©2010 International Journal of Computer Applications (0975 - 8887)
Volume 1 – No. 29

Solution of matrix Riccati differential equation for nonlinear singular system using neural networks

J. Abdul Samath
Department of Computer Applications
Sri Ramakrishna Institute of Technology
Coimbatore - 641 010, Tamilnadu, India

N. Selvaraju
Department of Computer Applications
Sri Ramakrishna Institute of Technology
Coimbatore - 641 010, Tamilnadu, India

ABSTRACT
In this paper, the solution of the matrix Riccati differential equation(MRDE) for nonlinear singular system is obtained using neural networks. The goal is to provide optimal control with reduced calculus effort by comparing the solutions of the MRDE obtained from well known traditional Runge Kutta(RK)method and nontraditional neural network method. Accuracy of the neural solution to the problem is qualitatively better. The advantage of the proposed approach is that, once the network is trained, it allows instantaneous evaluation of solution at any desired number of points spending negligible computing time and memory. The computation time of the proposed method is shorter than the traditional RK method. An illustrative numerical example is presented for the proposed method.

Key words:
Matrix Riccati differential equation, Nonlinear, Optimal control, Singular system, Runge
Kutta method and Neural networks.

1. INTRODUCTION
Neural networks or simply neural nets are computing systems, which can be trained to learn a complex relationship between two or many variables or data sets. Having the structures similar to their biological counterparts, neural networks are representational and computational models processing information in a parallel distributed fashion composed of interconnecting simple processing nodes [36]. Neural net techniques have been successfully applied in various fields such as function approximation, signal processing and adaptive (or) learning control for nonlinear systems. Using neural networks, a variety of off-line learning control algorithms have been developed for nonlinear systems [17, 25]. A variety of numerical algorithms have been developed for solving the algebraic Riccati equation. In recent years, neural network problems have attracted considerable attention of many researchers for numerical aspects for algebraic Riccati equations see [13, 14, 37, 3].

Singular systems contain a mixture of algebraic and differential equations. In that sense, the algebraic equations represent the constraints to the solution of the differential part. These systems are also known as degenerate, descriptor or semi-state and generalized state-space systems. The complex nature of singular system causes many difficulties in the analytical and numerical treatment of such systems, particularly when there is a need for their control. The system arises naturally as a linear approximation of system models or linear system models in many applications such as electrical networks, aircraft dynamics, neutral delay systems, chemical, thermal and diffusion processes, large-scale systems, robotics, biology, etc., see [6, 7, 11, 19]. Most of the research on nonlinear singular systems has focused primarily on issues related to solvability and numerical solutions for such systems [5, 9]. The literature on feedback control of nonlinear singular systems is sparse. The feedback stabilization problem for nonlinear singular systems is addressed by McClamroch [22].

In this paper, we make use of a result that generalizes the LQ theory to nonlinear systems to provide a nonlinear design method [4, 21]. This nonlinear quadratic (NLQ) method applies to systems having a broad class of nonlinear dynamics with state dependent weighting matrices. In brief, it turns out that the infinite time horizon LQ regulator problem, when solved afresh at every point on the state trajectory, leads to an asymptotically optimal control policy. The LQ regulator problem converges to the optimal control close to the origin.

As the theory of optimal control of linear systems with quadratic performance criteria is well developed, the results are most complete and close to use in many practical designing problems. The theory of the quadratic cost control problem has been treated as a more interesting problem and the optimal feedback with minimum cost control has been characterized by the solution of a Riccati equation. Da Prato and Ichikawa [12] showed that the optimal feedback control and the minimum cost are characterized by the solution of a Riccati equation. Solving the MRDE is the central issue in optimal control theory. The needs for solving such equations often arise in analysis and synthesis such as linear-quadratic optimal control systems, robust control systems with $H2$ and $H\infty$ - control [38] performance criteria, stochastic filtering and control systems, model reduction, differential games etc. One of the most intensely studied nonlinear matrix equations arising in mathematics and engineering is the Riccati equation. This equation, in one form or another, has an important role in optimal control problems, multivariable and large scale systems, scattering theory, estimation, detection, transportation and radiative transfer [15]. The solution of this equation is difficult to obtain from two points of view. One is nonlinear and the other is in matrix form. Most general methods to solve MRDE with a terminal boundary condition are obtained on transforming MRDE into an equivalent linear differential Hamiltonian system [16]. By using this approach, the solution of MRDE is obtained by partitioning the transition matrix of the associated Hamiltonian system [35]. Another class of method is based on transforming MRDE into a linear matrix differential equation and then solving MRDE analytically or computationally [20, 31, 32]. However, the method in[30] is restricted for cases when certain coefficients of MRDE are nonsingular. In [16], an analytic procedure of solving the MRDE of the linear quadratic control problem for homing missile systems is presented. The solution K(t) of MRDE is obtained by using $K(t) = p(t) / f(t)$, where $f(t)$ and $p(t)$ are solutions of certain first order ordinary linear differential equations. However, the given technique is restricted to single input.

