Neural, Parallel, and Scientific Computations 16 (2008) 467-480

OPTIMAL CONTROL FOR STOCHASTIC LINEAR QUADRATIC SINGULAR SYSTEM WITH INDEFINITE CONTROL COST AND CROSS TERM USING NEURAL NETWORKS

N. KUMARESAN AND P. BALASUBRAMANIAM

Department of Mathematics
Gandhigram Rural University
Gandhigram - 624 302, Tamilnadu, India
kaygeekay1@yahoo.co.in pbalgri@rediffmail.com

ABSTRACT: In this paper, optimal control for stochastic linear singular system with indefinite control cost and cross term in the cost functional is obtained using neural networks. The goal is to provide optimal control with reduced calculus effort by comparing the solutions of the matrix Riccati differential equation (MRDE) obtained from well known traditional Runge Kutta (RK) method and nontraditional neural network method. To obtain the optimal control, the solution of MRDE is computed by feed forward neural network (FFNN). Accuracy of the solution of the neural network approach 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: Optimal control, Stochastic linear singular system with cross term, Matrix Riccati differential equation, Runge Kutta method and Neural networks

AMS (MOS) Subject Classifications: 49K45, 92B20, 93E20

1. INTRODUCTION

Stochastic linear quadratic regulator (LQR) problems have been studied by many researchers (Athens, 1971; Bensoussan, 1983; Bucci & Pandolfi, 2000; Davis, 1977; Wonham, 1968). Chen et al., (1998) have shown that the stochastic LQR problem is well posed if there are solutions to the Riccati equation and an optimal feedback control can then be obtained. For LQR problems, it is natural to study an associated Riccati equation. However, the existence and uniqueness of the solution of the Riccati equation in general, seem to be very difficult problems due to the presence of the complicated nonlinear term. Zhu & Li, (2003) used the iterative method for solving stochastic Riccati equations for stochastic LQR problems. There are several numerical methods to solve conventional Riccati equation as a result of the nonlinear process

The work of the authors is supported by the UGC-SAP (DRS) grant.

Received February 23, 2007

essential error accumulations may occur. In order to minimize the error, recently the conventional Riccati equation has been analyzed using neural network approach by Balasubramaniam et al., (2006) and Balasubramaniam et al., (2007). This paper is the extension of the neural network approach for solving stochastic Riccati equation.

Neural network approach is a nontraditional approach in which an artificial network is constructed based upon the behavior of the natural biological neurons (Wilde, 1997) and it is trained to learn a complex relationship between two or many variables or data sets. The advantage of this method is that once the network is trained, it allows instantaneous evaluation of solution at any desired number of points with negligible computing time and memory. Neural network approaches have been successfully applied in various fields such as function approximation, signal processing and adaptive (or) learning control for nonlinear systems (refer Karakasoglu, 1993; Narendra & Parthasarathy, 1990).

There are two different approaches for implementing neural networks in control. They are direct neural control design and indirect neural control design. Direct neural control design means that a neural network directly implements the controller — that is, the controller is a neural network. The network must be trained as the controller according to some criteria, using either numerical input — output data or a mathematical model of the system.

Indirect neural control design is based on a neural network model of the system to be controlled. In this case, the controller itself may not be a neural network, but it is derived from a plant that is modelled by a neural network. A neural network can be used to approximate control laws that govern the controlling inputs to the plant.

In this paper, neural network approach is used for function approximation. The function in the second term of the trial solution is approximated by neural network approach in order to obtain optimal control. Neural network approach is not applied directly in the components of the control.

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 Campbell, 1980; Campbell, 1982; Lewis, 1986).

Many practical processes can be modelled as descriptor systems such as constrained control problems, electrical circuits, certain population growth models and singular perturbations. In the past years, stability and control problems of descriptor systems have been extensively studied due to the fact that the descriptor system better describes physical systems than the state space systems. Compared with state space systems, the descriptor system has a more complicated yet richer structure. Furthermore, the study of the dynamic performance of descriptor systems is much more difficult than that for state space systems since descriptor systems usually have three types of modes, namely, finite dynamic modes, impulsive modes and non dynamic modes (Dai, 1989), while the latter two do not appear in the state space systems.

