

## Research Article

# The Extended Hamiltonian Algorithm for the Solution of the Algebraic Riccati Equation

Zhikun Luo, Huafei Sun, and Xiaomin Duan

*School of Mathematics, Beijing Institute of Technology, Beijing 100081, China*

Correspondence should be addressed to Huafei Sun; [huafeisun@bit.edu.cn](mailto:huafeisun@bit.edu.cn)

Received 17 March 2014; Revised 27 April 2014; Accepted 8 May 2014; Published 20 May 2014

Academic Editor: Naseer Shahzad

Copyright © 2014 Zhikun Luo et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

We use a second-order learning algorithm for numerically solving a class of the algebraic Riccati equations. Specifically, the extended Hamiltonian algorithm based on manifold of positive definite symmetric matrices is provided. Furthermore, this algorithm is compared with the Euclidean gradient algorithm, the Riemannian gradient algorithm, and the new subspace iteration method. Simulation examples show that the convergence speed of the extended Hamiltonian algorithm is the fastest one among these algorithms.

## 1. Introduction

The algebraic Riccati equations (AREs) have been widely used in control system syntheses [1, 2], especially in optimal control [3], robust control [4], signal processing [5], and the LMI-based design [6]. And, in the past several decades, there has been an increasing interest in the solution problems of the AREs [7–9]. The lower matrix bounds for the AREs were discussed in [10, 11]. The structure-preserving doubling algorithm (SDA) was given under assumptions which are weaker than stabilizability and detectability, as well as practical issues involved in the application of the SDA to continuous-time AREs [12]. It is the purpose of this paper to investigate the unique (under suitable hypotheses) symmetric positive definite solution of an algebraic Riccati equation. Some effective methods, such as the Schur method (SM) [13] and the new subspace iteration method (NSIM) [14], were given for the numerical solution of the algebraic Riccati equation. Recently, the Euclidean gradient algorithm (EGA) and the Riemannian gradient algorithm (RGA) for the numerical solution of the algebraic Riccati equation with a cost function of the Riemannian distance on the curved Riemannian manifold were proposed in [15].

It is well known that the Euclidean gradient algorithm (EGA) and the Riemannian gradient algorithm (RGA) are first-order learning algorithms; hence the convergence speeds

of the EGA and RGA are very slow. The inclusion of a momentum term had been found to increase the rate of convergence dramatically [16, 17]. Based on this, the extended Hamiltonian algorithm (EHA) on manifolds was developed in [18], while its numerical implementation was discussed in [19]. Furthermore, the EHA is a second-order learning algorithm and converges faster than the first-order learning algorithm for optimization problem if certain conditions are satisfied.

In this paper, we will apply the EHA to give the numerical solution of the AREs. Furthermore, we give two simulations to show the efficiency of our algorithm.

Briefly, the rest of the paper is organized as follows. Section 2 is concluded with some fundamental knowledge on manifold that will be used throughout the paper. Section 3 introduces the extended Hamiltonian algorithm on manifold of positive definite symmetric matrices and Section 4 illustrates the convergence speeds of the EHA compared with other algorithms using two numerical examples. Section 5 concludes the paper.

## 2. Preliminaries

In this section we recall some differential geometric facts of the space of positive definite symmetric matrices that will be used in the present analysis. More details can be found in [20–23].

### 2.1. The Frobenius Inner Product and the Euclidean Distance.

Let  $M(n)$  be the set of  $n \times n$  real matrices and  $GL(n)$  be its subset containing all the nonsingular matrices.  $GL(n)$  is a Lie group, that is, a group which is also a differentiable manifold and for which the operations of group multiplication and inverse are smooth. The tangent space at the identity is called the corresponding Lie algebra denoted by  $\mathfrak{gl}(n)$ . It is the space of all linear transformations in  $\mathbb{R}^n$ , that is,  $M(n)$ .

On  $M(n)$ , we use the Euclidean inner product, known as the Frobenius inner product and defined by

$$\langle P, Q \rangle_F = \text{tr}(P^T Q), \quad (1)$$

where  $\text{tr}$  stands for the trace of the matrix and superscript  $T$  denotes the transpose. The associated norm  $\|P\|_F = [\text{tr}(P^T P)]^{1/2}$  is used to define the Euclidean distance on  $M(n)$  by

$$d_F(P, Q) = \|P - Q\|_F. \quad (2)$$

2.2. *Geometry of Positive Definite Symmetric Matrices.* We denote by

$$\text{Sym}(n) = \{s \in M(n) \mid s = s^T\} \quad (3)$$

the vector space of symmetric matrices and denote by

$$\text{PD}(n) = \{x \in \text{Sym}(n) \mid x > 0\} \quad (4)$$

the set of all  $n \times n$  positive definite symmetric matrices, which is a differentiable manifold of dimension  $n(n+1)/2$ . Here  $x > 0$  means the quadratic form  $z^T x z > 0$  for all  $z \in \mathbb{R}^n \setminus \{0\}$ . Furthermore,  $\text{PD}(n)$  is an open convex cone; namely, if  $x_1$  and  $x_2$  are in  $\text{PD}(n)$ , so is  $x_1 + tx_2$  for any  $t > 0$ .