There is rarely an analytical solution although several numerical computation approaches have been proposed (for example, see [28]). A variety of numerical algorithms have been developed for solving the algebraic Riccati equation. The approximating sequence of Riccati equations feedback algorithm for nonlinear optimal control provides outstanding performance in many practical applications, in particular, nonlinear solitary wave motion [1] and optimal maneuvering of super-tankers at high speeds [8]. These makes up for seeking an efficient method for solving the nonlinear optimal control problems. However, many existing numerical algorithms for solving MRDE for nonlinear optimal control optimality have not been proved. Using neural networks, a variety of off-line learning control algorithms have been developed for nonlinear systems [24, 29]. Neural networks have been used to control nonlinear systems (see [10, 27, 33, 34]). It has been shown that they can effectively extend adaptive control techniques to nonlinearly parameterized systems. In Miller et. al [23] first proposed using neural networks to find optimal control laws using the HJB equation. Parisini and Zoppoli [29] used neural networks to derive optimal control laws for discrete-time stochastic nonlinear system. The status of neural network control appears recently in Narendra and Lewis [24].

Although parallel algorithms can compute the solutions faster than sequential algorithms, there is no report on neural network solutions for MRDE. Recently solution of MRDE for linear singular system is obtained by training feedforward neural network using Levenberg-Marquardt algorithm [2].

This paper is focused upon the implementation of neurocomputing approach for solving MRDE from assumed trail solution by training the feedforward neural network till the error function become zero. The accuracy of MRDE in this approach is better than all the existing numerical methods. In this method the numerical solution is more or less equivalent to the exact solution. The structured neural network architecture is trained to prove the efficiency of solutions in shorter computation time. An example is given, which illustrates the advantage of the fast and accurate solutions of MRDE using neural networks.

This paper is organized as follows. In section 2, the statement of the problem is given. In section 3, solution of the MRDE is presented. In section 4, numerical example is discussed. The final conclusion section demonstrates the efficiency of the method.

2. Statement of the Problem

Consider the nonlinear dynamical singular system that can be expressed in the form:

$$E\dot{x}(t)=f(x,u)=A(x)x(t)+B(x)u(t), f(0,0)= \quad t \in [0, t_f] \quad (1)$$

where the matrix E is possibly singular, $x(t) \in R^n$ is a generalized state space vector, $u(t) \in R^m$ is a control variable. Then at each point

$\bar{x}$, on the state trajectory, the nonlinear system (1) can be defined as a linear system by

$$E\dot{x}(t) = A(\bar{x})(t) + B(\bar{x})u(t), \quad x(0) = x_0 \qquad (2)$$

where $A(\bar{x}) \in R^{nxn}$ and $B(\bar{x}) \in R^{nxm}$ are known constant coefficient matrices associated with $x(t)$ and $u(t)$ respectively, x_0 is given initial state vector and $m \le n$. If all state variables are measurable, then a linear state feedback control law

$$u(t)= -R^{-1}B^T \lambda(t)$$

can be obtained to the system described by equation (2), where

$$\lambda(t)=K(t)Ex(t),$$

$K(t)=(k_{ij}) \in R^{nxn}$ is a symmetric matrix and is the solution of MRDE.

In order to minimize both state and control signals of the feedback control system, a quadratic performance index is usually minimized:

$$\min J = \frac{1}{2} x^T (t_f) E^T SEx(t_f) + \frac{1}{2}\int_0^{t_f} [x^T (t)Qx(t) + u^T (t) Ru(t)]\,dt$$

where the superscript T denotes the transpose operator, $S \in R^{nxn}$ and $Q \in R^{nxn}$ are symmetric and positive definite (or semidefinite) weighting matrices for $x(t)$, $R \in R^{mxm}$ is a symmetric and positive definite weighting matrix for $u(t)$.