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 & Ichikawa (1988) 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 H_2 and H_∞ control (Zhou, 1998) 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 (Jamshidi, 1980). 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 (Jodar & Navarro, 1992). By using this approach, the solution of MRDE is obtained by partitioning the transition matrix of the associated Hamiltonian system (Vaughu, 1969). Another class of method is based on transforming MRDE into a linear matrix differential equation and then solving MRDE analytically or computationally (Lovren & Tomic, 1994; Razzaghi, 1978; Razzaghi, 1979). However, the method in (Razzaghi, 1997) is restricted for cases when certain coefficients of MRDE are non-singular. In(Jodar & Navarro, 1992), 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 $K(t) = \frac{p(t)}{f(t)}$, 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.

A variety of numerical algorithms (Choi, 1990) 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 (reader may refer Ellacott, 1994; Ham & Collins, 1996; Wang & Wu, 1998).

Solving MRDE by neural network is a novel approach. Computation time is very minimum when compared with existing methods stated in the survey paper (Choi, 1990). High accuracy of numerical solution compared with traditional method like Runge Kutta method has been analyzed. Once network is trained, it allows instantaneous evaluation of solutions at any desired number of points with less memory occupation. The neural network approach for solving MRDE is different from the previous methods discussed in the above mentioned survey paper in various directions stated sequentially as follows:

(i) In direct integration method, there is a difficulty to apply for higher order Riccati differential equation.

(ii) In negative exponential method, the computation of transformation matrix is numerically unreliable in a fixed precision of arithmetic.

(iii) In Davison Maki method, Pade's approximation is used to compute the solution. Accuracy of Pade's solution is not better than this neural network approach.

(iv) In ASP(Automatic Synthesis Program), matrix iteration procedure for computation of matrix exponential of size $2n \times 2n$ is time consuming.

(v) In a method using an algebric Riccati solution, the computation of matrix exponential of size $n \times n$ is also time consuming.

(vi) In square root method, accuracy of the solution is not satisfactory.

(vii) In analytic approximation method, the accuracy of solution depends upon accuracy of solution of algebraic Riccati equation.

(viii) A matrix valued approach, the accuracy of solution depends upon the solution of Sylvester equation.

Although parallel algorithms can compute the solutions faster than sequential algorithms, there have been no report on neural network solutions for MRDE that is compared with RK method solutions. This paper focuses upon the implementation of neuro computing approach for solving MRDE in order to get the optimal solution. The solution was found with uniform accuracy and trained neural network provides a compact expression for the analytical solution over the entire finite domain. An example is given which illustrates the advantage of the fast and accurate solutions compared to RK method.

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 stochastic linear dynamical singular system that can be expressed in the form:

$$Fdx(t) = [Ax(t) + Bu(t)]dt + Du(t)dW(t), \quad x(0) = 0, \quad t \in [0, t_f], \tag{1}$$

where the matrix F is possibly singular, $x(t) \in R^n$ is a generalized state space vector, $u(t) \in R^m$ is a control variable and it takes value in some Euclidean space, $W(t)$ is a Brownian motion and $A \in \mathbb{R}^{n\times n}$, $B \in \mathbb{R}^{n\times m}$ and $D \in \mathbb{R}^{n\times m}$are known coefficient matrices associated with $x(t)$ and $u(t)$ respectively, x_0 is given initial state vector and $m \leq n$.