The exponential of any symmetric matrix is a positive definite symmetric matrix and the (principal) logarithm of any positive definite symmetric matrix is a symmetric matrix. Thus the exponential map from  $\text{Sym}(n)$  to  $\text{PD}(n)$  is one-to-one and onto. It follows that  $\text{Sym}(n)$  provides a parameterization of  $\text{PD}(n)$  via the exponential map.

2.3. *Metric and Distances on  $\text{PD}(n)$ .* Let  $g$  denote the Riemannian metric on manifold  $\text{PD}(n)$ . For any  $x \in \text{PD}(n)$  the tangent space  $T_x \text{PD}(n)$  is identified with  $\text{Sym}(n)$ . On  $T_x \text{PD}(n)$  we define the inner product and the corresponding norm as follows:

$$\begin{aligned} g_x(u, v) &= \langle u, v \rangle_x = \text{tr}(x^{-1} u x^{-1} v), \\ \|v\|_x &= \left[ \text{tr}((x^{-1} v)^2) \right]^{1/2}, \end{aligned} \quad (5)$$

which depend on the point  $x$ . The positive definiteness of this metric is a consequence of the positive definiteness of the Frobenius inner product for

$$\begin{aligned} g_x(v, v) &= \langle v, v \rangle_x = \text{tr}(x^{-1/2} v x^{-1/2} x^{-1/2} v x^{-1/2}) \\ &= \langle x^{-1/2} v x^{-1/2}, x^{-1/2} v x^{-1/2} \rangle. \end{aligned} \quad (6)$$

Compared to manifold  $\text{PD}(n)$  with the Frobenius inner product (1) resulting in a flat Riemannian manifold, manifold  $\text{PD}(n)$  with the Riemannian metric (5) becomes a curved Riemannian manifold.

To measure closeness of two positive definite symmetric matrices  $x_1$  and  $x_2$  one can use the Euclidean distance (2) of the ambient space  $M(n)$ ; that is,

$$d_F(x_1, x_2) = \|x_1 - x_2\|_F. \quad (7)$$

It will be more appropriate to use the Riemannian distance induced from (5), which is intrinsic to  $\text{PD}(n)$  and defined by

$$d_R(x_1, x_2) = \left\| \log(x_1^{-1/2} x_2 x_1^{-1/2}) \right\|_F = \left( \sum_{i=1}^n \ln^2 \lambda_i \right)^{1/2}, \quad (8)$$

where  $\lambda_i$ ,  $i = 1, 2, \dots, n$ , are the positive eigenvalues of  $x_1^{-1} x_2$ . The positivity of the  $\lambda_i$  is a consequence of the similarity between the (in general nonsymmetric) matrix  $x_1^{-1} x_2$  and the positive definite symmetric matrix  $x_1^{-1/2} x_2 x_1^{-1/2}$ . It is straightforward to see that the Riemannian distance (8) is invariant under inversion, that is,

$$d_R(x_1^{-1}, x_2^{-1}) = d_R(x_1, x_2) \quad (9)$$

and is also invariant under congruent transformations; that is,

$$d_R(x_1, x_2) = d_R(P^T x_1 P, P^T x_2 P), \quad (10)$$

where  $P \in GL(n)$ .

Now, we discuss the geodesics on manifold  $\text{PD}(n)$  under the different metrics. The geodesic passing through  $x$  in the direction of  $v \in \text{Sym}(n)$  with respect to the Euclidean metric (2) is given by

$$\gamma(t) = x + tv, \quad 0 \leq t < \delta \quad (11)$$

for some positive number  $\delta > 0$ . The restriction that  $t$  is not too large is necessary in order to ensure that the line stays within  $\text{PD}(n)$ . Similarly, the geodesic passing through  $x$  in the direction of  $v \in \text{Sym}(n)$  with respect to the Riemannian metric (5) is expressed by

$$\gamma(t) = x^{1/2} \exp(x^{-1/2} v x^{-1/2} t) x^{1/2}, \quad (12)$$

which does not suffer the previous restriction. Let the exponential map  $\text{Exp}: \text{Sym}(n) \rightarrow \text{PD}(n)$ ; then

$$x_2 = \text{Exp}_{x_1}(v) = x_1^{1/2} \exp(x_1^{-1/2} v x_1^{-1/2}) x_1^{1/2}. \quad (13)$$

Notice that the geodesic (12) is defined for all values of  $t$ , which is called geodesic completeness. The Hopf-Rinow theorem states that a geodesic complete manifold is also a complete metric space. And, manifold  $\text{PD}(n)$  with the Riemannian metric (5) has nonpositive sectional curvature.