It is well known in the control literature that to minimize J is equivalent to minimize the Hamiltonian equation

$$H(x(t),u(t),\lambda(t))=\frac{1}{2}x^T(t)Qx(t)+\frac{1}{2}u^T(t)Ru(t)+ \lambda^T(t) [A(\bar{x})(t)+B(\bar{x})u(t)]$$

The Relative MRDE is

$$E^T\dot{K}(t)E+E^TK(t)A(\bar{x})+A(\bar{x})^TK(t)E+Q-E^TK(t)BR^{-1}B^TK(t)E=0 \quad (3)$$

with terminal condition(TC) $K(t_f) = E^TSE$

After substituting the appropriate matrices in the above equation, it is transformed into a system of differential equations. Therefore solving MRDE is equivalent to solving the system of nonlinear differential equations.

3. Solution of MRDE

Consider the system of differential equation for (3)

$$\dot{k}_{ij}(t) = \emptyset_{ij}(k_{ij}(t)),\ (k_{ij})(t_f)=A_{ij}\ (i, j = 1,2, \ldots .n) \quad (4)$$

3.1. Runge Kutta Solution.

RK algorithms have always been considered as the best tool for the numerical integration of ordinary differential equations(*ODEs*). The system (4) contains n^2 first order ODEs with n^2 variables, RK method is explained for a system of two first order ODEs with two variables.

$$K_{11}(i+1) = k_{11}(i)+\frac{1}{6}(k1 + 2k2 + 2k3 +k4)$$

$$K_{12}(i+1) = k_{12}(i)+\frac{1}{6}(l1 + 2l2 + 2l3 +l4)$$

where

$$k1 = h * \emptyset_{11} (k_{11},k_{12})$$

$$l1 = h* \emptyset_{12} (k_{11},k_{12})$$

$$k2 = h* \emptyset_{11}(k_{11} + \frac{k1}{2}, k_{12} + \frac{l1}{2})$$

$$l2 = h* \emptyset_{12}(k_{11} + \frac{k1}{2}, k_{12} + \frac{l1}{2})$$

$$k3 = h* \emptyset_{11}(k_{11} + \frac{k2}{2}, k_{12} + \frac{l2}{2})$$

$$l3 = h* \emptyset_{12}(k_{11} + \frac{k2}{2}, k_{12} + \frac{l2}{2})$$

$$k4 = h *\emptyset_{11}(k_{11} + k3, k_{12}+l3)$$

$$l4 = h*\emptyset_{12}(k_{11} + k3, k_{12}+l3)$$

In the similar way, the original system (4) can be solved for n^2 first order ODE's.

3.2 Neural Network Solution.

In this approach, new feedforward neural network is used to transfer the trail solution of eqution (4) to the neural network solution of (4). The trail solution is expressed as the difference of two terms as below (see [18]).

$$(k_{ij})_a(t) = A_{ij} - tN_{ij}(t, w_{ij}) \quad (5)$$

The first term satisfies the TCs and contains no adjustable parameters. The second term employs a feedforward neural network and parameters *wij* correspond to the weights of the neural architecture.

Consider a multilayer perceptron with n input units, one hidden layer with n sigmoidal units and a linear output unit. The extension to the case of more than one hidden layer can be obtained accordingly. For a given input vector, the output of the network is $N_{ij} = \sum_{i=1}^{n}(v_i)\sigma(z_i)$ where $z_i=\sum_{j=1}^{n}(w_{ij})t_j + u_i$, w_{ij} denotes the weight from the input unit j to the hidden unit i, v_i denotes the weight from the hidden unit i to the output, u_i denotes the bias of the hidden unit i and $\sigma(z)$ is the sigmoid transfer function.

The error quantity to be minimized is given by

$$E_r=\sum_{i,j=1}^{n}((\dot{k}_{ij})_a - \emptyset_{ij}(t, (k_{ij})_a))^2 \quad (6)$$

The neural network is trained till the error function (6) becomes zero. Whenever E_r becomes zero, the trail solution (5) becomes the neural network solution of the equation (4).

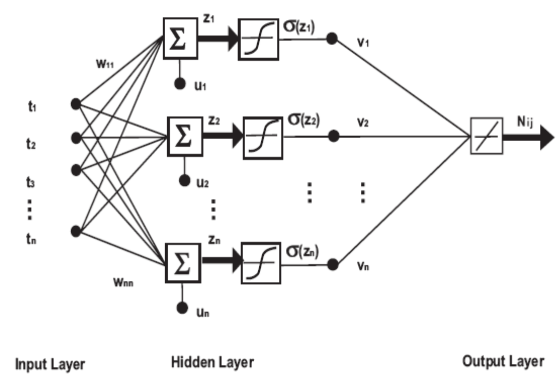

FIGURE 1. Neural Network Architecture

3.3. Structure of the FFNN.

The architecture consists of n input units, one hidden layer with n sigmoidal units and a linear output. Each neuron produces its output by computing the inner product of its input and its appropriate weight vector. During the training, the weights and biases of the network are iteratively adjusted by Nguyen and Widrow rule [26]. The neural network architecture is given in the Fig. 1 for computing *Nij* . The neural network algorithm was implemented in MATLAB on a PC, CPU 1.7 GHz for the neuro computing approach to solve MRDE (3) for the nonlinear system (1).

