

Article

A Successive over Relaxation Implicit Iterative Algorithm for Solving Stochastic Linear Systems with Markov Jumps

Tianrui Wu, Peiqi Huang * and Hong Chen

School of Science, Nanjing Forestry University, Nanjing 210037, China; wutianrui0930@njfu.edu.cn (T.W.); hongchen_lucky@njfu.edu.cn (H.C.)

* Correspondence: pqhuang@njfu.edu.cn

Abstract: In order to solve continuous stochastic Lyapunov equations, a novel implicit iterative algorithm is presented by means of successive over relaxation (SOR) iteration in this article. Throughout this method, three tuning parameters are added for the improvement of the convergence rate. It is shown that this algorithm is monotonically bounded, and the convergence condition is also given and extended. Applying the latest updated estimates, this algorithm can attain a better convergence performance compared with other existing iterative algorithms when choosing appropriate tuning parameters. Finally, a numerical example is provided to illustrate the feasibility and priority of this approach.

Keywords: iterative algorithms; SOR; continuous stochastic Lyapunov equations; Markov jumps

MSC: 65F45; 93E20



Citation: Wu, T.; Huang, P.; Chen, H. A Successive over Relaxation Implicit Iterative Algorithm for Solving Stochastic Linear Systems with Markov Jumps. *Mathematics* **2024**, *12*, 1080. <https://doi.org/10.3390/math12071080>

Academic Editor: Alicia Cordero

Received: 10 March 2024

Revised: 29 March 2024

Accepted: 2 April 2024

Published: 3 April 2024



Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

In the design of control systems, various matrix equations are widely used [1–6]. Specifically, some coupled matrix equations are important tools for studying the stability of Markov jump linear systems [7]. It can be seen in [8] that the mean square stability of an Itô system can be characterized by the corresponding Lyapunov-type matrix equation. Markov jump systems are useful tools that have been instrumental in modeling certain real-world systems susceptible to random factors. This concept, which is highlighted by reference [9], has garnered significant interest among researchers within this discipline. By employing Markov jumps, scientists and engineers can gain insight into how various components of these complex systems might behave under conditions of uncertainty or change, providing valuable tools for predicting outcomes and making informed decisions. In [10], the notation of exact detectability is presented for linear systems with state-multiplicative noise. Based on critical stability and exact detectability, some useful properties are derived in [10] for Itô Lyapunov matrix equations. Uniform definitions are given for the notions of observable and detectable properties in the realm of continuous-time stochastic systems reported in [11], where the so-called stochastic Lyapunov matrix equations are reviewed. In [12], some stochastic stability properties are presented for the Markov jump linear system owing to the existence of the solution of coupled Lyapunov matrix equations (CLMEs). In the study referenced in [13], an approach is developed to design a suboptimal controller, denoted as H_2/H_∞ , specifically for continuous-time linear Itô stochastic systems with Markov jumps (MJIS systems). The extended concept of exact detectability in the literature [14] has been incorporated into the linear MJIS systems, making it more convenient to analyze the properties of control systems. Research has suggested that under exact detectability conditions, the mean square stability of a linear MJIS system depends on whether the corresponding Lyapunov matrix equations have positive semi-definite solutions. Two iterative algorithms are established for both continuous-time and discrete-time MJIS systems in [15]. And their convergence are studied using linear positive operators. According to the characteristic of

the solution of corresponding Lyapunov equations, a necessary and sufficient condition for the mean square stability of detectable systems of this type is developed in [16]. In fact, the CLMEs, characterized by their intricate coupling dynamics and appearance in MJIS systems, can indeed be considered as a peculiar example of generalized ones.

For the above case, ref. [17] proposes a novel implicit iterative approach specifically tailored to the solution of the CLMEs in discrete-time situations. Ref. [18] proposes a method to tackle the complex challenge of solving coupled equations, through minimizing a quadratic function embedded within the iterative process. In addition, ref. [19] provides a direct method for the coupled matrix equations, which is less effective for large-scale systems. In recent years, various kinds of iterative methods have been put forward by different researchers. Ref. [20] presents a new method of least-square iteration to deal with equations of this kind, in which a hierarchical identification principle is employed. In [21], a full-rank gradient-based algorithm, as well as a reduced-rank one, is constructed. In fact, due to the similarity of coupled and general Lyapunov matrix equations, both can be solved by applying the iterative algorithms. Moreover, all of the above algorithms use the latest estimated value generated by the last iteration process.

In 2016, an implicit iterative algorithm [22] was put forward to solve continuous CLMEs by using some adjustable parameters. The successive over relaxation (SOR) technique is known to be a classic way to increase the convergence rate of Jacobi iterations in [23]. Due to this advantage, it was used to solve the discrete periodic Lyapunov matrix equations [24]. In [25], a novel iterative algorithm is proposed to deal with coupled Lyapunov equations arising in continuous-time Itô stochastic systems with Markov jumps, in which two adjustment parameters are introduced. And it combines both the latest and the former step information to estimate the unknown matrices. Ref. [26] establishes a new iterative method for solving the CLMEs associated with continuous-time Markov jump linear systems by the SOR technique. A relaxation parameter is included, which can be selected appropriately to improve the convergence performance.

Inspired by the aforementioned facts, an SOR implicit iterative algorithm is established in this paper for solving continuous stochastic Lyapunov equations. And three tunable parameters are included to improve the convergence speed of the algorithm. Subsequently, the monotonicity and boundedness of the proposed algorithm are analyzed, and the convergence condition is further investigated. With the up-to-date knowledge of the iteration value, this approach exhibits better convergence than others by choosing the tuning parameters appropriately.

In this paper, we use $\mathbb{R}^{m \times n}$ to represent the set of all the $m \times n$ -dimensional real matrices. $E(\cdot)$ denotes the mathematical expectation. And A^T , A^{-1} represent the transpose and inverse matrices of A , respectively. $\rho(A)$ is written as the spectral radius of A . The notation $\mathbb{I}[a, b]$ denotes a discrete finite set $\{a, a + 1, \dots, b\}$, in which both a and b are integers and $a < b$. The vectorization of a matrix $A = [a_1, a_2, \dots, a_n] \in \mathbb{R}^{n \times n}$ is defined as $\text{vec}(A) = [a_1^T, a_2^T, \dots, a_n^T]^T$. $M \otimes W$ means the Kronecker product of matrices M and W . If P is a positive definite matrix, we record it as $P > 0$. Similarly, $(P_1(m), P_2(m), \dots, P_N(m)) > 0$ indicates that in this sequence, $P_i(m) > 0$, $i \in \mathbb{I}[1, N]$. $\|\cdot\|_F$ is used to represent Frobenius norm.