### 3. The Extended Hamiltonian Algorithm for the Solution of AREs

**3.1. Problem Formulation.** The infinite time state regulator, namely, the state equation of linear time-invariant system, is in the form

$$\dot{y}(t) = Ay(t) + Bw(t), \quad y(t_0) = y_0, \quad (14)$$

where  $y(t) \in \mathbb{R}^n$  and  $w(t) \in \mathbb{R}^m$  are the state and the control vector, respectively,  $A$  and  $B$  are constant matrices of appropriate dimensions, and the terminal time  $t_f = \infty$ . The problem would be to find an optimal control  $w(t)$  satisfying (14) while minimizing the quadratic performance index as follows:

$$J = \frac{1}{2} \int_0^{\infty} [y^T(t)Qy(t) + w^T(t)Rw(t)] dt, \quad (15)$$

where the designed parameters satisfy that both  $Q$  and  $R$  are positive definite symmetric matrices. It has been proved that the optimal control  $w^*(t)$  of the cost function (15) is uniquely given by [24]

$$w^*(t) = -R^{-1}B^Txy(t) \quad (16)$$

and the quadratic optimal performance index is gotten by

$$J^*(y(t_0)) = \frac{1}{2}y^T(t_0)xy(t_0), \quad (17)$$

where  $x$  is a positive definite symmetric matrix satisfying the AREs; that is,

$$xA + A^Tx - xBR^{-1}B^Tx + Q = 0. \quad (18)$$

It is assumed that the pair  $(A, B)$  is completely controllable and that the pair  $(A, D)$  is completely observable, where  $DD^T = Q$ . Under these assumptions, (18) is known to have a unique positive definite solution  $x$  [25]. There are, of course, other solutions to (18), but for the algorithm presented in this paper the emphasis will be on computing the positive definite one. Moreover, the optimal trajectory  $y(t)$  satisfies the following optimal asymptotically stable closed-loop system:

$$\dot{y}(t) = (A - BR^{-1}B^Tx)y(t) = (A - BK)y(t), \quad (19)$$

where the state-feedback gain  $K$  is given by

$$K = R^{-1}B^Tx. \quad (20)$$

In order to solve (18), our purpose is to seek a matrix  $x$  on manifold  $PD(n)$  such that the matrix  $xBR^{-1}B^Tx - xA - A^Tx$  is as close as possible to the given matrix  $Q$ .

**3.2. Distances and Gradients.** In fact, the idea of formulating an equation on a manifold as a minimal-distance problem was suggested previously in [26]. Hence we use the two

distance functions (7) and (8), and the difference between  $Q$  and  $xBR^{-1}B^Tx - xA - A^Tx$  can be expressed as follows:

$$J_F = \frac{1}{2}d_F^2(Q, xBR^{-1}B^Tx - xA - A^Tx) \\ = \frac{1}{2} \operatorname{tr} \left[ (Q + xA + A^Tx - xBR^{-1}B^Tx)^2 \right], \quad (21)$$

$$J_R = \frac{1}{2}d_R^2(Q, xBR^{-1}B^Tx - xA - A^Tx) \\ = \frac{1}{2} \left\| \log(Q^{-1/2}(xBR^{-1}B^Tx - xA - A^Tx)Q^{-1/2}) \right\|_F^2 \\ = \frac{1}{2} \operatorname{tr} \left\{ \log^2 [Q^{-1/2}(xBR^{-1}B^Tx - xA - A^Tx)Q^{-1/2}] \right\}. \quad (22)$$

Let  $J : PD(n) \rightarrow \mathbb{R}$  be a scalar function of  $n \times n$  matrix. The Riemannian gradient of  $J$  at  $x$ , denoted by  $\nabla_x J$ , is a map from  $PD(n)$  to  $\operatorname{Sym}(n)$  that satisfies  $\langle \nabla_x J, s \rangle_x = \langle \partial_x J, s \rangle_F$ , for every  $s \in \operatorname{Sym}(n)$ , where  $\partial_x J$  is the Euclidean gradient of  $J$  at  $x$ . It follows from the inner product (5) that [27]

$$\nabla_x J = x(\partial_x J)x. \quad (23)$$

Next, we will state some known facts, which are very helpful in computing  $\partial_x J_F$  and  $\nabla_x J_R$ .

**Lemma 1** (see [28]). *Let  $A, B$  denote two constant matrices and let  $X, Y$  denote matrix functions. One has the following properties:*

- (1)  $d \operatorname{tr}(X) = \operatorname{tr}(dX)$ ,
- (2)  $d(A) = 0$ ,
- (3)  $d(AXB) = A(dX)B$ ,
- (4)  $d(X \pm Y) = dX \pm dY$ ,
- (5)  $d(XY) = (dX)Y + X(dY)$ ,

where  $d$  denotes the differential of a matrix (scalar) function.

**Lemma 2** (see [28]). *Let  $f(x)$  be a scalar function of  $n \times n$  matrix  $x$ , if*

$$df(x) = \operatorname{tr}(Wdx) \quad (24)$$

holds; then

$$\partial_x f(x) = W^T, \quad (25)$$

where  $\partial_x f(x)$  denotes the gradient of  $f(x)$  and  $W$  denotes the derivative of  $f(x)$ .