Neural network Algorithm

Step 1. Feed the input vector t_j.

Step 2. Initialize randomized weight matrix w_{ij} and bias u_i

Step 3. Compute $z_i = \sum_{j=1}^{n} w_{ij}t_j + u_i$

Step 4. Pass z_i into n sigmoidal functions.

Step 5. Initialize the weight vector v_i from the hidden unit to output unit.

Step 6. Calculate $N_{ij} = \sum_{i=1}^{n}(v_i)\sigma(z_i)$

Step 7. Compute purelin function (N_{ij})

Step 8. Repeat the neural network training till the following error function

$$E_r = \sum_{i,j=1}^{n}((\dot{k}_{ij})_a - \emptyset_{ij}(t, (k_{ij})_a))^2 = 0$$

The solution of MRDE can be obtained using above two methods. In the similar way, we can find out the solution of MRDE at each value of $\bar{x}$ and then resultant optimal control can be found out for the nonlinear singular system.

4. Numerical Examples

Example 1

Consider the optimal control problem:

Minimize

$$J = \frac{1}{2}x^T(t_f)E^TSEx(t_f) + \frac{1}{2}\int_0^{tf}[x^T(t)Qx(t)+u^T(t)Ru(t)]\,dt$$

subject to the nonlinear singular system

$$E\dot{x}(t) = f(x,u)=A(x)x+B(x)u,\ f(0,0)=0$$

where

$$S = \begin{bmatrix}1 & 0\\ 0 & 0\end{bmatrix},\ E = \begin{bmatrix}1 & 0\\ 0 & 0\end{bmatrix},\ A(x) = \begin{bmatrix}x & 2\\ 0 & 4\end{bmatrix},$$

$$B(x) = \begin{bmatrix}0 & 0\\ x & 0\end{bmatrix},\ R=1,\ Q=\begin{bmatrix}1 & 0\\ 0 & 0\end{bmatrix}.$$

The numerical implementation could be adapted by taking $x = \bar{x}=1$ $t_f=2$ for solving the related MRDE of the above nonlinear singular system. The appropriate matrices are substituted in equation (3), the MRDE is transformed into system of differential equation in k_{11} and k_{12}. In this problem, the value of k_{22} of the symmetric matrix $K(t)$ is free and let $k_{22} = 0$.Then the optimal Control of the system can be found out by the solution of MRDE. The numerical solutions of MRDE are calculated and displayed in the table 1 using the RK-method and the neural network approach. A multilayer perception having one hidden layer with 10 hidden units and one linear output unit is used. The sigmoid activation function of each hidden units is $\sigma(t) = 1/1+e^{-z}$.

t	Neural network solution		Runge Kutta solution	
	K_{11}	K_{12}	K_{11}	K_{12}
0.0	10.3913	-4.5357	9.9216	-4.4608
0.2	10.0458	-4.3845	9.6785	-4.3393
0.4	9.5260	-4.1551	9.2915	-4.1457
0.6	8.7785	-3.8210	8.6988	-3.8494
0.8	7.7716	-3.3626	7.8448	-3.4224
1.0	6.5287	-2.7832	6.7150	-2.8575
1.2	5.1508	-2.1223	5.3785	-2.1892
1.4	3.7952	-1.4518	3.9916	-1.4958
1.6	2.6099	-0.8480	2.7350	-0.8675
1.8	1.6764	-0.3603	1.7288	-0.3644
2.0	1.0000	0.0000	1.0000	0.0000