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

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

where the superscript T denotes the transpose operator, $S \in \mathbb{R}^{n\times n}$ and $Q \in \mathbb{R}^{n\times n}$ are symmetric and positive definite (or semidefinite) weighting matrices for $x(t)$, $R \in \mathbb{R}^{m\times m}$ is a symmetric and indefinite (in particular negative definite) weighting matrix for $u(t)$ and $H \in \mathbb{R}^{m\times n}$ is a coefficient matrix. It will be assumed that $|sF - A| \neq 0$ for some s. This assumption guarantees that any input $u(t)$ will generate one and only one state trajectory $x(t)$.

The relative MRDE for the stochastic linear singular system (1) is

$$\left.\begin{array}{ll} F^T\dot{K}(t)F & +F^TK(t)A + A^TK(t)F + Q \\ & -F^T(K(t)B + H^T)(R + D^TK(t)D)^{-1}(B^TK(t) + H)F = 0 \\ & K(t_f) = F^TSF \end{array}\right\} \tag{2}$$

in which $R < 0$, $G(t) = (R + D^TK(t)D) > 0$ and

$$\begin{aligned} G(t)^+ &\geq R + D^TK(t)^+D \\ &\geq R + D^TK(t)^-D \\ &\geq G(t)^-. \end{aligned}$$

Here $K(t) \in \mathbb{R}^{n\times n}$ is a symmetric matrix, $G(t)^+ \in C[0, t_f; \widehat{M}_+^m]$, $G(t)^- \in C[0, t_f; \widehat{M}_+^m]$ and $\widehat{M}_+^m$ is the subspace of all positive definite $m \times m$ symmetric matrices of M^m.

As per the Remark 3.1 (Chen et al., 1998), the above MRDE admits a solution, then the optimal feedback control is given by

$$u(t) = -(R + D^T K(t)D)^{-1}(B^T K(t) + H)Fx(t).$$

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 (2)

$$\dot{k}_{ij}(t) = \phi_{ij}(k_{ij}(t)), \quad k_{ij}(t_f) = A_{ij} \quad (i, j = 1, 2, ..., n). \tag{3}$$

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 (3) 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.

$$\begin{aligned} k_{11}(i+1) &= k_{11}(i) + \frac{1}{6}\Big(k1 + 2k2 + 2k3 + k4\Big) \\ k_{12}(i+1) &= k_{12}(i) + \frac{1}{6}\Big(l1 + 2l2 + 2l3 + l4\Big) \end{aligned}$$

where

$$\begin{aligned} k1 &= h * \phi_{11}\Big(k_{11}, k_{12}\Big) \\ l1 &= h * \phi_{12}\Big(k_{11}, k_{12}\Big) \\ k2 &= h * \phi_{11}\Big(k_{11} + \frac{k1}{2}, k_{12} + \frac{l1}{2}\Big) \\ l2 &= h * \phi_{12}\Big(k_{11} + \frac{k1}{2}, k_{12} + \frac{l1}{2}\Big) \\ k3 &= h * \phi_{11}\Big(k_{11} + \frac{k2}{2}, k_{12} + \frac{l2}{2}\Big) \\ l3 &= h * \phi_{12}\Big(k_{11} + \frac{k2}{2}, k_{12} + \frac{l2}{2}\Big) \\ k4 &= h * \phi_{11}\Big(k_{11} + k3, k_{12} + l3\Big) \\ l4 &= h * \phi_{12}\Big(k_{11} + k3, k_{12} + l3\Big). \end{aligned}$$

In the similar way, the original system (3) 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 trial solution of equation (3) to the neural network solution of (3). The trial solution is expressed as the difference of two terms as below (see Lagaris, et. al., 1998)

$$(k_{ij})_a(t) = A_{ij} - tN_{ij}(t_j, w_{ij}). \tag{4}$$

The first term satisfies the TCs and contains no adjustable parameters. The second term employs a feedforward neural network and parameters w_{ij} 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 sigmoidal transfer function.

The error quantity to be minimized is given by

$$E_r = \sum_{i,j=1}^{n} \Big((\dot{k}_{ij})_a - \phi_{ij}((k_{ij})_a) \Big)^2. \tag{5}$$

The neural network is trained until the error function (5) becomes zero. Whenever E_r becomes zero, the trial solution (4) becomes the neural network solution of the equation (3).