**Theorem 3.** *Let  $J_F(x)$  be a scalar function of  $n \times n$  matrix  $x$ , if*

$$J_F(x) = \frac{1}{2} \operatorname{tr} \left[ (Q + xA + A^Tx - xBR^{-1}B^Tx)^2 \right] \quad (26)$$

holds; then

$$\partial_x J_F(x) \\ = (A - BR^{-1}B^Tx)(Q + xA + A^Tx - xBR^{-1}B^Tx) \\ + (Q + xA + A^Tx - xBR^{-1}B^Tx)(A^T - xBR^{-1}B^T). \quad (27)$$

*Proof.* Since  $Q + xA + A^T x - xBR^{-1}B^T x$  is symmetric, we have from Lemma 1 that

$$\begin{aligned}
 dJ_F(x) &= \frac{1}{2} d \left\{ \text{tr} \left[ \left( Q + xA + A^T x - xBR^{-1}B^T x \right)^2 \right] \right\} \\
 &= \frac{1}{2} \text{tr} \left[ d \left( Q + xA + A^T x - xBR^{-1}B^T x \right)^2 \right] \\
 &= \text{tr} \left[ \left( Q + xA + A^T x - xBR^{-1}B^T x \right) \right. \\
 &\quad \left. \times d \left( Q + xA + A^T x - xBR^{-1}B^T x \right) \right] \\
 &= \text{tr} \left[ \left( Q + xA + A^T x - xBR^{-1}B^T x \right) \right. \\
 &\quad \left. \times \left( dxA + A^T dx - dxBR^{-1}B^T x - xBR^{-1}B^T dx \right) \right] \\
 &= \text{tr} \left\{ \left[ \left( A - BR^{-1}B^T x \right) \left( Q + xA + A^T x - xBR^{-1}B^T x \right) \right. \right. \\
 &\quad \left. \left. + \left( Q + xA + A^T x - xBR^{-1}B^T x \right) \right. \right. \\
 &\quad \left. \left. \times \left( A^T - xBR^{-1}B^T \right) \right] dx \right\}. \tag{28}
 \end{aligned}$$

Using the above Lemma 2, we have

$$\begin{aligned}
 \partial_x J_F(x) &= \left[ \left( A - BR^{-1}B^T x \right) \left( Q + xA + A^T x - xBR^{-1}B^T x \right) \right. \\
 &\quad \left. + \left( Q + xA + A^T x - xBR^{-1}B^T x \right) \left( A^T - xBR^{-1}B^T \right) \right]^T \\
 &= \left( A - BR^{-1}B^T x \right) \left( Q + xA + A^T x - xBR^{-1}B^T x \right) \\
 &\quad + \left( Q + xA + A^T x - xBR^{-1}B^T x \right) \left( A^T - xBR^{-1}B^T \right). \tag{29}
 \end{aligned}$$

Therefore, it is shown that (27) is valid.  $\square$

**Theorem 4.** Let  $J_R(x)$  be a scalar function of  $n \times n$  matrix  $x$ , if

$$J_R(x) = \frac{1}{2} \text{tr} \left\{ \log^2 \left[ Q^{-1/2} \left( xBR^{-1}B^T x - xA - A^T x \right) Q^{-1/2} \right] \right\} \tag{30}$$

holds; then

$$\begin{aligned}
 \nabla_x J_R(x) &= x \left\{ \left( xBR^{-1}B^T x - xA - A^T x \right)^{-1} \right. \\
 &\quad \times \log \left[ Q^{-1/2} \left( xBR^{-1}B^T x - xA - A^T x \right) Q^{-1/2} \right] \\
 &\quad \times \left( xBR^{-1}B^T - A^T \right) \\
 &\quad \left. - \left( BR^{-1}B^T x + A \right) \left( xBR^{-1}B^T x - xA - A^T x \right)^{-1} \right. \\
 &\quad \left. \times \log \left[ Q^{-1/2} \left( xBR^{-1}B^T x - xA - A^T x \right) Q^{-1/2} \right] \right\} x. \tag{31}
 \end{aligned}$$

*Proof.* Since  $xBR^{-1}B^T - xA - A^T x$  is symmetric, it follows immediately from Lemma 1 that

$$\begin{aligned}
 dJ_R(x) &= \frac{1}{2} d \left\{ \text{tr} \log^2 \left[ Q^{-1/2} \left( xBR^{-1}B^T x - xA - A^T x \right) Q^{-1/2} \right] \right\} \\
 &= \frac{1}{2} \text{tr} \left\{ d \log^2 \left[ Q^{-1/2} \left( xBR^{-1}B^T x - xA - A^T x \right) Q^{-1/2} \right] \right\} \\
 &= \text{tr} \left\{ \log \left[ Q^{-1/2} \left( xBR^{-1}B^T x - xA - A^T x \right) Q^{-1/2} \right] \right. \\
 &\quad \left. \times d \log \left[ Q^{-1/2} \left( xBR^{-1}B^T x - xA - A^T x \right) Q^{-1/2} \right] \right\} \\
 &= \text{tr} \left\{ \log \left[ Q^{-1/2} \left( xBR^{-1}B^T x - xA - A^T x \right) Q^{-1/2} \right] \right. \\
 &\quad \left. \times \left[ Q^{-1/2} \left( xBR^{-1}B^T x - xA - A^T x \right) Q^{-1/2} \right]^{-1} \right. \\
 &\quad \left. \times Q^{-1/2} d \left( xBR^{-1}B^T x - xA - A^T x \right) Q^{-1/2} \right\} \\
 &= \text{tr} \left\{ \log \left[ Q^{-1/2} \left( xBR^{-1}B^T x - xA - A^T x \right) Q^{-1/2} \right] \right. \\
 &\quad \left. \times \left( xBR^{-1}B^T x - xA - A^T x \right)^{-1} \right. \\
 &\quad \left. \times \left( dxBR^{-1}B^T x - xBR^{-1}B^T dx - dxA - A^T dx \right) \right\} \\
 &= \text{tr} \left\{ \left[ \left( BR^{-1}B^T x - A \right) \right. \right. \\
 &\quad \times \log \left[ Q^{-1/2} \left( xBR^{-1}B^T x - xA - A^T x \right) Q^{-1/2} \right] \\
 &\quad \times \left( xBR^{-1}B^T x - xA - A^T x \right)^{-1} \\
 &\quad \left. - \log \left[ Q^{-1/2} \left( xBR^{-1}B^T x - xA - A^T x \right) Q^{-1/2} \right] \right. \\
 &\quad \times \left( xBR^{-1}B^T x - xA - A^T x \right)^{-1} \\
 &\quad \left. \times \left( xBR^{-1}B^T + A^T \right) \right] dx \right\}. \tag{32}
 \end{aligned}$$