TABLE 1. Solutions of MRDE when $\bar{x} = 1$

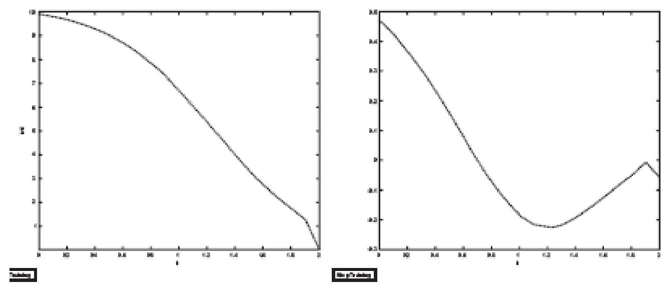

FIGURE 2. Solution and error curve for k_{11} when $\bar{x} = 1$

Now taking $\bar{x}$ =2, the numerical solution of the MRDE is obtained in RK and neural network methods and displayed in the table 2. In the similar way, we can find out the solution for MRDE at each value of $\bar{x}$ and then resultant optimal control can be found out for the nonlinear singular system in a reduced calculus effort.

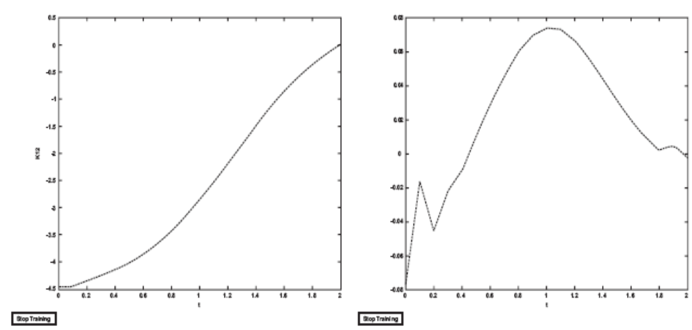

FIGURE 3. Solution and error curve for k_{12} when $\bar{x} = 1$

t	Neural network solution K_{11}	Neural network solution K_{12}	Runge Kutta solution K_{11}	Runge Kutta solution K_{12}
0.0	6.641875	-2.624978	5.999812	-2.499906
0.2	6.639428	-2.624060	5.999380	-2.499690
0.4	6.632725	-2.621544	5.997947	-2.498974
0.6	6.614400	-2.614659	5.993204	-2.496602
0.8	6.564597	-2.595909	5.977546	-2.488773
1.0	6.431401	-2.545478	5.926244	-2.463122
1.2	6.090066	-2.414296	5.762341	-2.381171
1.4	5.304865	-2.101215	5.278236	-2.139118
1.6	3.885856	-1.489596	4.127262	-1.563631
1.8	2.202871	-0.668793	2.393931	-0.696965
2.0	1.0000	0.0000	1.000000	0.000000

TABLE 2. Solutions of MRDE when $\bar{x} = 2$

4.1. Solution curves using Neural networks.

The solution of MRDE and the error between the solution by neural network and traditional RK method is displayed in figures 2, 3, 4 and 5. The numerical values of the required solution are listed in the Tables 1 and 2. The computation time for neural network solution is 1.6330 sec. whereas the RK method is 2.1930 sec. Hence the neural solution is faster than RK method. The numerical implementation could be adapted by taking $x = -1$ and $tf = 2$ for solving the related MRDE of the above nonlinear singular system. The appropriate matrices are substituted in equation (3), the MRDE is transformed into system of differential equation in $k11$ and $k12$. In this problem, the value of $k22$ of the symmetric matrix $K(t)$ is free and let $k22 = 0$. The optimal control of the system can be found out by the solution of MRDE. The numerical solution of MRDE are calculated and displayed in the table 3 using the RK method and the neural network approach.