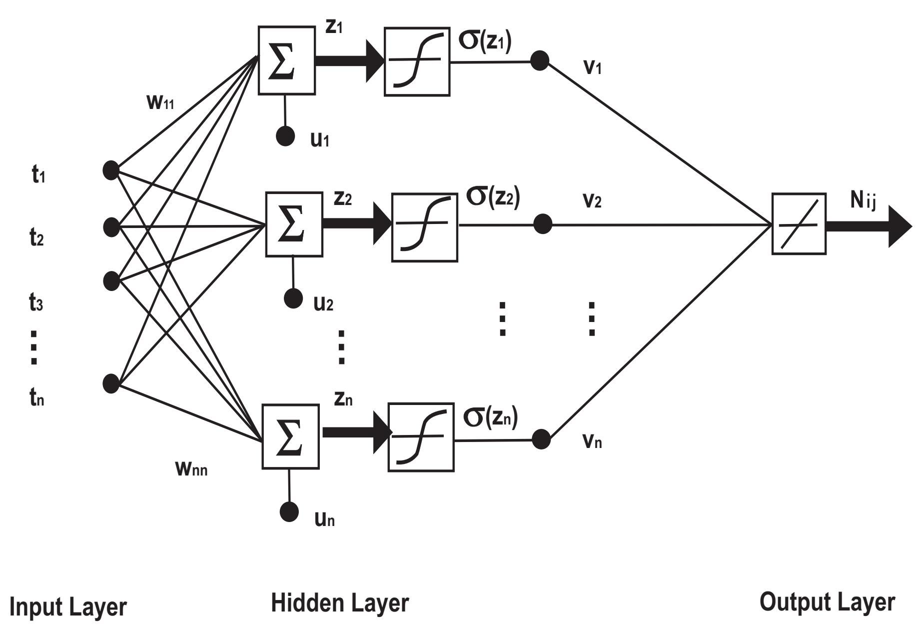

FIGURE 1. Neural network architecture

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 (Paplinski, 2004). The neural network architecture is given in the Fig. 1 for computing N_{ij}. The neural network algorithm was implemented in MATLAB on a PC, CPU 1.7 GHz for the neuro computing approach to solve MRDE (2) for the linear 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 until the following error function.

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

4. NUMERICAL EXAMPLE

Consider the two dimensional optimal control problem:
Minimize

$$J = E\Big\{\frac{1}{2}x^T(t_f)F^TSFx(t_f) + \frac{1}{2}\int_0^{t_f}[x^T(t)Qx(t) + u^T(t)Hx(t) + u^T(t)Ru(t)]dt\Big\}$$

subject to the stochastic linear singular system

$$Fdx(t) = [Ax(t) + Bu(t)]dt + Du(t)dW(t), \quad x(0) = x_0,$$

where $S \in \mathbb{R}^{2\times 2}$ and $Q \in \mathbb{R}^{2\times 2}$ are symmetric and positive definite (or semidefinite) weighting matrices for $x(t)$, $R \in \mathbb{R}^{1\times 1}$ is a symmetric and indefinite (in particular negative definite) weighting matrix for $u(t)$ and $H \in \mathbb{R}^{1\times 2}$ is a coefficient matrix, the matrix F is possibly singular, $x(t) \in R^2$ is a generalized state space vector, $u(t) \in R^1$ is a control variable and it takes value in some Euclidean space, W(t) is a Brownian motion and $A \in \mathbb{R}^{2\times 2}$, $B \in \mathbb{R}^{2\times 1}$ and $D \in \mathbb{R}^{2\times 1}$are known coefficient matrices associated with $x(t)$ and $u(t)$ respectively, x_0 is given initial state vector. The numerical implementation could be adapted by taking $t_f = 2$ for solving the related MRDE of the above stochastic linear singular system with the following appropriate matrices:

$$S = \begin{bmatrix} 2 & 0 \\ 0 & 0 \end{bmatrix}, \quad F = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}, \quad A = \begin{bmatrix} 1 & 2 \\ 0 & 4 \end{bmatrix}, \quad B = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \quad R = -1,$$

$$Q = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}, \quad D = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \quad H = \begin{bmatrix} 1 & 0 \end{bmatrix}.$$