Using the above Lemma 2, we have

$$\begin{aligned}
 \partial_x J_R(x) &= \left\{ \left( BR^{-1}B^T x - A \right) \right. \\
 &\quad \times \log \left[ Q^{-1/2} \left( xBR^{-1}B^T x - xA - A^T x \right) Q^{-1/2} \right] \\
 &\quad \times \left( xBR^{-1}B^T x - xA - A^T x \right)^{-1} \\
 &\quad \left. - \log \left[ Q^{-1/2} \left( xBR^{-1}B^T x - xA - A^T x \right) Q^{-1/2} \right] \right. \\
 &\quad \left. \times \left( xBR^{-1}B^T x - xA - A^T x \right)^{-1} \left( xBR^{-1}B^T + A^T \right) \right\}^T
 \end{aligned}$$

$$\begin{aligned}
&= (xBR^{-1}B^T x - xA - A^T x)^{-1} \\
&\quad \times \log [Q^{-1/2} (xBR^{-1}B^T x - xA - A^T x) Q^{-1/2}] \\
&\quad \times (xBR^{-1}B^T - A^T) \\
&\quad - (BR^{-1}B^T x + A) (xBR^{-1}B^T x - xA - A^T x)^{-1} \\
&\quad \times \log [Q^{-1/2} (xBR^{-1}B^T x - xA - A^T x) Q^{-1/2}].
\end{aligned} \tag{33}$$

Combining (23) and (33), (31) is established. This completes the proof of Theorem 4.  $\square$

**3.3. The Extended Hamiltonian Algorithm on PD( $n$ ).** The Euclidean gradient algorithm (EGA) and the Riemannian gradient algorithm (RGA) are first-order learning algorithms and inherit some drawbacks that are well known about gradient-based optimization like, for instance, the slow-learning phenomenon in presence of plateau in the error surface. In order to overcome this difficulty, Fiori generalized EGA and RGA and proposed the extended Hamiltonian algorithm on manifold. In particular, on manifold PD( $n$ ), the extended Hamiltonian algorithm can be expressed by [18]

$$\begin{aligned}
\dot{x} &= v, \\
\dot{v} &= vx^{-1}v - \nabla_x V - \mu v,
\end{aligned} \tag{34}$$

where  $x \in \text{PD}(n)$ ,  $v$  denotes the instantaneous learning velocity,  $V : \text{PD}(n) \rightarrow \mathbb{R}$  denotes a cost function,  $\nabla_x V$  denotes the Riemannian gradient of  $V$ , and the constant  $\mu > 0$  denotes a viscosity term. The function  $K_x : T_x \text{PD}(n) \times T_x \text{PD}(n) \rightarrow \mathbb{R}$  denotes the kinetic energy of the particle in a point  $x$  and has the corresponding expression  $K_x = (1/2) \text{tr}[(vx^{-1})^2]$ .

System (34) may be implemented by [19]

$$\begin{aligned}
x_{k+1} &= x_k^{1/2} \exp(\eta x_k^{-1/2} v_k x_k^{-1/2}) x_k^{1/2}, \\
v_{k+1} &= \eta [v_k x_k^{-1} v_k - \nabla_{x_k} V] + (1 - \eta\mu) v_k,
\end{aligned} \tag{35}$$

where  $\eta$  is a small positive number known as the learning rate. The effectiveness of the algorithm is ensured if and only if

$$\sqrt{2\lambda_{\max}} < \mu < \frac{1}{\eta}, \tag{36}$$

where  $\lambda_{\max}$  denotes the largest eigenvalue of the Hessian matrix of the cost function  $V(x)$ . See [19] for more details.

Since the Riemannian distance is the shortest one on manifold PD( $n$ ), we take the Riemannian distance (22) as the cost function  $V(x)$  of (18); that is,

$$\begin{aligned}
V(x) &= \frac{1}{2} d_R^2(Q, xBR^{-1}B^T x - xA - A^T x) \\
&= \frac{1}{2} \left\| \log(Q^{-1/2} (xBR^{-1}B^T x - xA - A^T x) Q^{-1/2}) \right\|_F^2.
\end{aligned} \tag{37}$$

