Solving for a Class of Linear Matrix Inequalities Using Neural Networks

Chun-Liang Lin, Chi-Chii Lai, Teng-Hsien Huang and Tasi-Yuan Lin

Department of Automatic Control Engineering

Feng Chia University, Taichung, Taiwan, 40724, R.O.C.

Email: chunlin@fcu.edu.tw

Abstract

This paper proposes a new approach solving for a class of LMIs, which are commonly encountered in the robust control system analysis and design, using recurrent neural networks. The nature of parallel and distributed neural processing renders these networks possessing the computational advantages over the traditional sequential algorithms in real-time applications. The proposed networks are proven to be asymptotically in the large and capable of LMIs solving. Illustrative examples are provided to demonstrate the proposed results.

Key words: *recurrent neural network, linear matrix inequality, robust control, quadratic stability*

1. INTRODUCTION

LMIs are becoming basic tools in control as Lyapunov and Riccati equations became basic tools in the 1960s. They have emerged as a powerful formulation and design technique for a variety of linear control problems [1]. Construction for design objectives in systems and control theory can usually be cast or recast as semi-definite programming problems, i.e. LMI problems, for examples, robustness analysis and robust controller design [9,10,12], gain scheduled controller design [13], H_∞ design with pole placement constraints [4], stability analysis of fuzzy control system [18],..., etc. See [3] for an extensive research survey. Due to recent advances in convex optimization, efficient algorithms exist for solving LMI's; see [3,15] for excellent surveys of these methods and [7] for a MATLAB TOOLBOX. It turns out that the analysis problem can be solved using a numerical algorithm.

Traditional algorithms solving for LMIs perform sequentially numerical computations. For real-time control applications, when the solution is to be obtained within a time of the order of a hundred nanoseconds, a digital computer performing numerical computation may not comply with the desired computation time. In recent years, intensive investigation activity has been devoted to the application of artificial neural networks for solving the matrix equations [5,16,19] and linear programming problems [6,17,20] due to their parallel and distributed nature.

The presented approach constitutes a solution to the LMI problems by applying recurrent neural networks. Because of the parallel distributed nature of the neural networks, and advances in VLSI technology make it possible to fabricate microelectronic networks of high complexity, thus they can be available computational models for the real-time control purpose [5,21]. Many researches have been reported for solving systems of linear equations and related problems with artificial neural networks. However, there have few literatures focusing on on-line solutions to the Riccati matrix equations or related equations [12]. Specially, it was rarely seen that one solves LMIs based on neural network processing [11]. Throughout this paper, an emphasis is placed on synthesizing a variety of recurrent neural networks to solve LMIs. The algorithm involves four important extensions beyond traditional numerical methods. First, it takes advantage of the special structure of the matrix inequalities, e.g. Lyapunov or Riccati. Second, the neural dynamics enables us to solve many optimization problems in real-time due to the massively parallel operations of the computing units and due to the better convergence properties in comparison with iterative schemes. Third, the neuro dynamical system implemented on the basis of differential equations usually exhibits more robustness to certain variations and it tends to retain information better through time. Finally, the most important one is that it possesses the potential for synthesizing real-time robust adaptive controllers for time-varying or gain-scheduled systems. We consider in this paper the following problems:

- Lyapunov matrix inequalities for stability and performance analysis of linear differential inclusion
- joint synthesis of state feedback and Lyapunov or Riccati inequalities for linear differential inclusions

As certain representative LMIs are studied, however, the proposed approach can be directly extended to solve other types of LMIs.

2. NEURAL DYNAMIC EQUATION

Formulation of LMI based on Lyapunov's methods will be our main focus. Advantages of using the methods are threefold. First, many problems from Lyapunov theory can be cast as convex or quasiconvex problem. Second, the method can just as well be used to find bounds on system performance provided one do not insist on analytic solution. Third, the neural network solving for Lyapunov equations can be realized using hardware. These features make Lyapunov's method useful in control applications.