The above matrices are substituted in equation (2), the MRDE is transformed into system of differential equation in k_{11} and k_{12}. In this problem, the value of k_{22} for 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 perceptron having one hidden layer with 10 hidden units and one linear output unit is used. The sigmoid activation function of each hidden units is $\sigma(t) = \frac{1}{1+e^{-t}}$.

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 Table 1. The computation time for neural network solution is 1.6 sec. whereas the RK method is 2.2 sec. Hence it is proved that the computation time for the rate of convergence of the neural network solution is faster than the solution of RK method.

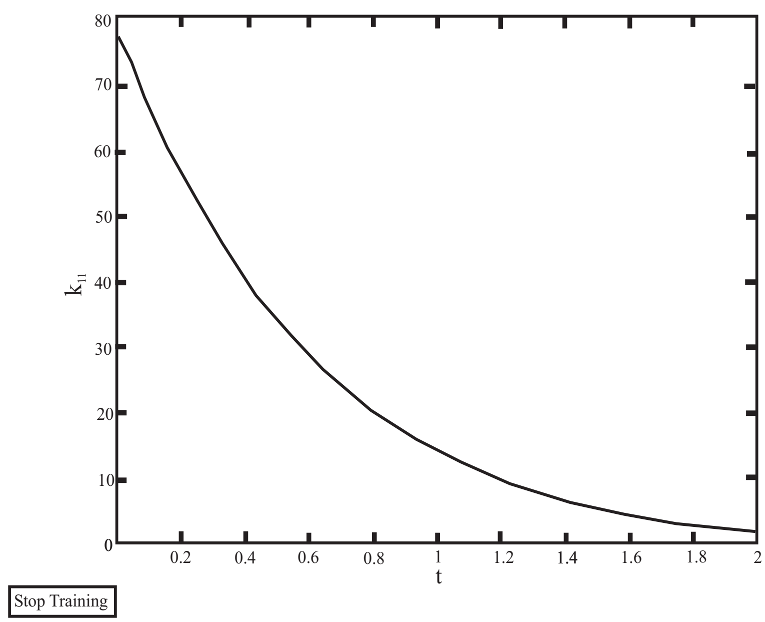

FIGURE 2. Solution curve for k_{11}

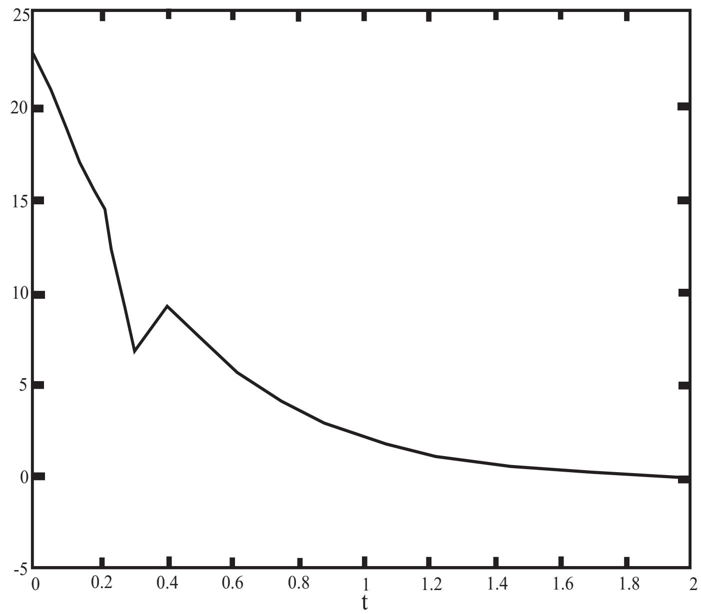

FIGURE 3. Error curve for k_{11}

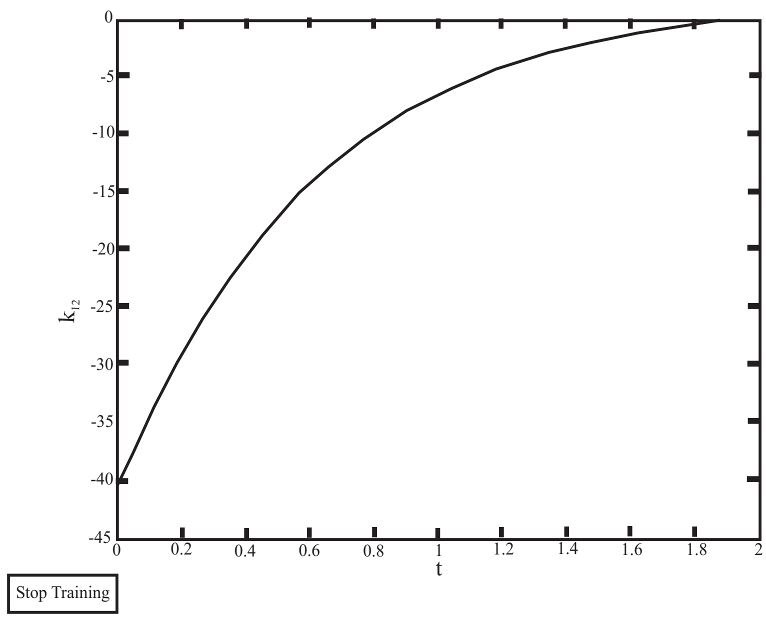

FIGURE 4. Solution curve for k_{12}