2. Problem Formulation and Preliminaries

There exists a continuous-time Itô stochastic system with Markov jumps:

$$dx(t) = A_0(\theta(t))x(t)dt + \sum_{s=1}^r A_s(\theta(t))x(t)d\omega_s(t), \quad (1)$$

where $x(t) \in \mathbb{R}^n$ is the system state; $A_0(\theta(t))$, $A_s(\theta(t))$ are both the coefficient matrices; $\omega_s(t) \in \mathbb{R}$, $s \in \mathbb{I}[1, r]$ are real random processes, which are defined on a complete probability space $\{\Omega, \mathcal{F}, \mathcal{P}\}$ with $E(\omega_i(t)) = 0$ and $E(\omega_i(t)\omega_j(s)) = \delta_{ij}$, $i, j \in \mathbb{I}[1, r]$. $\{\theta(t), t \geq 0\}$ is a continuous-time discrete-state Markov process. The transition rate matrix is $\Pi = [\pi_{ij}]_{N \times N}$,

while $\pi_{ij} \geq 0$ for $j \neq i$, and $\sum_{j=1}^N \pi_{ij} = 0, i \in \mathbb{I}[1, N]$. The initial values are set to be $x(0) = x_0$ and $\theta(0) = \theta_0$. Next, the definition of asymptotically mean square stability (AMSS) is introduced for system (1).

Definition 1 ([15]). *System (1) has AMSS if for any $x_0 \in \mathbb{R}^n$ and $\theta_0 \in \mathbb{I}[1, N]$, there exists*

$$\lim_{t \rightarrow \infty} E(\|x(t, x_0, \theta_0)\|^2 | x_0, \theta_0) = 0.$$

The corresponding continuous stochastic Lyapunov equation of system (1) can be written as follows:

$$A_{0i}^T P_i + P_i A_{0i} + \sum_{s=1}^r A_{si}^T P_i A_{si} + \sum_{j=1}^N \pi_{ij} P_j = -Q_i, \quad i \in \mathbb{I}[1, N]. \tag{2}$$

In this equation, $P_i, i \in \mathbb{I}[1, N]$ are unknown solutions of Lyapunov Equation (2) and $Q_i > 0, i \in \mathbb{I}[1, N]$, are known matrices. This equation is usually used to analyze the stability of system (1). Regarding the asymptotically mean square stability of (1), the following result is obtained.

Lemma 1 ([15]). *System (1) has AMSS if and only if there is a unique solution $(P_1, P_2, \dots, P_N) > 0$ of the corresponding Lyapunov Equation (2), for any given $Q_i > 0$.*

Thus, the stability of the system (1) is equivalent to finding a unique solution of Equation (2). While Lyapunov Equation (2) can be changed into the following format:

$$A_i^T P_i + P_i A_i = - \sum_{s=1}^r A_{si}^T P_i A_{si} - \sum_{j=1, j \neq i}^N \pi_{ij} P_j - \beta_i P_i - Q_i, \tag{3}$$

with

$$A_i = A_{0i} + 0.5\pi_{ii}I_n - 0.5\beta_i I_n, \quad \beta_i \geq 0, i \in \mathbb{I}[1, N]. \tag{4}$$

Through Lemma 1 and the associated theories of stability, we can obtain the following conclusions:

Lemma 2 ([25]). *Any matrix $A_i, i \in \mathbb{I}[1, N]$ is Hurwitz-stable if system (1) has AMSS.*

An iterative algorithm in the implicit form is proposed in [25] to deal with the coupled Lyapunov Equation (2). It is summarized as follows:

Theorem 1 ([25]). *Assume the Markov jump system (1) has AMSS. If the following algorithm (5) satisfies $Q_i > 0, \beta_i \geq 0$, and the tuning parameters $\alpha_j \in [0, 1]$, then the sequence $(P_1(m), P_2(m), \dots, P_N(m))$ generated by the algorithm (5) with zero initial conditions $P_i(0) = 0, i \in \mathbb{I}[1, N]$ converges to the unique positive definite solution of the continuous coupled Lyapunov matrix Equation (2). The following is the designed iteration form:*

$$\begin{aligned} A_i^T P_i(m+1) + P_i(m+1)A_i = & - \sum_{s=1}^r A_{si}^T P_i(m)A_{si} - \sum_{j=1}^{i-1} \pi_{ij} [\alpha_j P_j(m+1) \\ & + (1 - \alpha_j)P_j(m)] - \sum_{j=i+1}^N \pi_{ij} P_j(m) \\ & - \beta_i P_i(m) - Q_i, \quad i \in \mathbb{I}[1, N]. \end{aligned} \tag{5}$$

3. Main Result

Enlightened by the idea of SOR technique, in what follows, we plan to find a new algorithm to obtain the solution of Equation (2). Since we have

$$A_i^T P_i + P_i A_i = (1 - \gamma)(A_i^T P_i + P_i A_i) + \gamma(A_i^T P_i + P_i A_i), \quad i \in \mathbb{I}[1, N], \tag{6}$$

and γ is a tunable parameter. From the preceding section, we rewrite Equation (2) as (3) by substituting the first term on the right-hand side of (6) with the right-hand item of (3); we obtain the following relation:

$$A_i^T P_i + P_i A_i = (1 - \gamma) \left(- \sum_{s=1}^r A_{si}^T P_i A_{si} - \sum_{j=1}^{i-1} \pi_{ij} [\alpha_j P_j + (1 - \alpha_j) P_j] - \sum_{j=i+1}^N \pi_{ij} P_j - \beta_i P_i - Q_i \right) + \gamma(A_i^T P_i + P_i A_i), \quad i \in \mathbb{I}[1, N]. \tag{7}$$

Utilizing the latest updated estimation, we bring up an implicit iterative algorithm below to solve Equation (2):

$$\begin{aligned} &A_i^T P_i(m+1) + P_i(m+1) A_i \\ &= (1 - \gamma) \left(- \sum_{s=1}^r A_{si}^T P_i(m) A_{si} - \sum_{j=1}^{i-1} \pi_{ij} [\alpha_j P_j(m+1) + (1 - \alpha_j) P_j(m)] - \sum_{j=i+1}^N \pi_{ij} P_j(m) - \beta_i P_i(m) - Q_i \right) \\ &\quad + \gamma(A_i^T P_i(m) + P_i(m) A_i), \quad i \in \mathbb{I}[1, N]. \end{aligned} \tag{8}$$

Remark 1. If the parameter $\gamma = 0$, then the proposed algorithm (8) is reduced to algorithm (5).

Remark 2. For every iteration step of algorithm (8), N standard continuous Lyapunov matrix equations need to be solved. The algorithm proposed in this paper is an implicit form.