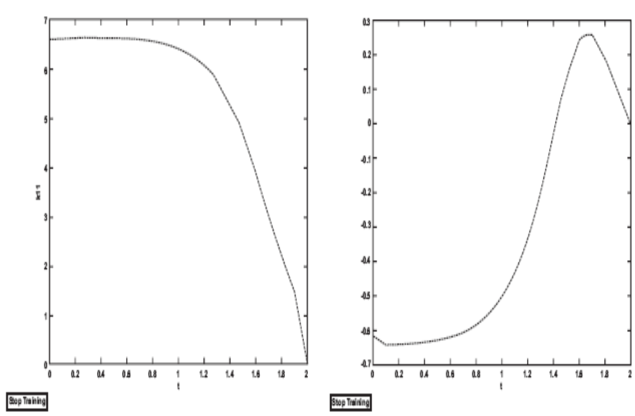

FIGURE 4. Solution and error curve for k_{11} when $\bar{x} = 2$

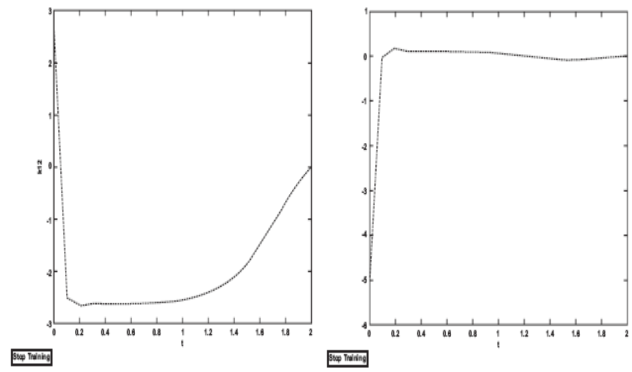

FIGURE 5. Solution and error curve for k_{12} when $\bar{x} = 2$

Example 2

Consider the optimal control problem:

Minimize

$$J = \frac{1}{2} x^T (t_f) E^T S E x(t_f) + \frac{1}{2} \int_0^{t_f} [x^T(t) Q x(t) + u^T(t) R u(t)]\, dt$$

subject to the nonlinear singular system

$$E\dot{x}(t) = f(x,u) = A(x)x + B(x)u,\ f(0,0) = 0$$

where

$S = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}$, $E = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}$, $A(x) = \begin{bmatrix} x & 1 \\ 0 & -3 \end{bmatrix}$, $B(x) = \begin{bmatrix} 0 & 0 \\ 1 & 0 \end{bmatrix}$,

$R = 1$, $Q = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}$

t	Neural network solution K_{11}	Neural network solution K_{12}	Runge Kutta solution K_{11}	Runge Kutta solution K_{12}
0.0	0.495550	-0.154309	0.495550	-0.154309
0.2	0.499580	-0.153072	0.502069	-0.165977
0.4	0.505688	-0.151199	0.509771	-0.163410
0.6	0.514947	-0.148359	0.521012	-0.159663
0.8	0.528990	-0.144053	0.537430	-0.154190
1.0	0.550303	-0.137519	0.561425	-0.146192
1.2	0.582686	-0.127593	0.596534	-0.134489
1.4	0.631974	-0.112492	0.647985	-0.117338
1.6	0.707185	-0.089466	0.723562	-0.092146
1.8	0.822408	-0.054227	0.834961	-0.055013
2.0	1.0000	0.0000	1.0000	0.0000

TABLE 3. Solutions of MRDE when $\bar{x} = -1$

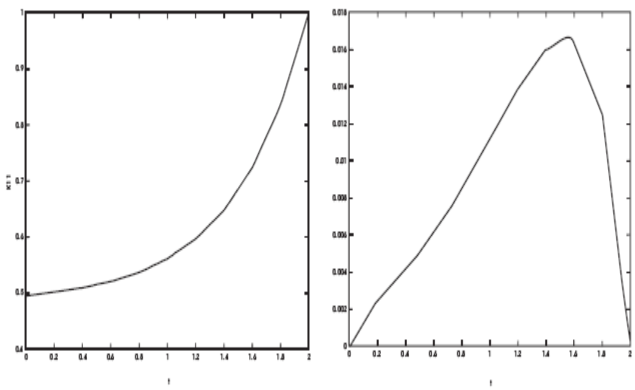

FIGURE 6. Solution and error curve for k_{11} when $\bar{x} = -1$

	Neural network solution		Runge Kutta solution	
t	K_{11}	K_{12}	K_{11}	K_{12}
0.0	0.239328	-0.207060	0.234042	-0.255319
0.2	0.239405	-0.207039	0.234450	-0.255183
0.4	0.239608	-0.206983	0.235326	-0.254891
0.6	0.240140	-0.206838	0.237211	-0.254263
0.8	0.241540	-0.206455	0.241267	-0.252911
1.0	0.245218	-0.205450	0.249991	-0.250003
1.2	0.254884	-0.202808	0.268768	-0.243744
1.4	0.280306	-0.195862	0.309220	-0.230260
1.6	0.347286	-0.177571	0.396538	-0.201154
1.8	0.524603	-0.129209	0.585828	-0.138057
2.0	1.0000	0.0000	1.0000	0.0000