The class of generalized Lyapunov inequalities are commonly used for determining a robust stabilizing control law. The LMI is described by (1)

$$L(P) = BPA + A^T PB^T + D < 0, \quad P > 0 \tag{1}$$

where A, B, $D = D^T > 0$ are given matrices of appropriate sizes, $P = P^T \in \Re^{m\times m}$ is the variable. The inequality symbol '<' ('>') in (1) means that $L(\cdot)$ (P) is negative definite (positive definite), i.e., $u^T L(P)u < 0\,(P > 0)$, $\forall u \notin 0, u \in \Re^n$. If $L(P)$ is simplified as $L(P) = PA + A^T P$, then it is well known from the linear system theory that this LMI is feasible (i.e. the LMI will be solvable and result in $P > 0$), if and only if, the matrix A is asymptotically stable. To simplify the representation let us only consider the general form:

$$L(P) < 0, \quad P > 0 \tag{2}$$

To solve for LMIs in terms of recurrent neural networks, we first impose two slack matrices converting the LMI into the linear matrix equalities of the forms:

$$G_1(P, R_1) = L(P) + \tilde{R}_1\tilde{R}_1^T = 0 \tag{3a}$$

$$G_2(P, R_2) = P - \tilde{R}_2\tilde{R}_2^T = 0 \tag{3b}$$

where P is named as the solution matrix, $G_1 \in \Re^{n\times n}$ and $G_2 \in \Re^{m\times m}$ are the objective matrices, the slack matrices $\tilde{R}_{1,2}$ are restricted to be nonsingular, positive definite as the following forms

$$\tilde{R}_1 = H_1(R_1) = \begin{bmatrix} h_{11}(r_{1,11}) & 0 & 0 & & 0 \\ r_{1,21} & h_{12}(r_{1,22}) & 0 & & 0 \\ r_{1,31} & r_{1,32} & h_{13}(r_{1,33}) & & \\ & & & & 0 \\ r_{1,n1} & r_{1,n2} & r_{1,n3} & & h_{1n}(r_{1,nn}) \end{bmatrix} \tag{4a}$$

$$\tilde{R}_2 = H_2(R_2) = \begin{bmatrix} h_{21}(r_{2,11}) & 0 & 0 & & 0 \\ r_{2,21} & h_{22}(r_{2,22}) & 0 & & 0 \\ r_{2,31} & r_{2,32} & h_{23}(r_{2,33}) & & \\ & & & & 0 \\ r_{2,m1} & r_{2,m2} & r_{2,m3} & & h_{2m}(r_{2,mm}) \end{bmatrix} \tag{4b}$$

with $R_1 = [r_{1,ij}]_{n\times n}$, $R_2 = [r_{2,ij}]_{m\times m}$, $\tilde{r}_{s,jj} = h_s(r_{s,jj}) > 0$, $\forall s, j$. The existence of the decomposition $\tilde{R}_s\tilde{R}_s^T$ for a positive definite matrix is ensured by the matrix theory [14].

Lemma 1: A principal submatrix of a positive definite matrix is also positive definite.

Lemma 2: The matrix A is positive definite, if and only if, there is a unique lower triangular matrix $\tilde{R}$ with positive diagonal elements such that $A = \tilde{R}\tilde{R}^T$

It follows from Lemma 1 that the matrices $\tilde{R}_{1,2}$ defined in (4) are positive definite. Lemma 2 is known as the Cholesky decomposition. It ensures that the LMI in (2) can be expressed in the form of (3).