Remark 3. It can be seen that the algorithm (8) proposed in this article is very similar to the SOR iteration method in ordinary linear equations. Therefore, in the following text, algorithm (8) is abbreviated as the SOR implicit iterative algorithm.

First, the following lemma illustrates that the sequence generated by (8) is bounded.

Lemma 3. Assume system (1) has AMSS, and $Q_i > 0, i \in \mathbb{I}[1, N]$. If $\beta_i \geq 0$, and the tuning parameters $\alpha_j \in [0, 1]$, then on the condition of zero initial values, there is an upper bound of the sequence $(P_1(m), P_2(m), \dots, P_N(m))$ generated by algorithm (8) when $0 \leq \gamma < 1$. In other words, for any integer $m \geq 0$, we have

$$P_i(m) < P_i, \quad i \in \mathbb{I}[1, N]. \tag{9}$$

Proof. According to Lemmas 1 and 2, since the matrices $A_i, i \in \mathbb{I}[1, N]$ are Hurwitz-stable and Equation (2) has a unique solution (P_1, P_2, \dots, P_N) , we can subtract (8) from (7) to obtain

$$A_i^T [P_i - P_i(m+1) - \gamma(P_i - P_i(m))] + [P_i - P_i(m+1) - \gamma(P_i - P_i(m))] A_i = V_i(m), \tag{10}$$

and

$$\begin{aligned}
 V_i(m) = & (1 - \gamma) \left(- \sum_{s=1}^r A_{si}^T (P_i - P_i(m)) A_{si} - \sum_{j=1}^{i-1} \pi_{ij} [\alpha_j (P_j - P_j(m+1)) \right. \\
 & + (1 - \alpha_j) (P_j - P_j(m))] - \sum_{j=i+1}^N \pi_{ij} (P_j - P_j(m)) \\
 & \left. - \beta_i (P_i - P_i(m)) \right). \tag{11}
 \end{aligned}$$

In the following text, the proof of this theorem is mainly based on mathematical induction. Because of the zero initial value condition of the algorithm mentioned before, apparently, (9) holds for $m = 0$. Now, we suppose when $m = k$ and $k \geq 0$, (9) holds. This means

$$P_i(k) < P_i, \quad i \in \mathbb{I}[1, N]. \tag{12}$$

Let $m = k, i = 1$ in relation (11); we can obtain

$$\begin{aligned}
 V_1(k) = & (1 - \gamma) \left(- \sum_{s=1}^r A_{s1}^T (P_1 - P_1(k)) A_{s1} - \sum_{j=2}^N \pi_{1j} (P_j - P_j(k)) \right. \\
 & \left. - \beta_1 (P_1 - P_1(k)) \right). \tag{13}
 \end{aligned}$$

Due to the assumptions set by mathematical induction above and $0 \leq \gamma < 1$, we can obtain $V_1(k) < 0$. Therefore, from (10), we know that

$$A_1^T [P_1 - P_1(k+1) - \gamma(P_1 - P_1(k))] + [P_1 - P_1(k+1) - \gamma(P_1 - P_1(k))] A_1 < 0. \tag{14}$$

A_1 is Hurwitz-stable, so we can obtain the following inequality from Theorem 3.1 in [27]:

$$P_1 - P_1(k+1) - \gamma(P_1 - P_1(k)) > 0.$$

That is,

$$P_1 - P_1(k+1) > \gamma(P_1 - P_1(k)).$$

Since $P_1 > P_1(k)$ from assumption (12), we have

$$P_1 > P_1(k+1). \tag{15}$$

Using mathematical induction again, we now assume that

$$P_j > P_j(k+1), \quad j \in \mathbb{I}[1, l-1], l \geq 2. \tag{16}$$

Let $m = k, i = l$ in Equations (10) and (11); we can obtain

$$A_l^T [P_l - P_l(k+1) - \gamma(P_l - P_l(k))] + [P_l - P_l(k+1) - \gamma(P_l - P_l(k))] A_l = V_l(k),$$

and

$$\begin{aligned}
 V_l(k) = & (1 - \gamma) \left(- \sum_{s=1}^r A_{sl}^T (P_l - P_l(k)) A_{sl} - \sum_{j=1}^{l-1} \pi_{lj} [\alpha_j (P_j - P_j(k+1)) \right. \\
 & + (1 - \alpha_j) (P_j - P_j(k))] - \sum_{j=l+1}^N \pi_{lj} (P_j - P_j(k)) - \beta_l (P_l - P_l(k)) \left. \right).
 \end{aligned}$$

For $0 \leq \gamma < 1$; from assumptions (12) and (16), we can obtain $V_l(k) < 0$. That is,

$$A_l^T [P_l - P_l(k+1) - \gamma(P_l - P_l(k))] + [P_l - P_l(k+1) - \gamma(P_l - P_l(k))] A_l < 0.$$

Due to the Hurwitz stability of A_l , the following inequality holds:

$$P_l - P_l(k + 1) - \gamma(P_l - P_l(k)) > 0.$$

It means that

$$P_l - P_l(k + 1) > \gamma(P_l - P_l(k)).$$

According to assumption (12), we have the conclusion $P_l > P_l(k + 1)$. Combine it with relation (16), by knowledge of mathematical induction, we know that

$$P_i > P_i(k + 1), \quad i \in \mathbb{I}[1, N]. \tag{17}$$

Eventually, with (17) and assumption (12), it could be summarized that for any integer $m \geq 0$, (9) holds. Thus, we complete this proof. \square

Second, it is found that the sequence generated by (8) is monotonically increasing.

Lemma 4. Assume that system (1) has AMSS, and $Q_i > 0, i \in \mathbb{I}[1, N]$. If $\beta_i \geq 0, 0 \leq \gamma < 1$, and the tuning parameters $\alpha_j \in [0, 1]$, then on the condition of zero initial values, the sequence $(P_1(m), P_2(m), \dots, P_N(m))$ produced by algorithm (8) is strictly monotonically incremental. In other words, for any integer $m \geq 0$, we have

$$P_i(m) < P_i(m + 1), \quad i \in \mathbb{I}[1, N]. \tag{18}$$

Proof. The matrices $A_i, i \in \mathbb{I}[1, N]$ are Hurwitz-stable and Equation (2) has the only solution (P_1, P_2, \dots, P_N) according to Lemmas 1 and 2. Let $m = k$ and $m = k + 1$ in (8); subtract the former from the latter to obtain

$$A_i^T [P_i(k + 2) - P_i(k + 1) - \gamma(P_i(k + 1) - P_i(k))] + [P_i(k + 2) - P_i(k + 1) - \gamma(P_i(k + 1) - P_i(k))]A_i = M_i(k), \tag{19}$$

where

$$M_i(k) = (1 - \gamma) \left(- \sum_{s=1}^r A_{si}^T (P_i(k + 1) - P_i(k)) A_{si} - \sum_{j=1}^{i-1} \pi_{ij} [\alpha_j (P_j(k + 2) - P_j(k + 1)) + (1 - \alpha_j) (P_j(k + 1) - P_j(k))] - \sum_{j=i+1}^N \pi_{ij} (P_j(k + 1) - P_j(k)) - \beta_i (P_i(k + 1) - P_i(k)) \right), \quad i \in \mathbb{I}[1, N]. \tag{20}$$