In order to apply (35) to solve (18), we plug (31) into (34) and get the final extended Hamiltonian algorithm on manifold PD( $n$ ) as follows:

$$\begin{aligned}
\dot{x} &= v, \\
a &\stackrel{\text{def}}{=} (xBR^{-1}B^T x - xA - A^T x)^{-1} \\
&\quad \times \log [Q^{-1/2} (xBR^{-1}B^T x - xA - A^T x) Q^{-1/2}],
\end{aligned} \tag{38}$$

$$b \stackrel{\text{def}}{=} x [a (xBR^{-1}B^T - A^T) - (BR^{-1}B^T x + A) a] x,$$

$$\dot{v} = vx^{-1}v - b - \mu v,$$

which is implemented by

$$\begin{aligned}
x_{k+1} &= x_k^{1/2} \exp(\eta x_k^{-1/2} v_k x_k^{-1/2}) x_k^{1/2}, \\
a_k &\stackrel{\text{def}}{=} (x_k BR^{-1} B^T x_k - x_k A - A^T x_k)^{-1} \\
&\quad \times \log [Q^{-1/2} (x_k BR^{-1} B^T x_k - x_k A - A^T x_k) Q^{-1/2}], \\
b_k &\stackrel{\text{def}}{=} x_k [a_k (x_k BR^{-1} B^T - A^T) - (BR^{-1} B^T x_k + A) a_k] x_k, \\
v_{k+1} &= \eta (v_k x_k^{-1} v_k - b_k) + (1 - \eta\mu) v_k,
\end{aligned} \tag{39}$$

where  $\eta > 0$  denotes the learning rate and the constant  $\mu$  denotes a viscosity term.

Now, based on the above discussion, we give the iterative formula of the EHA for the numerical solution of (18).

**Algorithm 5 (EHA).** For any  $x$  belonging to the considered manifold PD( $n$ ), the extended Hamiltonian algorithm is given by the following.

- (1) Set  $x_0$  as an initial input matrix  $x$  and  $v_0$  as an initial direction of  $v$ , and then choose a desired tolerance  $\varepsilon > 0$ .
- (2) Compute  $V(x_k)$ .
- (3) If  $V(x_k) < \varepsilon$  then stop.
- (4) Update  $x$  and  $v$  by (39) and return to step (2).

## 4. Simulations

In this section, we give two computer simulations to demonstrate the effectiveness and performance of the proposed algorithm.

*Example 1.* First consider a second-order algebraic Riccati equation. In the present experiment,  $\eta = 0.01$ ,  $\mu = 3$ , and  $\varepsilon = 10^{-8}$ . Moreover,  $A, B, Q, R$ , and  $D$  used in this simulation are as follows:

$$\begin{aligned}
A &= \begin{pmatrix} 0 & 0 \\ 1 & 0 \end{pmatrix}, & B &= \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \\
Q &= \begin{pmatrix} 2 & 0 \\ 0 & 2 \end{pmatrix}, & R &= \frac{1}{2}, & D &= \begin{pmatrix} \sqrt{2} & 0 \\ 0 & \sqrt{2} \end{pmatrix},
\end{aligned} \tag{40}$$

which satisfy the unique definite solution condition.

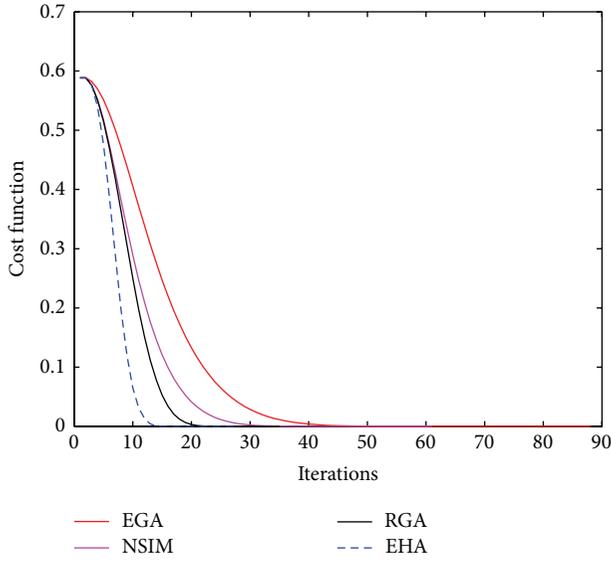


FIGURE 1: Comparison of convergence speeds of the EGA, the RGA, the NSIM, and the EHA.

To study the performance differences among the EHA, the EGA, the RGA, and the NSIM, these algorithms are, respectively, applied to solve (18). In this paper, the optimal step size  $\eta$ , which corresponds to the best convergence speed in each algorithm, is obtained by numerical experiments. For instance, it can be found that the iterations will reduce gradually as the step size changes from 0.01 to 0.1, while the algorithm will be divergent once the step size is bigger than 0.1. Thus this optimal step size  $\eta$  is convenient to be obtained experimentally. Based on (36), the constant  $\mu > 0$  depends on the selection of initial value. That is to say, different initial values will be corresponding to different  $\mu$ ; hence the best  $\mu$  is also obtained by the experiment.

As Figure 1 shows, the behavior of the cost function is shown. Clearly, in the early stages of learning, the EHA decreases much faster than the EGA, the RGA, and the NSIM with the same learning step size. The result shows that the EHA has the fastest convergence speed among four algorithms and needs 28 iterations to obtain the numerical solution of the ARE as follows:

$$\begin{pmatrix} 1.4142136 & 0.9999999 \\ 1.0000000 & 2.8284271 \end{pmatrix}. \quad (41)$$

However, RGA realizes the given error through 36 iterations and the slowest one among them is the EGA with 88 steps. And Figure 2 shows that the kinetic energy function finally converges to 0.

