

$$X^{(k)}(t) = \mathcal{Q}(X^{(k)}) := (\tilde{A}(t))^T X^{(k)}(t+1) \tilde{A}(t) - \left[(\tilde{A}(t))^T X^{(k)}(t+1) B_2(t) + L^{(k-1)}(t) \right] \\ \times \left[D_2^T(t) D_2(t) + B_2^T(t) X^{(k)}(t+1) B_2(t) \right]^{-1} \\ \times \left[(\tilde{A}(t))^T X^{(k)}(t+1) B_2(t) + L^{(k-1)}(t) \right]^T + M^{(k-1)}(t) \quad (11)$$

where

$$F_1^{(k-1)}(t) = (I_{m_1} \ 0) F^{(k)}(t), \\ = - (I_{m_1} \ 0) [R_\gamma(t) + B^T(t) X^{(k-1)}(t+1) B(t)]^{-1} [B^T(t) X^{(k-1)}(t+1) A(t) + D^T(t) C(t)] \\ \tilde{A}(t) = A(t) + B_1(t) F_1^{(k-1)}(t) \\ L^{(k-1)}(t) = (C(t) + D_1(t) F_1^{(k-1)}(t))^T D_2(t) \\ M^{(k-1)} = (C(t) + D_1(t) F_1^{(k-1)}(t))^T (C(t) + D_1(t) F_1^{(k-1)}(t)) - \gamma^2 (F_1^{(k-1)}(t))^T F_1^{(k-1)}(t)$$

We present the above set of Equations (11) as one equation with a block structure. We denote $\check{X} = \text{diag}(X(0), X(1), \dots, X(\theta))$ and J is a permutation matrix such that

$$J^{-1} \check{X} J = \text{diag}(X(1), X(2), \dots, X(\theta), X(0))$$

and

$$\begin{cases} \check{A} = \text{diag}(\check{A}(0), \check{A}(1), \dots, \check{A}(\theta)), & \check{B}_2 = \text{diag}(B_2(0), B_2(1), \dots, B_2(\theta)) \\ \check{D} = \text{diag}(D_2^T(0) D_2(0), D_2^T(1) D_2(1), \dots, D_2^T(\theta) D_2(\theta)) \\ \check{L} = \text{diag}(L^{(k-1)}(0), L^{(k-1)}(1), \dots, L^{(k-1)}(\theta)) \\ \check{M} = \text{diag}(M^{(k-1)}(0), M^{(k-1)}(1), \dots, M^{(k-1)}(\theta)) \\ \check{F}_{\check{X}} = -[\check{D} + \check{B}_2^T J^{-1} \check{X} J \check{B}_2]^{-1} (\check{A}^T J^{-1} \check{X} J \check{B}_2 + \check{L})^T \end{cases}$$

The set of equations $X^{(k)}(t) = \mathbb{Q}(X^{(k)})$ is transformed:

$$\check{X} = \check{A}^T J^{-1} \check{X} J \check{A} - (\check{A}^T J^{-1} \check{X} J \check{B}_2 + \check{L}) [\check{D} + \check{B}_2^T J^{-1} \check{X} J \check{B}_2]^{-1} (\check{A}^T J^{-1} \check{X} J \check{B}_2 + \check{L})^T + \check{M}$$

or

$$\check{X} = (J \check{A} + J \check{B}_2 \check{F}_{\check{X}})^T \check{X} (J \check{A} + J \check{B}_2 \check{F}_{\check{X}}) + (\check{F}_{\check{X}})^T \check{D} \check{F}_{\check{X}} + \check{M} \quad (12)$$

We present two iterative approaches to implement iteration Equation (12).