Subsequently, mathematical induction is utilized to help prove the above lemma. Since $P_i(0) = 0, i \in \mathbb{I}[1, N]$, it is easily known that (18) holds for $m = 0$. Now, we suppose when $m = k, k \geq 0$, (18) holds, which means

$$P_i(k) < P_i(k + 1), \quad i \in \mathbb{I}[1, N]. \tag{21}$$

Let $i = 1$ in (20); we can obtain

$$M_1(k) = (1 - \gamma) \left(- \sum_{s=1}^r A_{s1}^T (P_1(k+1) - P_1(k)) A_{s1} - \sum_{j=2}^N \pi_{1j} (P_j(k+1) - P_j(k)) - \beta_1 (P_1(k+1) - P_1(k)) \right). \tag{22}$$

For $0 \leq \gamma < 1$, according to assumption (21), it can be easily found that $M_1(k) < 0$. Therefore, from (19), we can obtain the following inequality:

$$A_1^T [P_1(k+2) - P_1(k+1) - \gamma(P_1(k+1) - P_1(k))] + [P_1(k+2) - P_1(k+1) - \gamma(P_1(k+1) - P_1(k))] A_1 < 0. \tag{23}$$

For the Hurwitz stability of A_1 , we have the following inequality from Theorem 3.1 in [27]:

$$P_1(k+2) - P_1(k+1) - \gamma(P_1(k+1) - P_1(k)) > 0.$$

Equivalently,

$$P_1(k+2) - P_1(k+1) > \gamma(P_1(k+1) - P_1(k)).$$

From assumption (21) that $P_1(k+1) > P_1(k)$, we have

$$P_1(k+2) > P_1(k+1). \tag{24}$$

Using mathematical induction repeatedly, we now assume that

$$P_j(k+2) > P_j(k+1), \quad j \in \mathbb{I}[1, l-1], l \geq 2. \tag{25}$$

For relation (20), let $i = l$; we can obtain

$$M_l(k) = (1 - \gamma) \left(- \sum_{s=1}^r A_{sl}^T (P_l(k+1) - P_l(k)) A_{sl} - \sum_{j=1}^{l-1} \pi_{lj} \left[\alpha_j (P_j(k+2) - P_j(k+1)) + (1 - \alpha_j) (P_j(k+1) - P_j(k)) \right] - \sum_{j=l+1}^N \pi_{lj} (P_j(k+1) - P_j(k)) - \beta_l (P_l(k+1) - P_l(k)) \right).$$

Since $0 \leq \gamma < 1$, if induction assumptions (21) and (25) hold, we can obtain $M_l(k) < 0$. For relation (19) with $i = l$, we have that

$$A_l^T [P_l(k+2) - P_l(k+1) - \gamma(P_l(k+1) - P_l(k))] + [P_l(k+2) - P_l(k+1) - \gamma(P_l(k+1) - P_l(k))] A_l < 0. \tag{26}$$

A_l is Hurwitz-stable; hence,

$$P_l(k+2) - P_l(k+1) - \gamma(P_l(k+1) - P_l(k)) > 0.$$

This is equivalent to

$$P_l(k+2) - P_l(k+1) > \gamma(P_l(k+1) - P_l(k)).$$

In terms of assumption (21), we have $P_i(k + 2) > P_i(k + 1)$. Based on this and relation (25), the following is evident from mathematical induction:

$$P_i(k + 2) > P_i(k + 1), \quad i \in \mathbb{I}[1, N]. \tag{27}$$

Consequently, with (27) and assumption (21), it is proven that for any integer $m \geq 0$, (18) holds. Thus, the proof is finished. \square

Combining Lemmas 3 and 4, the following convergence property of algorithm (8) can be deduced.

Theorem 2. Assume system (1) has AMSS, and $Q_i > 0, i \in \mathbb{I}[1, N]$. If $\beta_i \geq 0, 0 \leq \gamma < 1$, and the tuning parameters $\alpha_j \in [0, 1]$, then on the condition of zero initial values $P_i(0) = 0, i \in \mathbb{I}[1, N]$, the sequence $(P_1(m), P_2(m), \dots, P_N(m))$ produced by algorithm (8) monotonically converges to the unique solution (P_1, P_2, \dots, P_N) of Equation (2).

Proof. From Lemmas 3 and 4 and the conditions of Theorem 2, the sequence $(P_1(m), P_2(m), \dots, P_N(m))$ produced by the SOR implicit algorithm (8) increases monotonically and has an upper bound, which is the solution of (2). It means that

$$0 = P_i(0) < P_i(1) < P_i(2) < \dots < P_i(N) < P_i, \quad i \in \mathbb{I}[1, N].$$

Thus, the sequence $(P_1(m), P_2(m), \dots, P_N(m))$ is convergent. Take the limit on $P_i(k)$, and denote

$$\lim_{k \rightarrow \infty} P_i(k) = P_{i\infty}, \quad i \in \mathbb{I}[1, N].$$

By simultaneously taking limits on both sides of algorithm (8), we can easily obtain

$$\begin{aligned} A_i^T P_{i\infty} + P_{i\infty} A_i &= (1 - \gamma) \left(- \sum_{s=1}^r A_{si}^T P_{i\infty} A_{si} - \sum_{j=1}^{i-1} \pi_{ij} [\alpha_j P_{j\infty} + (1 - \alpha_j) P_{j\infty}] \right. \\ &\quad \left. - \sum_{j=i+1}^N \pi_{ij} P_{j\infty} - \beta_i P_{i\infty} - Q_i \right) + \gamma (A_i^T P_{i\infty} + P_{i\infty} A_i). \end{aligned}$$