*Example 2.* A third-order ARE is considered following the similar procedure as Example 1. In this experiment,  $\eta = 0.01$ ,  $\mu = 2.5$ , and  $\varepsilon = 10^{-8}$ , and

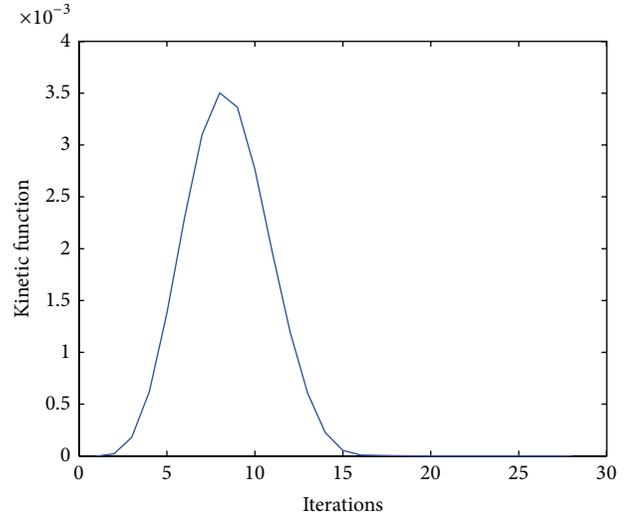


FIGURE 2: Kinetic energy in the EHA.

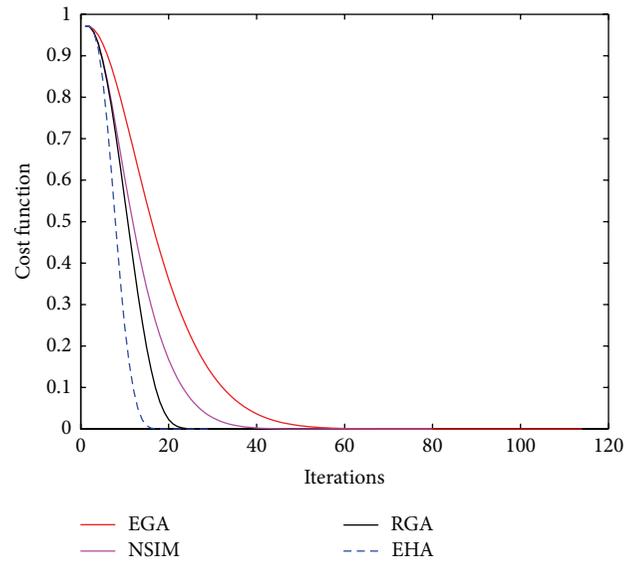


FIGURE 3: Comparison of convergence speeds of the EGA, the RGA, the NSIM, and the EHA.

$$\begin{aligned} A &= \begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix}, & B &= \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}, \\ Q &= \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}, & R &= 1, & D &= \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}. \end{aligned} \quad (42)$$

As Figure 3 shows, the convergence speed of the EHA is still the fastest one among four algorithms. Finally, we get the numerical solution of the ARE as follows:

$$\begin{pmatrix} 3.6502815 & 6.1622776 & 5.0644951 \\ 6.1622776 & 13.7792717 & 12.3245553 \\ 5.0644951 & 12.3245553 & 12.3650582 \end{pmatrix}. \quad (43)$$

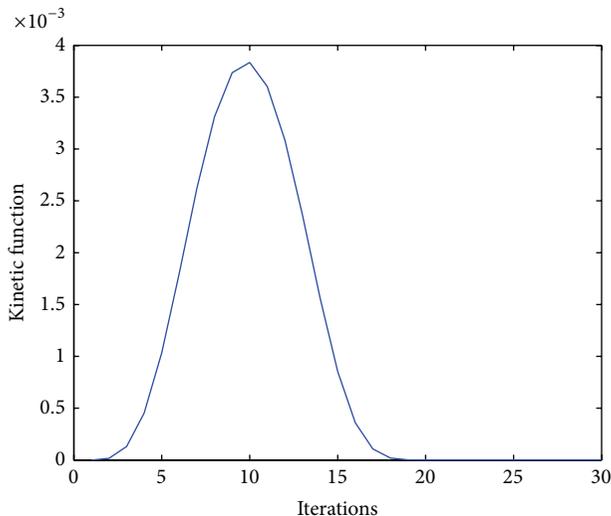


FIGURE 4: Kinetic energy in the EHA.

Moreover, the kinetic energy function also converges to 0 in the Figure 4.

## 5. Summary

In this paper, a second-order learning method called the extended Hamiltonian algorithm is recalled from literature and applied to solve the numerical solution for the algebraic Riccati equation. And, the convergence speed of the provided algorithm is compared with the EGA, the RGA, and the NSIM using two simulation examples. It is shown that the convergence speed of the extended Hamiltonian algorithm is the fastest one among these algorithms.

## Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

## Acknowledgments

This subject is supported by the National Natural Science Foundations of China (nos. 61179031 and 10932002).