1. A Block Stein Iteration. The first method for solving Equation (12) is a method of A Block Stein Iteration. We take $\check{Y}_0 = \text{diag}(2I, \dots, 2I)$ and use the following recurrence equation:

$$\check{Y}^{(i)} = (J \check{A} + J \check{B}_2 \check{F}_{\check{Y}^{(i-1)}})^T \check{Y}^{(i)} (J \check{A} + J \check{B}_2 \check{F}_{\check{Y}^{(i-1)}}) + (\check{F}_{\check{Y}^{(i-1)}})^T \check{D} \check{F}_{\check{Y}^{(i-1)}} + \check{M}^{(k)} \quad (13)$$

$i = 1, 2, \dots$ The iteration process stops with $\|\check{Y}^{(i_0)} - \mathbb{Q}(\check{Y}^{(i_0)})\| \leq \text{eps}$. We use the MATLAB command `dlyap` for solving Equation (13).

2. Successive Approximations: Equation by Equation. The second method for the solution of Equation (11) is a method of Successive Approximations: Equation by Equation. We apply this approach to Equation (11) with $Y^{(0)}(0) = I$. Our iteration is: ($i = 1, 2, \dots$)

$$Y^{(i)}(\theta) = \mathbb{Q}(Y^{(i-1)}(0)), \quad Y^{(i)}(\theta - 1) = \mathbb{Q}(Y^{(i)}(\theta)), \quad \dots, \quad Y^{(i)}(0) = \mathbb{Q}(Y^{(i)}(1)) \quad (14)$$

The iteration process stops with $\max_t \|\check{Y}^{(i_0)}(t) - \mathbb{Q}(\check{Y}^{(i_0)}(t))\| \leq \text{eps}$.

The following algorithm is applied so as to execute the iterative procedure described by Equation (11):

Algorithm OMS (One Matrix Sequence).

1. We take $X^{(0)}(t) = 0, t \in \mathbb{Z}, \varepsilon$ as a small positive number.
2. We compute $X^{(1)}(\cdot)$ as θ -periodic and stabilizing solution of the following Riccati equation:

$$\begin{aligned} X^{(1)}(t) = & A^T(t) X^{(1)}(t+1) A(t) - \left[A^T(t) X^{(1)}(t+1) B_2(t) + C^T(t) D_2(t) \right] + C^T(t) C(t) \\ & \times \left[D_2^T(t) D_2(t) + B_2^T(t) X^{(1)}(t+1) B_2(t) \right]^{-1} \left[A^T(t) X^{(1)}(t+1) B_2(t) + C^T(t) D_2(t) \right]^T \end{aligned}$$

3. For $k = 2, 3, 4, \dots$ we compute the stabilizing solution $X^{(k)}(\cdot)$ of the discrete-time Riccati equations with a defined sign Equation (11). This step might be performed with one of the following approaches:

3.1. A Block Stein Iteration presented via Equation (13).

3.2. Successive Approximations: Equation by Equation presented via Equation (14).

3.3. The algorithm stops if for some integer $k_0 > 0$, the inequality $\max_t \|X^{(k_0)}(t) - \mathcal{R}(t, X^{(k_0)})\| \leq \varepsilon$ holds.

4. The stabilizing solution is $\tilde{X}(t) = X^{(k_0)}(t), t = 0, \dots, \theta$.

We should note that the approaches involved in the algorithm via points in 2 and 3 preserve the convergence properties of the iterative procedure derived in Theorem 3.2 derived in [8].

3. Numerical Experiments

In this section we present the performance of the latter two algorithms when finding the stabilizing solution to a set of periodic discrete-time Riccati equations with indefinite sign in the quadratic part Equation (1). Our experiments are executed in MATLAB on a 2.16 GHz Intel(R) Duo CPU computer. We remark the fact, the computer realization of the block methods and the method of Successive Approximations: Equation by Equation needs $O(n^2(\theta + 1)^2)$ and $O((\theta + 1)n^2)$ computer memory cells, respectively.