This is equivalent to

$$\begin{aligned} &(A_{0i} + 0.5\pi_{ii}I_n - 0.5\beta_i I_n)^T P_{i\infty} + P_{i\infty} (A_{0i} + 0.5\pi_{ii}I_n - 0.5\beta_i I_n) \\ &= (1 - \gamma) \left(- \sum_{s=1}^r A_{si}^T P_{i\infty} A_{si} - \sum_{j=1}^{i-1} \pi_{ij} [\alpha_j P_{j\infty} + (1 - \alpha_j) P_{j\infty}] \right. \\ &\quad \left. - \sum_{j=i+1}^N \pi_{ij} P_{j\infty} - \beta_i P_{i\infty} - Q_i \right) + \gamma \left[(A_{0i} + 0.5\pi_{ii}I_n - 0.5\beta_i I_n)^T P_{i\infty} \right. \\ &\quad \left. + P_{i\infty} (A_{0i} + 0.5\pi_{ii}I_n - 0.5\beta_i I_n) \right], \end{aligned}$$

so

$$\begin{aligned} &(1 - \gamma) [(A_{0i} + 0.5\pi_{ii}I_n - 0.5\beta_i I_n)^T P_{i\infty} + P_{i\infty} (A_{0i} + 0.5\pi_{ii}I_n - 0.5\beta_i I_n)] \\ &= (1 - \gamma) \left(- \sum_{s=1}^r A_{si}^T P_{i\infty} A_{si} - \sum_{j=1}^{i-1} \pi_{ij} [\alpha_j P_{j\infty} + (1 - \alpha_j) P_{j\infty}] \right. \\ &\quad \left. - \sum_{j=i+1}^N \pi_{ij} P_{j\infty} - \beta_i P_{i\infty} - Q_i \right) \\ &= (1 - \gamma) \left(- \sum_{s=1}^r A_{si}^T P_{i\infty} A_{si} - \sum_{j=1}^N \pi_{ij} P_{j\infty} + \pi_{ii} P_{i\infty} - \beta_i P_{i\infty} - Q_i \right). \end{aligned}$$

Dividing both sides by $1 - \gamma$ simultaneously, we obtain

$$\begin{aligned} & (A_{0i} + 0.5\pi_{ii}I_n - 0.5\beta_i I_n)^T P_{i\infty} + P_{i\infty}(A_{0i} + 0.5\pi_{ii}I_n - 0.5\beta_i I_n) \\ &= - \sum_{s=1}^r A_{si}^T P_{i\infty} A_{si} - \sum_{j=1}^N \pi_{ij} P_{j\infty} + \pi_{ii} P_{i\infty} - \beta_i P_{i\infty} - Q_i, \end{aligned}$$

which can be equivalently written as follows:

$$A_{0i}^T P_{i\infty} + P_{i\infty} A_{0i} + \sum_{s=1}^r A_{si}^T P_{i\infty} A_{si} + \sum_{j=1}^N \pi_{ij} P_{j\infty} = -Q_i. \tag{28}$$

Compared with matrix Equation (2), we find that the sequence $(P_{1\infty}, P_{2\infty}, \dots, P_{N\infty})$ is exactly the same solution of the coupled Lyapunov matrix Equation (2). \square

It has been demonstrated that algorithm (8) can operate under a zero initial condition. However, this condition is too harsh. Consequently, a sufficient condition is provided for the convergence of algorithm (8) under an arbitrary initial value.

Theorem 3. Assume that system (1) has AMSS, and $Q_i > 0, i \in \mathbb{I}[1, N]$. Matrices M and W are expressed from Equation (29) as follows:

$$M = -diag(I_n \otimes A_1^T + A_1^T \otimes I_n, I_n \otimes A_2^T + A_2^T \otimes I_n, \dots, I_n \otimes A_N^T + A_N^T \otimes I_n) + (1 - \gamma)\underline{M},$$

where

$$\underline{M} = - \begin{bmatrix} 0 & 0 & \dots & 0 & 0 \\ \alpha_1 \pi_{21} I_{n^2} & 0 & \dots & 0 & 0 \\ \dots & \dots & \ddots & \dots & \dots \\ \alpha_1 \pi_{N-1,1} I_{n^2} & a_2 \pi_{N-1,2} I_{n^2} & \dots & 0 & 0 \\ a_1 \pi_{N1} I_{n^2} & a_2 \pi_{N2} I_{n^2} & \dots & \alpha_{N-1} \pi_{N,N-1} I_{n^2} & 0 \end{bmatrix}.$$

And

$$\begin{aligned} W &= -\gamma diag(I_n \otimes A_1^T + A_1^T \otimes I_n, I_n \otimes A_2^T + A_2^T \otimes I_n, \dots, I_n \otimes A_N^T + A_N^T \otimes I_n) \\ &+ (1 - \gamma)diag(\sum_{s=1}^r A_{s1}^T \otimes A_{s1}^T + \beta_1 I_{n^2}, \sum_{s=1}^r A_{s2}^T \otimes A_{s2}^T + \beta_2 I_{n^2}, \dots, \sum_{s=1}^r A_{sN}^T \otimes A_{sN}^T + \beta_N I_{n^2}) \\ &+ (1 - \gamma)\underline{W}, \end{aligned}$$

where

$$\underline{W} = \begin{bmatrix} 0 & \pi_{12} I_{n^2} & \dots & \pi_{1,N-1} I_{n^2} & \pi_{1N} I_{n^2} \\ (1 - \alpha_1) \pi_{21} I_{n^2} & 0 & \dots & \pi_{2,N-1} I_{n^2} & \pi_{2N} I_{n^2} \\ \dots & \dots & \ddots & \dots & \dots \\ (1 - \alpha_1) \pi_{N-1,1} I_{n^2} & (1 - \alpha_2) \pi_{N-1,2} I_{n^2} & \dots & 0 & \pi_{N-1,N} I_{n^2} \\ (1 - \alpha_1) \pi_{N1} I_{n^2} & (1 - \alpha_2) \pi_{N2} I_{n^2} & \dots & (1 - \alpha_{N-1}) \pi_{N,N-1} I_{n^2} & 0 \end{bmatrix}.$$

If the tuning parameters α_j and β_i , and the relaxation parameter γ , satisfy $\rho(M^{-1}W) < 1$, then the sequence $(P_1(m), P_2(m), \dots, P_N(m))$ produced by the SOR implicit algorithm (8), and with any initial condition $(P_1(0), P_2(0), \dots, P_N(0)), i \in \mathbb{I}[1, N]$ will converge to the solution of Equation (2).