## References

- [1] K. Zhou, J. Doyle, and K. Glover, *Robust and Optimal Control*, Prentice Hall, Upper Saddle River, NJ, USA, 1996.
- [2] Z. Lin, "Global control of linear systems with saturating actuators," *Automatica*, vol. 34, no. 7, pp. 897–905, 1998.
- [3] D.-Z. Zheng, "Optimization of linear-quadratic regulator systems in the presence of parameter Perturbations," *IEEE Transactions on Automatic Control*, vol. 31, no. 7, pp. 667–670, 1986.
- [4] M. L. Ni and H. X. Wu, "A Riccati equation approach to the design of linear robust controllers," *Automatica*, vol. 29, no. 6, pp. 1603–1605, 1993.
- [5] T. Kaneko, S. Fiori, and T. Tanaka, "Empirical arithmetic averaging over the compact Stiefel manifold," *IEEE Transactions on Signal Processing*, vol. 61, no. 4, pp. 883–894, 2013.
- [6] H.-P. Liu, F.-C. Sun, K.-Z. He, and Z.-Q. Sun, "Simultaneous stabilization for singularly perturbed systems via iterative linear matrix inequalities," *Acta Automatica Sinica*, vol. 30, no. 1, pp. 1–7, 2004.
- [7] V. Sima, *Algorithms for Linear-Quadratic Optimization*, vol. 200 of *Monographs and Textbooks in Pure and Applied Mathematics*, Marcel Dekker, New York, NY, USA, 1996.
- [8] P. Benner, J.-R. Li, and T. Penzl, "Numerical solution of large-scale Lyapunov equations, Riccati equations, and linear-quadratic optimal control problems," *Numerical Linear Algebra with Applications*, vol. 15, no. 9, pp. 755–777, 2008.
- [9] H. G. Xu and L. Z. Lu, "Properties of a quadratic matrix equation and the solution of the continuous-time algebraic Riccati equation," *Linear Algebra and Its Applications*, vol. 222, pp. 127–145, 1995.
- [10] H. H. Choi and T.-Y. Kuc, "Lower matrix bounds for the continuous algebraic Riccati and Lyapunov matrix equations," *Automatica*, vol. 38, no. 8, pp. 1147–1152, 2002.
- [11] R. Davies, P. Shi, and R. Wiltshire, "New lower solution bounds of the continuous algebraic Riccati matrix equation," *Linear Algebra and Its Applications*, vol. 427, no. 2-3, pp. 242–255, 2007.
- [12] E. K.-W. Chu, H.-Y. Fan, and W.-W. Lin, "A structure-preserving doubling algorithm for continuous-time algebraic Riccati equations," *Linear Algebra and Its Applications*, vol. 396, pp. 55–80, 2005.
- [13] A. J. Laub, "A Schur method for solving algebraic Riccati equations," *IEEE Transactions on Automatic Control*, vol. 24, no. 6, pp. 913–921, 1979.
- [14] Y. Lin and V. Simoncini, "A new subspace iteration method for the algebraic Riccati equation," preprint, <http://arxiv.org/abs/1307.3843>.
- [15] X. Duan, *Information geometry of matrix Lie groups and applications [Ph.D. thesis]*, Beijing Institute of Technology, 2013, (Chinese).
- [16] D. Rumelhart and J. McClelland, *Parallel Distributed Processing*, MIT Press, 1986.
- [17] N. Qian, "On the momentum term in gradient descent learning algorithms," *Neural Networks*, vol. 12, no. 1, pp. 145–151, 1999.
- [18] S. Fiori, "Extended Hamiltonian learning on Riemannian manifolds: theoretical aspects," *IEEE Transactions on Neural Networks*, vol. 22, no. 5, pp. 687–700, 2011.
- [19] S. Fiori, "Extended Hamiltonian learning on Riemannian manifolds: numerical aspects," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 23, no. 1, pp. 7–21, 2012.
- [20] M. Moakher, "A differential geometric approach to the geometric mean of symmetric positive-definite matrices," *SIAM Journal on Matrix Analysis and Applications*, vol. 26, no. 3, pp. 735–747, 2005.
- [21] M. Moakher, "On the averaging of symmetric positive-definite tensors," *Journal of Elasticity*, vol. 82, no. 3, pp. 273–296, 2006.
- [22] A. Schwartzman, *Random ellipsoids and false discovery rates: statistics for diffusion tensor imaging data [Ph.D. thesis]*, Stanford University, 2006.
- [23] J. Jost, *Riemannian Geometry and Geometric Analysis*, Universitext, Springer, Berlin, Germany, Third edition, 2002.
- [24] R. E. Kalman, "Contributions to the theory of optimal control," *Boletin de la Sociedad Matematica Mexicana*, vol. 5, no. 2, pp. 102–119, 1960.

- [25] R. Kalman, "When is a linear control system optimal?" *Journal of Basic Engineering*, vol. 86, no. 1, pp. 51–60, 1964.
- [26] S. Fiori, "Solving minimal-distance problems over the manifold of real-symplectic matrices," *SIAM Journal on Matrix Analysis and Applications*, vol. 32, no. 3, pp. 938–968, 2011.
- [27] R. Gregório and P. Oliveira, "A proximal technique for computing the Karcher mean of symmetric positive definite matrices," *Optimization Online*, 2013.
- [28] X. Zhang, *Matrix Analysis and Application*, Springer, New York, NY, USA, 2004.