5. CONCLUSION

The optimal control for the stochastic linear singular system with indefinite control cost and cross term is 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

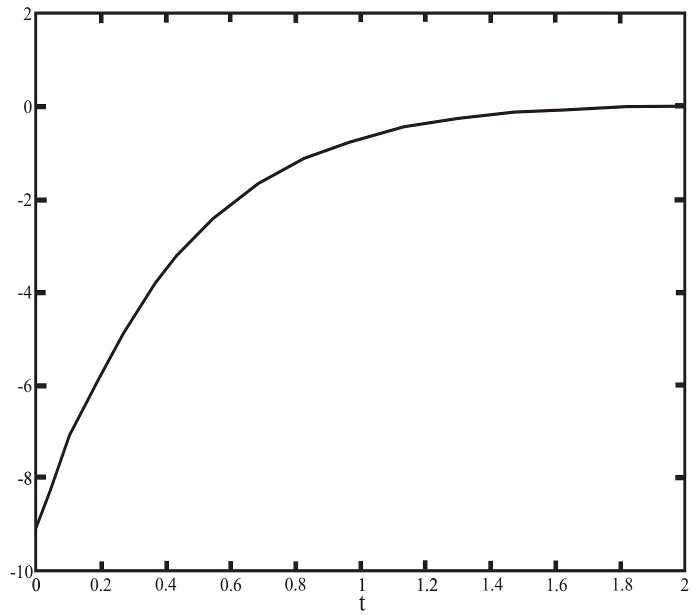

FIGURE 5. Error curve for k_{12}

t	Neural network solution		Runge Kutta solution	
	k_{11}	k_{12}	k_{11}	k_{12}
0.0	79.323700	-41.035915	102.287193	-50.143597
0.2	56.859161	-29.130484	71.621239	-34.810619
0.4	40.630684	-20.529818	50.013248	-24.006624
0.6	28.908840	-14.317326	34.789246	-16.394623
0.8	20.444633	-9.830992	24.065388	-11.032695
1.0	14.336570	-6.592892	16.514980	-7.257491
1.2	9.934965	-4.258411	11.204577	-4.602289
1.4	6.773743	-2.579914	7.479240	-2.739620
1.6	4.523712	-1.381439	4.883680	-1.441840
1.8	2.966949	-0.543376	3.113033	-0.556517
2.0	2.000000	0.000000	2.000000	0.000000

TABLE 1. Solutions of MRDE

optimal controller. The efficient approximations of the optimal solution are done in MATLAB on PC, CPU 1.7 GHz.