Proof. We define the vectors ξ and q as

$$\begin{aligned} \xi &= [[vec(P_1)]^T, [vec(P_2)]^T, \dots, [vec(P_N)]^T]^T, \\ \xi(m) &= [[vec(P_1(m))]^T, [vec(P_2(m))]^T, \dots, [vec(P_N(m))]^T]^T, \\ q &= [(1 - \gamma)vec(Q_1)]^T, [(1 - \gamma)vec(Q_2)]^T, \dots, [(1 - \gamma)vec(Q_N)]^T]^T. \end{aligned}$$

Next, algorithm (8) can be rewritten in the form of Kronecker product:

$$\begin{aligned}
 & -(I_n \otimes A_i^T + A_i^T \otimes I_n) \text{vec}(P_i(m+1)) - (1-\gamma) \sum_{j=1}^{i-1} \alpha_j \pi_{ij} \text{vec}(P_j(m+1)) \\
 &= (1-\gamma) \left(\sum_{s=1}^r A_{si}^T \otimes A_{si}^T + \beta_i I_{n^2} \right) \text{vec}(P_i(m)) \\
 &\quad - \gamma (I_n \otimes A_i^T + A_i^T \otimes I_n) \text{vec}(P_i(m)) \\
 &\quad + (1-\gamma) \sum_{j=1}^{i-1} \pi_{ij} (1-\alpha_j) \text{vec}(P_j(m)) \\
 &\quad + (1-\gamma) \sum_{j=i+1}^N \pi_{ij} \text{vec}(P_j(m)) \\
 &\quad + (1-\gamma) \text{vec}(Q_i).
 \end{aligned} \tag{29}$$

The definitions of matrices M and W are based on the above equation. Combining the vector ζ , q , and matrices M and W defined earlier, we can further obtain

$$M\zeta(m+1) = W\zeta(m) + q.$$

By multiplying the inverse matrix of M from the left, we have

$$\zeta(m+1) = M^{-1}W\zeta(m) + M^{-1}q. \tag{30}$$

According to the relationship between the previous and subsequent vectors ζ , we can deduce repeatedly and obtain

$$\zeta(m+1) = (M^{-1}W)^{m+1}\zeta(0) + \sum_{j=0}^m (M^{-1}W)^j M^{-1}q.$$

If $\rho(M^{-1}W) < 1$, then when m approaches infinity, $(M^{-1}W)^k$ approaches a zero matrix; hence,

$$\lim_{m \rightarrow \infty} \zeta(m+1) = \lim_{m \rightarrow \infty} \sum_{j=0}^m (M^{-1}W)^j M^{-1}q = (M - W)^{-1}q. \tag{31}$$

Since the matrix Equation (2) could be rewritten as $\zeta = (M - W)^{-1}q$, the sequence $\{\zeta(m)\}$ generated by the SOR algorithm (8) converges to the unique solution of Equation (2). \square

4. Illustrative Example

Consider Equation (2), in which $N = 2$, $r = 1$. Q_1, Q_2 are both fourth-order identity matrices. The system matrices and probability transition rate matrix Π are defined as follows:

$$\begin{aligned}
 A_{01} &= \begin{bmatrix} -1.0000 & -1.0000 & -2.0000 & 1.0000 \\ -0.6667 & -3.5000 & 1.0000 & 1.1670 \\ 1.0000 & 0.5000 & -3.0000 & 0.5000 \\ -2.0000 & -2.5000 & 1.0000 & -2.5000 \end{bmatrix}, \\
 A_{02} &= \begin{bmatrix} -1.3333 & 1.0000 & -2.0000 & 0.3333 \\ 0.0000 & -3.5000 & 1.0000 & 1.5000 \\ 1.0000 & 1.5000 & -4.0000 & -0.5000 \\ -1.3333 & -3.5000 & 1.0000 & -1.1667 \end{bmatrix}.
 \end{aligned}$$

$$A_{11} = \begin{bmatrix} 0.9003 & 0.7826 & 0.6428 & 0.8436 \\ -0.5377 & 0.5242 & -0.1106 & 0.4764 \\ 0.2137 & -0.0871 & 0.2309 & -0.6475 \\ -0.0280 & -0.9630 & 0.5839 & -0.1886 \end{bmatrix},$$

$$A_{12} = \begin{bmatrix} 0.8709 & -0.8842 & -0.7222 & -0.4556 \\ 0.8338 & -0.2943 & -0.5945 & -0.6024 \\ -0.1796 & 0.6263 & -0.6026 & -0.9695 \\ 0.7873 & -0.9803 & 0.2076 & 0.4936 \end{bmatrix},$$

$$\Pi = \begin{bmatrix} -0.6 & 0.6 \\ 1 & -1 \end{bmatrix}.$$

This example is used in [25]. The iterative error at step k is defined as

$$\epsilon(k) = \sqrt{\sum_{i=1}^N \left\| A_{0i}^T P_i(k) + P_i(k) A_{0i} + \sum_{s=1}^r A_{si}^T P_i(k) A_{si} + \sum_{j=1}^N \pi_{ij} P_j(k) + Q_i \right\|_F^2}.$$

First, we investigated the convergence of algorithm (8) when $\alpha = \alpha_1 = 1$ and $\beta = \beta_1 = \beta_2$. The value of β was chosen to be the same as that in [25]. In order to make the Figure looks concise, in Figure 1, we only selected three sets of different tuning parameter values from [25] and compared the convergence performance after selecting a proper parameter γ with the Matlab function. We found that by adjusting the value of parameter γ , a smaller spectral radius $\rho(M^{-1}W)$ and fewer iteration steps k could be obtained. The convergence curves are shown in Figure 1. It should be mentioned that when γ takes the value of 0, algorithm (8) degenerates to algorithm (5), which is proposed in [25], so the two algorithms show the same convergence curves. The red convergence curve represents the convergence situation of algorithm (8), while the black curve represents that of algorithm (5). So, it is evident that when the values of α and β remained constant, selecting the appropriate parameter γ achieved better convergence results. In [25], the minimum spectral radius $\rho(M^{-1}W) = 0.3128$ was gained when $\alpha = 1$ and $\beta = -0.4240$. By using the Matlab function, we found that under this condition, it converged fast only when γ took the value of 0. But when $\alpha = 1$ and $\beta = -1$, we obtained a smaller $\rho(M^{-1}W) = 0.2638$ by choosing $\gamma = 0.147$. It is obvious that when the values of α and β were chosen to be the same as the algorithm (5) proposed in [25], we achieved a faster convergence rate by selecting different values of parameter γ with the help of Matlab (version R2017a).