3.1. Experiment I

Consider a discrete-time periodic linear system described by

$$A(0) = \begin{pmatrix} -1.2 & 0 & 0.4 \\ 4 & 2 & 0 \\ 0.12 & -8 & 1.6 \end{pmatrix}, A(1) = \begin{pmatrix} -2 & 2 & 0 \\ 4.2 & 6.4 & 0 \\ 1.2 & -10 & 2.8 \end{pmatrix}, A(2) = \begin{pmatrix} -1.2 & 0.8 & 0 \\ 8 & 0.3 & 0.5 \\ 1.2 & -8 & 4.5 \end{pmatrix}$$

$$B_1(0) = \begin{pmatrix} 1.8048 & 0.5877 \\ -0.3916 & -0.0175 \\ 1.2778 & 0.5082 \end{pmatrix}, B_1(1) = \begin{pmatrix} -0.4959 & 0.1848 \\ 2.4629 & -1.4624 \\ -0.9885 & 1.3702 \end{pmatrix}, B_1(2) = \begin{pmatrix} -1.0077 & -0.1932 \\ -0.4189 & 1.77 \\ 0.4144 & -1.1287 \end{pmatrix}$$

$$B_2(0) = \begin{pmatrix} 0.4089 & -1.0077 \\ -0.1932 & -0.4189 \\ 0.4144 & -1.1287 \end{pmatrix}, B_2(1) = \begin{pmatrix} -0.7202 & 0.5488 \\ 0.5385 & 0.1610 \\ -0.4652 & -1.0513 \end{pmatrix}, B_2(2) = \begin{pmatrix} -2.107 & -0.604 \\ -0.4065 & -0.075 \\ -0.109 & -0.807 \end{pmatrix}$$

$$C(0) = \begin{pmatrix} 0.233 & 1.5 & 0.3 \\ 0.2 & 1.5 & 0.28 \end{pmatrix}, C(1) = \begin{pmatrix} 0.72 & 2.0 & 1.5 \\ 0.3 & 1.5 & 0.3 \end{pmatrix}, C(2) = \begin{pmatrix} 1.5 & 0.06 & 2.5 \\ 1.5 & 1.2 & 0.85 \end{pmatrix}$$

$$D_1(0) = \begin{pmatrix} 0.35 & -0.45 \\ 0.28 & -0.13 \end{pmatrix}, D_1(1) = \begin{pmatrix} 0.1 & 0.4 \\ 0.95 & 0.48 \end{pmatrix}, D_1(2) = \begin{pmatrix} 7.07 & 2.2 \\ 1.22 & -0.95 \end{pmatrix}$$

$$D_2(0) = \begin{pmatrix} 0.15 & 0.75 \\ -0.18 & -0.25 \end{pmatrix}, D_2(1) = \begin{pmatrix} 2 & -1.85 \\ 1.46 & -0.98 \end{pmatrix}, D_2(2) = \begin{pmatrix} 3.4 & 4.82 \\ 3.28 & -0.35 \end{pmatrix}$$

with $\theta = 2$, $t = 0, 1, 2$, $\gamma = 6.5$, $n = 3$. For all of the discussed methods are executed consequently 100 and 200 runs with the same matrix coefficients. We use $tol = 10^{-5}$ (the number tol is the computational tolerance for the first sequence $\{X^{(k)}(t)\}_{k=0}^{\infty}$, see [7]) and $eps = 10^{-6}$ for all executions. The results could be summarized as follows.

Algorithm TMS with Block Successive Approximations Equation (8) requires 3 main iterations and it computes the stabilizing solution with $\max_t \|\tilde{X}^{(3)}(t) - \mathcal{R}(t, \tilde{X}^{(3)})\| = 8.1904 \times 10^{-8} < tol$. CPU time for execution of the algorithm is 0.4 s for 100 runs and 0.97 s for 200 runs.

Algorithm TMS with Block Stein Iteration Equation (9) requires 3 main iterations and it computes the stabilizing solution with $\max_t \|\tilde{X}^{(3)}(t) - \mathcal{R}(t, \tilde{X}^{(3)})\| = 1.0794 \times 10^{-9} < \text{tol}$. CPU time for execution of the algorithm is 0.78 s for 100 runs and 1.57 s for 200 runs.