The next step is to establish a convex computation energy function:

$$E[G(P,R_1,R_2)] = \sum_{i=1}^{n}\sum_{j=1}^{n} e_{1,ij}[g_{1,ij}(P,R_1)] + \sum_{i=1}^{m}\sum_{j=1}^{m} e_{2,ij}[g_{2,ij}(P,R_2)] \tag{5}$$

where $e_{s,ij}$ are the error functions, they measure the degree of constraint violation of elements $g_{s,ij}(P,R_s)$, $\forall s,i,j$. Solving for the LMI now becomes a constrained optimization problem. The derivation of the energy function enables us to transform the minimization problem into a set of ordinary differential equations based on neural networks with appropriate synaptic weights, input excitation, bias, and nonlinear or linear activation function. The dynamically neural network for solving linear matrix equalities (3a)-(3b) are described by

$$\frac{dP(t)}{dt} = -\eta_p \frac{\partial E}{\partial P} = -\eta_p \Omega \tag{6a}$$

$$\frac{dR_1(t)}{dt} = -\eta_{r1} \frac{\partial E}{\partial \tilde{R}_1} = -\eta_{r1} W_1 \tag{6b}$$

$$\frac{dR_2(t)}{dt} = -\eta_{r2} \frac{\partial E}{\partial \tilde{R}_2} = -\eta_{r2} W_2 \tag{6c}$$

$$\tilde{R}_s = H_s(R_s), \quad s = 1,2 \tag{6d}$$

where the derivative of a scalar valued-function E with respect to a matrix is defined by

$$\frac{\partial E}{\partial P} = \left[\frac{\partial E}{\partial p_{ij}}\right]_{m\times m}, \quad i,j = 1, \quad ,m$$

The derivatives $\partial E / \partial \tilde{R}_s$, $s = 1,2$ are defined similarly. In the above, $P(t)$ and $R_s(t), s = 1,2$ are activation state matrices of the recurrent neural network, $\eta_p, \eta_{rs} > 0$ are the learning rates and $\Omega = [\omega_{ij}]_{m\times m}$, $W_1 = [w_{1,ij}]_{n\times n}$, $W_2 = [w_{2,ij}]_{m\times m}$ with

$$\omega_{ij} = \sum_{k=1}^{n}\sum_{l=1}^{n} \frac{\partial g_{1,kl}(P,R_1)}{\partial p_{ij}} f_{1,kl}(g_{1,kl}(P,R_1)) + \sum_{k=1}^{m}\sum_{l=1}^{m} \frac{\partial g_{2,kl}(P,R_2)}{\partial p_{ij}} f_{2,kl}(g_{2,kl}(P,R_2)),\ i = 1, \quad ,m, j = 1, \quad ,m \tag{7a}$$

$$w_{1,ij} = \sum_{k=1}^{n}\sum_{l=1}^{n} \frac{\partial g_{1,kl}(P,R_1)}{\partial \tilde{r}_{1,ij}} f_{1,kl}(g_{1,kl}(P,R_1)), \quad i = 1, \quad ,n,\ j = 1, \quad ,n \tag{7b}$$

$$w_{2,ij} = \sum_{k=1}^{m}\sum_{l=1}^{m} \frac{\partial g_{2,kl}(P,R_2)}{\partial \tilde{r}_{2,ij}} f_{2,kl}(g_{2,kl}(P,R_2)), \quad i = 1, \quad ,m,\ j = 1, \quad ,m \tag{7c}$$

in which $f_{s,kl}(g_{s,kl}) = \dfrac{\partial e_{s,kl}}{\partial g_{s,kl}}$ can be viewed as the activation function with the input $g_{s,kl}$. Note that $e_{s,ij}(g_{s,ij})$ and $f_{s,ij}(g_{s,ij})$ are functions of $g_{s,ij}$ only. The constraints $\tilde{r}_{s,ii} > 0, \forall s, i$ imposing on $R_s(t)$ can be fulfilled by employing a limiting integrator with the continuously nonlinear transformation (see Fig. 1):

$$h_{si}(r_{s,ii}) = \varepsilon + \frac{p_{s,\max} - \varepsilon}{1 + e^{-\lambda r_{s,ii}}}, \quad \forall s, i \tag{8a}$$

where the steepness factor $\lambda > 0$ and ε is an arbitrarily small constant. Note that the nonlinear limiter is continuously differentiable with

$$h_{si}(r_{s,ii}) = \frac{\lambda(p_{s,\max} - \varepsilon)e^{-\lambda r_{s,ii}}}{(1+e^{-\lambda r_{s,ii}})^2} > 0, \forall s, i \quad (8b)$$