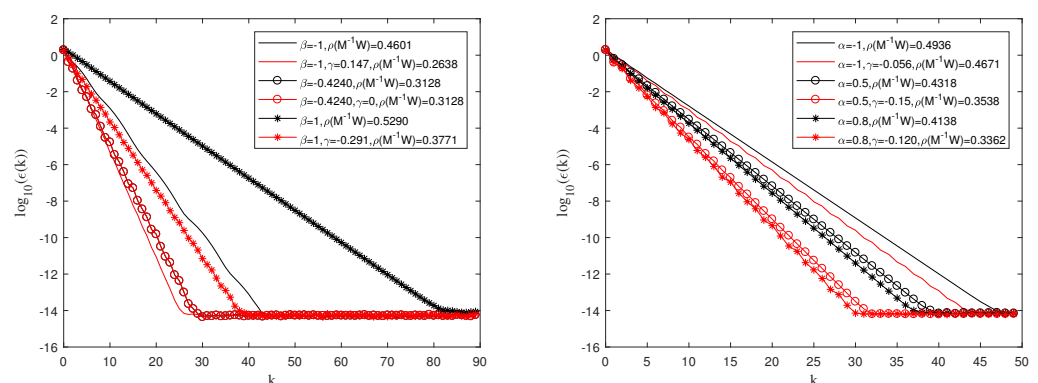


Figure 1. Convergence performance comparisons between algorithms (5) and (8) after selecting appropriate relaxation parameters when $\alpha=1$ (left) and $\beta = 0$ (right).

Figure 1 shows the algorithm’s convergence after selecting appropriate relaxation parameters. The left figure displays the case when $\alpha = 1$, and the right figure displays the case when $\beta = 0$. The value of α was chosen to be the same as the algorithm in [25]. By

adjusting the appropriate parameter γ , we still acquired a smaller spectral radius $\rho(M^{-1}W)$ and fewer iteration numbers. Obviously, when the values of α and β were the same as in algorithm (5) proposed in [25], a better convergence performance was obtained by choosing the appropriate parameter γ with the Matlab function. Figure 1 indicates that all the three parameters α , β , and γ simultaneously influence the convergence rate of the algorithm. And through extensive calculations, we found that the fastest convergence rate is approximately achieved when the spectral radius is around 0.3.

In algorithm (8), by employing a Matlab function, spectral radius $\rho(M^{-1}W)$ with different β and γ were calculated, with the value of α fixed. Through calculation verification, we found that if $\alpha \in [0, 1]$, no matter what value α took, it was always possible to obtain a spectral radius smaller than that of algorithm (5) under the same values of α and β by selecting the appropriate γ . For an iterative algorithm, a smaller spectral radius means a faster convergence speed and a better convergence performance. Since this study had three tuning parameters, it was necessary to first fix the value of one tuning parameter before analyzing the convergence effect of the algorithm. In Figure 2, due to space limitations, we only selected three representative α values as $\alpha = 0$, $\alpha = 0.5$, and $\alpha = 1$, since in the previous text we have $\alpha_j \in [0, 1]$.

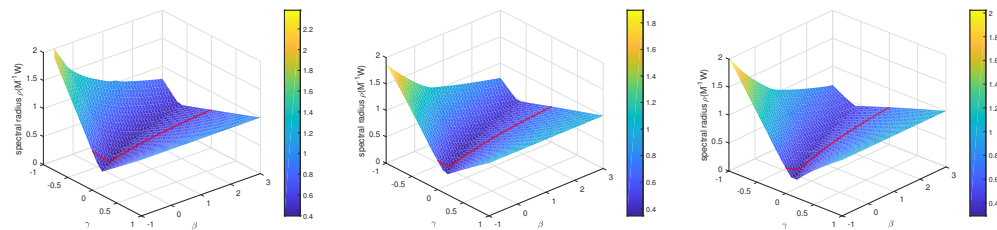


Figure 2. Spectral radii of $M^{-1}W$ for different parameters β and γ in algorithm (8) when $\alpha = 0$ (left), $\alpha = 0.5$ (middle), and $\alpha = 1$ (right).

Figure 2 displays the spectral radius $\rho(M^{-1}W)$ when β , γ take different values and $\alpha = 0, 0.5, 1$, respectively. The red line represents $\gamma = 0$, which is the spectral radius of algorithm (5). In Figure 3, the lowest valley of the surface, which has the smallest spectral radius, is significantly lower than the red curve. This means that when the values of α and β are the same as those of algorithm (5), through selecting the appropriate relaxation parameter γ , algorithm (8) always yields a smaller spectral radius.

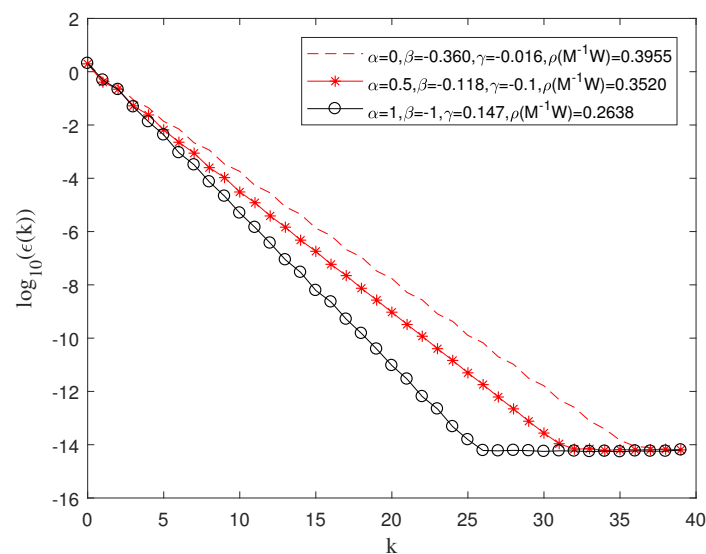


Figure 3. Convergence performance of different parameters in the preceding cases.

We found the optimal parameters of β and γ in Figure 2, when $\alpha = 0, 0.5$, and 1 , respectively, and presented the convergence curves for these three sets of parameters in Figure 3. It is worth mentioning that when the value of α was fixed to 1 , by adjusting the value of the parameter γ , we obtained many spectral radii around 0.26 . Algorithm (5) proposed in [25] obtained the minimum spectral radius around approximately 0.31 . Obviously, as the parameters changed, algorithm (8) could achieve a better convergence performance.

5. Conclusions

In this article, a novel implicit iterative algorithm for solving continuous stochastic Lyapunov equations is presented via SOR iteration. And the up-to-date information of the solution is utilized to be the latest estimates of the unknown matrices. This method includes three parameters. When proper parameters are chosen, the convergence performance can be greatly improved. Furthermore, the convergence properties are analyzed in detail. Based on the method proposed in [25], we added a parameter to improve the convergence of the algorithm. We always obtained a faster convergence rate by selecting a proper parameter γ with the Matlab function. Finally, a numerical example is employed to illustrate that algorithm (8) has a better convergence performance than algorithm (5) proposed in [25].