Algorithm TMS with Successive Approximations Equation by Equation (10) requires 3 main iterations and it computes the stabilizing solution with $\max_t \|\tilde{X}^{(3)}(t) - \mathcal{R}(t, \tilde{X}^{(3)})\| = 4.0976 \times 10^{-6} < \text{tol}$. CPU time for execution of iteration Equation (10) is 0.3750 s for 100 runs and 0.75 s for 200 runs.

Algorithm OMS with Block Stein Iteration Equation (13) requires 3 main iterations and it computes the stabilizing solution with $\|\tilde{X}^{(3)} - \mathcal{Q}(\tilde{X}^{(3)})\| = 5.8077 \times 10^{-8}$. CPU time for execution of iteration Equation (13) is 1.15 s for 100 runs and 2.41 s for 200 runs.

Algorithm OMS with Method of Successive Approximations Equation by Equation (14) requires 3 main iterations and it computes the stabilizing solution with $\|\tilde{X}^{(3)} - \mathcal{Q}(\tilde{X}^{(3)})\| = 2.5060 \times 10^{-8}$. CPU time for execution of iteration Equation (14) is 0.5150 s for 100 runs and 1.14 s for 200 runs.

3.2. Experiment II

We take $\theta = 2$, $t = 0, 1, 2$, and $n = 5, m_1 = m_2 = 2$, $\gamma = 7.5$. The matrix coefficients are described below:

$$A(0) = 1.8 \times \begin{pmatrix} -0.799 & -0.01 & -0.0326 & 0.1441 & -0.3071 \\ -0.1931 & -0.419 & -0.4564 & -0.2086 & -0.1385 \\ -0.1015 & 0.3199 & -0.37 & -0.1572 & -0.1701 \\ -0.1383 & 0.0281 & -0.1879 & 0.0640 & 0.18 \\ -0.3382 & 0.1764 & -0.1084 & -0.14 & 0.0555 \end{pmatrix}$$

$$A(1) = 1.9 \times \begin{pmatrix} 0.182 & -0.4412 & -0.5427 & 0.4532 & 0.1874 \\ -0.1532 & 0.0329 & 0.475 & 0.097 & -0.0548 \\ 0.7056 & -0.1355 & 0.6479 & -0.7397 & 0.3912 \\ 0.0364 & -0.4506 & 0.2467 & 0.0482 & -0.4428 \\ -0.3066 & -0.1566 & -0.1077 & 0.0262 & 0.82 \end{pmatrix}$$

$$A(2) = 1.7 \times \begin{pmatrix} 0.8507 & 0.0254 & -0.0743 & -0.2159 & 0.009 \\ -0.0823 & 0.5286 & -0.9114 & -0.4291 & -0.3859 \\ 0.0960 & 0.3086 & 0.20 & -0.1370 & 0.3882 \\ -0.2909 & 0.7105 & 0.0487 & 0.7335 & 0.2621 \\ -0.3103 & -0.1086 & 0.0126 & -0.3097 & 0.034 \end{pmatrix}$$

$$B_1(0) = \text{full}(\text{sprand}(5, 2, 0.6))/7; \quad B_2(0) = \frac{1}{3} \begin{pmatrix} 0.4089 & -1.0077 \\ -0.1932 & -0.4189 \\ 0.4144 & -1.1287 \\ 0.19 & -0.41 \\ 1.24 & 0.86 \end{pmatrix}$$

$$B_1(1) = \text{full}(\text{sprand}(5, 2, 0.6))/7; \quad B_2(1) = \frac{1}{3} \begin{pmatrix} -0.7202 & 0.5488 \\ 0.5385 & 0.1610 \\ -0.4652 & -1.0513 \\ 2.4629 & -1.4624 \\ -0.9885 & 1.37 \end{pmatrix}$$