TABLE 4. Solutions of MRDE when $\bar{x} = -2$

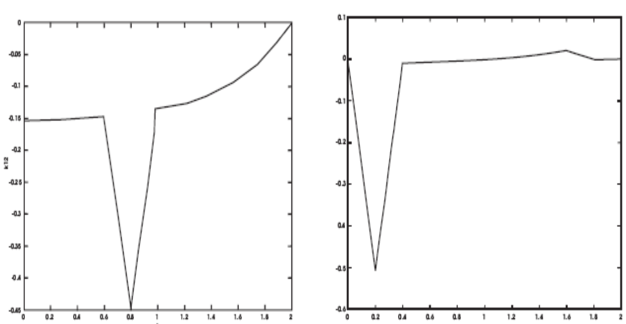

FIGURE 7. Solution and error curve for k_{12} when $\bar{x} = -1$

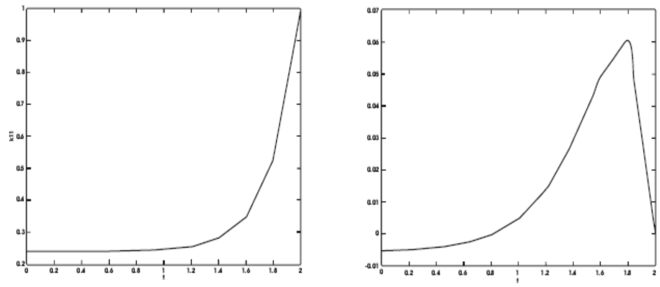

FIGURE 8. Solution and error curve for k_{11} when $\bar{x} = -2$

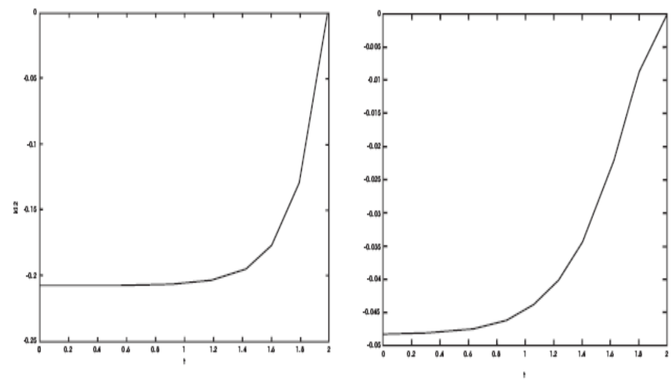

FIGURE 9. Solution and error curve for k_{12} when $\bar{x} = -2$

The solution of MRDE and the error between the solution by neural network and traditional RK method is displayed in figures 6, 7, 8 and 9. The numerical values of the required solution are listed in the tables 3 and 4. The computation time for neural network solution is 1.7333 sec. whereas the RK method is 2.8888 sec. Hence the neural network solution is faster than RK method.

5. Conclusion

The solution of MRDE can be obtained by neural network approach. A neuro computing approach can yield a solution of MRDE significantly faster than standard solution techniques like RK method.A numerical example is given to illustrate the derived results. The long calculus time of finding optimal control is avoided by using neuro optimal controller. The efficient approximations of the optimal solution are done in MATLAB on PC, CPU 1.7 GHz.

REFERENCES

[1] S. P. Banks, Exact boundary controllability and optimal control for a generalized Korteweg de Vries equation,Internat. Nonlinear Anal. 47 2001) 5537-5546.

[2] P. Balasubramaniam, J. Abdul Samath, N. Kumaresan and A. Vincent Antony Kumar, Solution of matrix Riccati differential

equation for the linear quadratic singular system using neural networks, Appl. Math. Comput. 182(2006) 1832–1839.

[3] P. Balasubramaniam, J. Abdul Samath and N. Kumaresan, Neuro Approach for solving Matrix Riccati Differential Equations, Neural, Parallel Sci. Comput., 2 (2007) 125–135.

[4] S. P. Banks and K. J. Mhana, Optimal control and stabilization for nonlinear systems, IMA J. Math. Control Inf.,9 (1992), 179–196.

[5] K. E. Brenan, S. L. Campbell and L.R. Petzold, Numerical Solution of Initial Value Problems in Differential-Algebraic Equations, Elsevier, New York, 1989.

[6] S. L. Campbell, Singular Systems of Differential Equations, Pitman, Marshfield, MA, 1980.

[7] S. L. Campbell, Singular Systems of Differential Equations II, Pitman, Marshfield, MA, 1982.

[8] T. Cimen and S. P. Banks, Nonlinear optimal tracking control with application to super-tankers for autopilot design, Automatica, 40 (2004), 1845–1863.

[9] S. L. Campbell and E. Griepentrog, Solvability of general differential-algebraic equations, SIAM J. Sci. Comput., 16 (1995), 257–270.

[10] F. C. Chen and C. C. Liu, Adaptively controlling nonlinear continuous-time systems using multilayer neural networks, IEEE Trans. Automat. Control, 39 (1994), 1306–1310.