Author Contributions: Conceptualization, P.H. and T.W.; methodology, P.H.; software, T.W.; validation, P.H., T.W. and H.C.; formal analysis, H.C.; resources, P.H.; writing—original draft preparation, T.W.; writing—review and editing, P.H. and H.C.; visualization, T.W.; supervision, P.H.; project administration, P.H. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by Natural Science Foundation of Jiangsu Province grant number BK20191386.

Data Availability Statement: No new data were created or analyzed in this study. Data sharing is not applicable to this article.

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

References

- Li, Y.; Karimi, H.R.; Zhang, Q.; Zhao, D.; Li, Y. Fault detection for linear discrete time-varying systems subject to random sensor delay: A Riccati equation approach. *IEEE Trans. Circuits Syst. Regul. Pap.* **2017**, *65*, 1707–1716. [\[CrossRef\]](#)
- Li, Y.; Zhao, W. Haar wavelet operational matrix of fractional order integration and its applications in solving the fractional order differential equations. *Appl. Math. Comput.* **2010**, *216*, 2276–2285. [\[CrossRef\]](#)
- Zhang, L.; Zhang, G.; Liu, W. Optimal control of second-order and high-order descriptor systems. *Optim. Control Appl. Methods* **2019**, *40*, 791–806. [\[CrossRef\]](#)
- Zhao, X.; Deng, F. Solution of the HJI equations for nonlinear H_∞ control design by state-dependent Riccati equations approach. *J. Syst. Eng. Electron.* **2011**, *22*, 654–660. [\[CrossRef\]](#)
- Hazell, A.; Limebeer, D.J. An efficient algorithm for discrete-time H_∞ preview control. *Automatica* **2008**, *44*, 2441–2448. [\[CrossRef\]](#)
- Hashemi, B.; Dehghan, M. The interval Lyapunov matrix equation: Analytical results and an efficient numerical technique for outer estimation of the united solution set. *Math. Comput. Model.* **2012**, *55*, 622–633. [\[CrossRef\]](#)
- Sun, H.J.; Zhang, J.; Wu, Y.Y. A Newton iterative method for coupled Lyapunov matrix equations. *J. Ind. Manag. Optim.* **2023**, *19*, 8791–8806. [\[CrossRef\]](#)
- Rami, M.A.; Zhou, X.Y. Linear matrix inequalities, Riccati equations, and indefinite stochastic linear quadratic controls. *IEEE Trans. Autom. Control* **2000**, *45*, 1131–1143. [\[CrossRef\]](#)
- Dragan, V.; Morozan, T. Stability and robust stabilization to linear stochastic systems described by differential equations with Markovian jumping and multiplicative white noise. *Stoch. Anal. Appl.* **2002**, *20*, 33–92. [\[CrossRef\]](#)
- Zhang, W.; Zhang, H.; Chen, B.S. Generalized Lyapunov equation approach to state-dependent stochastic stabilization/detectability criterion. *IEEE Trans. Autom. Control* **2008**, *53*, 1630–1642. [\[CrossRef\]](#)
- Li, Z.Y.; Wang, Y.; Zhou, B.; Duan, G.R. On unified concepts of detectability and observability for continuous-time stochastic systems. *Appl. Math. Comput.* **2010**, *217*, 521–536. [\[CrossRef\]](#)
- Ji, Y.; Chizeck, H.J. Controllability, stabilizability, and continuous-time Markovian jump linear quadratic control. *IEEE Trans. Autom. Control* **1990**, *35*, 777–788. [\[CrossRef\]](#)
- Huang, Y.; Zhang, W.; Feng, G. Infinite horizon H_2/H_∞ control for stochastic systems with Markovian jumps. *Automatica* **2008**, *44*, 857–863. [\[CrossRef\]](#)
- Ni, Y.; Zhang, W.; Fang, H. On the observability and detectability of linear stochastic systems with Markov jumps and multiplicative noise. *J. Syst. Sci. Complex.* **2010**, *23*, 102–115. [\[CrossRef\]](#)

15. Li, Z.Y.; Zhou, B.; Lam, J.; Wang, Y. Positive operator based iterative algorithms for solving Lyapunov equations for Itô stochastic systems with Markovian jumps. *Appl. Math. Comput.* **2011**, *217*, 8179–8195. [[CrossRef](#)]
16. Shen, L. On the detectability and observability of continuous stochastic Markov jump linear systems. *J. Math. Anal. Appl.* **2015**, *424*, 878–891. [[CrossRef](#)]
17. Wang, Q.; Lam, J.; Wei, Y.; Chen, T. Iterative solutions of coupled discrete Markovian jump Lyapunov equations. *Comput. Math. Appl.* **2008**, *55*, 843–850. [[CrossRef](#)]
18. Zhou, B.; Lam, J.; Duan, G.R. Convergence of gradient-based iterative solution of coupled Markovian jump Lyapunov equations. *Comput. Math. Appl.* **2008**, *56*, 3070–3078. [[CrossRef](#)]
19. Jodar, L.; Mariton, M. Explicit solutions for a system of coupled Lyapunov differential matrix equations. *Proc. Edinb. Math. Soc.* **1987**, *30*, 427–434. [[CrossRef](#)]
20. Ding, F.; Chen, T. Iterative least-squares solutions of coupled Sylvester matrix equations. *Syst. Control Lett.* **2005**, *54*, 95–107. [[CrossRef](#)]
21. Zhang, H. Reduced-rank gradient-based algorithms for generalized coupled Sylvester matrix equations and its applications. *Comput. Math. Appl.* **2015**, *70*, 2049–2062. [[CrossRef](#)]
22. Wu, A.G.; Duan, G.R.; Liu, W. Implicit iterative algorithms for continuous Markovian jump Lyapunov equations. *IEEE Trans. Autom. Control* **2016**, *61*, 3183–3189. [[CrossRef](#)]
23. Young, D.M. *Iterative Solution of Large Linear Systems*; Elsevier: Amsterdam, The Netherlands, 2014.
24. Wu, A.G.; Zhang, W.X.; Zhang, Y. An iterative algorithm for discrete periodic Lyapunov matrix equations. *Automatica* **2018**, *87*, 395–403. [[CrossRef](#)]
25. Wu, A.G.; Wang, X.; Sreeram, V. Iterative algorithms for solving continuous stochastic Lyapunov equations. *IET Control Theory Appl.* **2017**, *11*, 73–80. [[CrossRef](#)]
26. Wu, A.G.; Sun, H.J.; Zhang, Y. An SOR implicit iterative algorithm for coupled Lyapunov equations. *Automatica* **2018**, *97*, 38–47. [[CrossRef](#)]
27. Feitzinger, F.; Hylla, T.; Sachs, E.W. Inexact Kleinman–Newton method for Riccati equations. *SIAM J. Matrix Anal. Appl.* **2009**, *31*, 272–288. [[CrossRef](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.