REFERENCES

[1] Athens, M. (1971), Special issues on linear quadratic Gaussian problem, *IEEE Trans. Automat. Control*, **AC-16**, 527–869.

[2] Balasubramaniam, P., Abdul Samath, J., Kumaresan, N. & Vincent Antony Kumar, A. (2006), Solution of matrix Riccati differential equation for the linear quadratic singular system using neural networks, *Appl. Math. Comput.*, **182**, 1832–1839.

[3] Balasubramaniam, P., Abdul Samath, J. & Kumaresan, N. (2007), Optimal control for nonlinear singular systems with quadratic performance using neural networks, *Appl. Math. Comput.*, **187**, 1535–1543.

[4] Balasubramaniam, P., Abdul Samath, J., Kumaresan, N. & Vincent Antony Kumar, A. (2007), Neuro approach for solving matrix Riccati differential equation, *Neural, Parallel Sci. Comput.*, **15**, 125–135.

[5] Bensoussan, A. (1983), Lecture on stochastic control part I. in : *Nonlinear and Stochastic Control* , Lecture Notes in Math. 972, pp. 1–39, Springer-Verlag, Berlin.

[6] Bucci, F. & Pandolfi, L. (2000), The regulator problem with indefinite quadratic cost for boundary control systems: The finite horizon case. *Systems Control Lett.*, **39**, 79–86.

[7] Campbell, S. L. (1980), *Singular Systems of Differential Equations*, Pitman, Marshfield.

[8] Campbell, S. L. (1982), *Singular Systems of Differential Equations II*, Pitman, Marshfield.

[9] Chen, S. P., Li, X. J. & Zho, X. Y. (1998), Stochastic linear quadratic regulators with indefinite control weight costs, *SIAM J. Control Optim.*, **36**(5), 1685–1702.

[10] Chiu H. Choi, (1990), A survey of numerical methods for solving matrix Riccati differential equation, *IEEE Proceedings Southeastcon*, 696–700.

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

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

[13] Davis, M. H. A.(1977), *Linear Estimation and Stochastic Control*, Chapman and Hall, London.

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

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

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

[17] Jodar, L. & Navarro, E. (1992), Closed analytical solution of Riccati type matrix differential equations, *Indian J. Pure Appl. Math.*, **23**, 185–187.

[18] Karakasoglu, A., Sudharsanan, S. L. & Sundareshan, M. K. (1993), Identification and decentralized adaptive control using neural networks with application to robotic manipulators, *IEEE Trans. Neural Networks*, **4**, 919–930.

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

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

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

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

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

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

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

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

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

[28] Wang, J. & Wu, G. (1998), A multilayer recurrent neural network for solving continuous time algebraic Riccati equations, *Neural Networks*, **11**, 939–950.

[29] Wilde, P. De. (1997) *Neural Network Models*, Second ed., Springer-Verlag, London.

[30] Wonham, W. M. (1968), On a matrix Riccati equation of stochastic control, *SIAM J. Control Optim.*, **6**, 681–697.

[31] Zhou, K. & Khargonekar, P. P. (1998), An algebraic Riccati equation approach to H_∞ optimization, *Systems Control Lett.*, **11**, 85–91.

[32] Zhu, J. and Li, K. (2003), An iterative method for solving stochastic Riccati differential equations for the stochastic LQR problem, *Optim. Methods Softw.*, **18** , 721–732.