[11] L. Dai, Singular control systems, Lecture Notes in Control and Information Sciences, Springer , New York, 1989.

[12] G. Da Prato and A. Ichikawa, Quadratic control for linear periodic systems, Appl. Math. Optim.,18 (1988), 39–66.

[13] S. W. Ellacott, Aspects of the numerical analysis of neural networks, Acta Numer., 5 (1994), 145–202.

[14] F. M. Ham and E. G. Collins, A neurocomputing approach for solving the algebraic matrix Riccati equation,Proceedings IEEE International Conference on Neural networks, 1 (1996), 617 – 622.

[15] M. Jamshidi, An overview on the solutions of the algebraic matrix Riccati equation and related problems, LargeScale Systems, 1 (1980), 167–192.

[16] L. Jodar and E. Navarro, Closed analytical solution of Riccati type matrix differential equations, Indian J. Pureand Appl. Math., 2 (1992), 185–187.

[17] A. Karakasoglu, S. L. Sudharsanan and M. K. Sundareshan, Identification and decentralized adaptive control usingneural networks with application to robatic manipulators, IEEE Trans. Neural Networks, 4 (1993), 919–930.

[18] I. E. Lagaris, A. Likas and D. I. Fotiadis, Artificial neural networks for solving ordinary and partial differentialequations, IEEE Trans. Neural Networks 9 (1998), 987–1000.

[19] F. L. Lewis, A Survey of Linear Singular Systems, Circ. Syst. Sig. Proc., 5(1)(1986), 3–36.

[20] N. Lovren and M. Tomic, Analytic solution of the Riccati equation for the homing missile linear quadratic control problem, J. Guidance. Cont. Dynamics , 17 (1994), 619–621.

[21] D. Mccaffrey and S. P. Banks, Lagrangian manifolds and asymptotically optimal stabilizing feedback control, Syst.Control Lett., 43 (2001), 219-224.

[22] N. H. McClamroch, Feedback stabilization of control systems described by a class of nonlinear differential algebraicequations, Systems Control Lett., 15 (1990), 53–60.

[23] W. T. Miller, R. Sutton and P. Werbos, Neural networks for control, Cambridge, MA, MIT Press, 1990.

[24] K. S. Narendra and F. L. Lewis, Special issue on neural network feedback control, Automatica, 37 (2001), 1147-1148.

[25] K. S. Narendra and K. Parathasarathy, Identification and control of dynamical systems using neural networks,IEEE Trans. Neural networks, 1 (1990), 4–27.

[26] A. P. Paplinski, Lecture notes on feedforward multilayer neural networks, NNet(L.5), 2004.

[27] M. M. Polycarpou, Stable adaptive neural control scheme for nonlinear systems, IEEE Trans. Automat. Control,41 (1996), 447–451.

[28] E. Polak, Optimisation-Algorithms and consistent approximations, New York, Springer. 1997

[29]T. Parisini and R. Zoppoli, Neural approximations for infinitehorizon optimal control of nonlinear stochasticsystems, IEEE Trans. Neural Networks 9 (1998), 1388–1408.

[30] M. Razzaghi, A Schur method for the solution of the matrix Riccati equation, Int. J. Math. and Math. Sci., 20 (1997), 335–338.

[31] M. Razzaghi, Solution of the matrix Riccati equation in optimal control, Information Sci., 16 (1978), 61–73.

[32] M. Razzaghi, A computational solution for a matrix Riccati differential equation, Numerical Math., 32 (1979), 271–279.

[33]G. A. Rovithakis and M. A. Christodoulou, Adaptive control of

unknown plants using dynamical neural networks,IEEE Trans. Systems, Man and Cybernetics, 24 (1994), 400–412.

[34] N. Sadegh, A perceptron network for functional identification and control of nonlinear systems, IEEE Trans. Neural Networks, 4 (1993), 982–988.

[35] D. R. Vaughu, A negative exponential solution for the matrix Riccati equation, IEEE Trans Automat. Control, 14 (1969), 72–75.

[36] P. De. Wilde, Neural Network Models, Second ed., Springer-Verlag, London, 1997. [37] J. Wang and G. Wu, A multilayer recurrent neural network for solving continuous-time algebraic Riccati equations, Neural Networks, 11 (1998), 939–950.

[38] K. Zhou and Khargonekar, An algebraic Riccati equation approach to H optimization, Systems Control Lett., 11 (1998), 85–91.