$$B_1(2) = \text{full}(\text{sprand}(5, 2, 0.6)) / 7; \quad B_2(2) = \frac{1}{3} \begin{pmatrix} -2.107 & -0.604 \\ -0.4065 & -0.075 \\ -0.109 & -0.807 \\ 0.5877 & -0.3916 \\ -0.0175 & 1.2778 \end{pmatrix}$$

$$C(0) = 0.9 \begin{pmatrix} 0.233 & 1.5 & 0.3 & 1.5 & 0.3 \\ 1.5 & 0.2 & 1.5 & 0.28 & 0.15 \end{pmatrix}, \quad C(1) = 0.9 \begin{pmatrix} 0.72 & 2.0 & 1.5 & 0.3 & 1.25 \\ 0.3 & 0.06 & 2.5 & 1.04 & 1.15 \end{pmatrix}$$

$$C(2) = 0.9 \begin{pmatrix} 1.5 & 0.06 & 2.5 & 1.04 & 1.15 \\ 1.2 & 0.85 & 0.06 & 2.5 & 1.04 \end{pmatrix}$$

$$D_1(0) = \text{randn}(2, 2); \quad D_1(1) = \text{randn}(2, 2); \quad D_1(2) = \text{randn}(2, 2);$$

$$D_2(0) = \begin{pmatrix} 0.7615 & 0.4508 \\ -0.6894 & -1.565 \end{pmatrix}, \quad D_2(1) = \begin{pmatrix} -0.0630 & -0.5223 \\ -0.7534 & 0.226 \end{pmatrix}, \quad D_2(2) = \begin{pmatrix} -1.0126 & -1.3643 \\ -0.8901 & -2.0057 \end{pmatrix}$$

Two experimental trials are carried out. These are Test 1 and Test 2, where Test 1 consists of 100 examples and Test 2 of 200 examples. The matrix coefficients $B_1(0), B_1(1), B_1(2)$ and $D_1(0), D_1(1), D_1(2)$ differ for all of the examples as the random Matlab functions *sprand* and *randn* are used to define them. We execute Tests 1 and 2 successively.

Our results might be summarized as follows.

Block Successive Approximations Equation (8) with $\text{tol} = 10^{-7}$ and $\text{eps} = 10^{-7}$ requires 2 main iterations and it computes the stabilizing solution with $\|\tilde{X}^{(2)} - \mathbb{W}(\tilde{X}^{(2)})\| = 5.2147 \times 10^{-8}$. CPU time for execution of iteration Equation (8) is 2.5 s for Test 1 and 5.1 s for Test 2.

Block Stein Iteration Equation (9) with $\text{tol} = 10^{-6}$ and $\text{eps} = 10^{-6}$ requires 2 main iterations and it computes the stabilizing solution with $\|\tilde{X}^{(2)} - \mathbb{W}(\tilde{X}^{(2)})\| = 1.8306 \times 10^{-10}$. CPU time for execution of iteration Equation (9) is 1.8 s for Test 1 and 3.7 s for Test 2.

Method of Successive Approximations Equation by Equation (10) with $\text{tol} = 10^{-7}$ and $\text{eps} = 10^{-7}$ requires 2 main iterations and it computes the stabilizing solution with $\|\tilde{X}^{(2)} - \mathbb{W}(\tilde{X}^{(2)})\| = 5.7561 \times 10^{-9}$. CPU time for execution of iteration Equation (10) is 1.5 s for Test 1 and 3.1 s for Test 2.

Block Stein Iteration Equation (13) with $\text{tol} = 10^{-6}$ and $\text{eps} = 10^{-7}$ requires 2 main iterations and it computes the stabilizing solution with $\|\tilde{X}^{(2)} - \mathbb{Q}(\tilde{X}^{(2)})\| = 1.2059 \times 10^{-8}$. CPU time for execution of iteration Equation (13) is 2.9 s for Test 1 and 5.1 s for Test 2.

Method of Successive Approximations Equation by Equation (14) with $\text{tol} = 10^{-7}$ and $\text{eps} = 10^{-7}$ requires 2 main iterations and it computes the stabilizing solution with $\|\tilde{X}^{(2)} - \mathbb{Q}(\tilde{X}^{(2)})\| = 5.2861 \times 10^{-8}$. CPU time for execution of iteration Equation (14) is 2.1 s for Test 1 and 4.2 s for Test 2.

4. Conclusions

On basis of the performed computational experiments the following conclusions might be outlined. First of all, Block Stein iterations Equations (9) and (13) achieve highest accuracy. Nevertheless, this method uses block “equation” structure, *i.e.*, it works with block matrix coefficients, hence, the utilized RAM memory is greater than that required by the method of the Successive Approximations: Equation by Equation. Secondly, the method of Successive Approximations: Equation by Equation is faster than rest of the methods and at the same time it uses less RAM memory.

Acknowledgments: The present research paper was supported in a part by the EEA Scholarship Programme BG09 Project Grant D03-91 under the European Economic Area Financial Mechanism. This support is greatly appreciated.

Author Contributions: The contributions of all of the authors have been similar. All of them have worked together to develop the present manuscript.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Pastor, A.; Hernandez, V. Differential periodic Riccati equations existence and uniqueness of nonnegative definite solutions. *Math. Control Signal Syst.* **1993**, *6*, 341–362.
2. Abou-Kandil, H.; Freiling, G.; Ionescu, V.; Jank, G. *Matrix Riccati Equations in Control Systems Theory*; Birhauser: Basel, Switzerland, 2003.
3. Bittanti, S.; Colaneri, P. *Periodic Systems, Filtering and Control*; Springer-Verlag: London, UK, 2009.
4. Halanay, A.; Rasvan, V.I. *Stability and Stable Oscillations in Discrete Time Systems*; Gordon and Breach: Amsterdam, The Netherlands, 2000.
5. Varga, A. On solving periodic differential matrix equations with applications to periodic system norms computation. In Proceedings of the 44th IEEE Conference on Decision and Control, CDC-ECC '05, Seville, Spain, 12–15 December 2005.
6. Varga, A. On solving periodic Riccati equations. *Numer. Linear Algebra Appl.* **2008**, *15*, 809–835.
7. Dragan, V.; Aberkane, S.; Ivanov, I.G. On computing the stabilizing solution of a class of discrete-time periodic Riccati equations. *Int. J. Robust Nonlinear Control* **2015**, *25*, 1066–1093.
8. Dragan, V.; Aberkane, S.; Ivanov, I.G. An iterative procedure for computing the stabilizing solution of discrete-time periodic Riccati equations with an indefinite sign. In Proceedings of the 21st International Symposium on Mathematical Theory of Networks and Systems, MTNS 2014, Groningen, The Netherlands, 7–11 July 2014.
9. Dragan, V.; Ivanov, I.G. A numerical procedure to compute the stabilizing solution of game theoretic Riccati equations of stochastic control. *Int. J. Control* **2011**, *84*, 783–800.
10. Dragan, V.; Ivanov, I.G. Computation of the stabilizing solution of game theoretic Riccati equation arising in stochastic H_∞ control problems. *Numer. Algorithms* **2011**, *57*, 357–375.
11. Feng, Y.; Anderson, B.D.O. An iterative algorithm to solve state-perturbed stochastic algebraic Riccati equations in LQ zero-sum games. *Syst. Control Lett.* **2010**, *59*, 50–56.
12. Dragan, V.; Morozan, T.; Stoica, A.M. Iterative algorithm to compute the maximal and stabilizing solutions of a general class of discrete-time Riccati-type equations. *Int. J. Control* **2010**, *83*, 837–847.
13. Ivanov, I.G. An Improved Method for Solving a System of Discrete-Time Generalized Riccati Equations. *J. Numer. Math. Stoch.* **2011**, *3*, 57–70.



© 2016 by the authors; licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons by Attribution (CC-BY) license (<http://creativecommons.org/licenses/by/4.0/>).