See Fig. 1 for a graphical illustration of h_{si}. It should be noted that the limiter's level $p_{s,\max}$ is adjustable. Adjusting $p_{s,\max}$ the neural network will result in different solutions to the LMI (2). Therefore it can be used as a design parameter for the control design. Equations (6a)-(6c) mean that the activation state matrices $P(t)$ and $R_s(t)$ evolve in the direction of negative gradient of $E[G(P,R_1,R_2)]$ as time evolves. In other words, the steady activation state matrix $\lim_{t\to\infty} P(t) = \overline{P}$ minimizes $E[G(P,R_1,R_2)]$ in a gradient descent fashion. This is a direct matrix representation for the minimization of convex functions extending from the traditional gradient-descent algorithm.

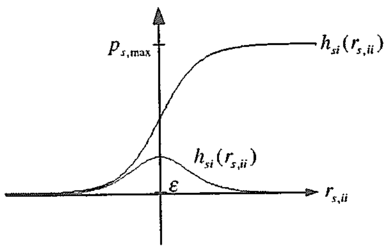

Fig. 1 A nonlinear transformation

Before extending the above results to solve practical LMI problems, we first present the following useful derivative operations of a scalar-valued function defined as (5) with respect to the solution matrix P. These equalities will be useful for putting various LMIs in the form of (6).

Lemma 3: Suppose that the energy function $E = E[G(X)]$, the solution matrix $X \in \Re^{m\times n}$ and the non-decreasing activation matrix $F = [f_{ij}(g_{ij})]$, then the following derivatives hold:

(i) $G(X) = AX$, $A \in \Re^{l\times m}$ then $\frac{\partial E}{\partial X} = A^T F$, $F \in \Re^{l\times n}$

(ii) $G(X) = AX^T$, $A \in \Re^{l\times n}$ then $\frac{\partial E}{\partial X} = F^T A$, $F \in \Re^{l\times m}$

(iii) $G(X) = XA$, $A \in \Re^{n\times l}$ then $\frac{\partial E}{\partial X} = FA^T$, $F \in \Re^{m\times l}$

(iv) $G(X) = X^T A$, $A \in \Re^{m\times l}$ then $\frac{\partial E}{\partial X} = AF^T$, $F \in \Re^{n\times l}$

(v) $G(X) = A^T XH$, $A \in \Re^{n\times n}$ and $H \in \Re^{n\times n}$, then $\frac{\partial E}{\partial X} = AFH^T$, $F \in \Re^{n\times n}$

(vi) $G(X) = AXH^T$, $A \in \Re^{n\times n}$ and $H \in \Re^{n\times n}$, then $\frac{\partial E}{\partial X} = A^T FH$, $F \in \Re^{n\times n}$

Based on (6) and Lemma 3, we can now obtain a set of neural dynamic equations solving for the LMI (1) as follows

$$\frac{dP}{dt} = -\eta_p[B^T F_1(P,\tilde{R}_1)A^T + AF_1(P,\tilde{R}_1)B + F_2(P,\tilde{R}_2)] \quad (9a)$$

$$\frac{dR_1}{dt} = -\eta_{r1}F_1(P,\tilde{R}_1)\tilde{R}_1 \quad (9b)$$

$$\frac{dR_2}{dt} = -\eta_{r2}F_2(P,\tilde{R}_2)\tilde{R}_2 \quad (9c)$$

where the learning rates $\eta_p, \eta_{r1,r2} > 0$, and the activation matrices are

$$F_1(P,\tilde{R}_1) = F_1(BPA + A^T PB^T + D + \tilde{R}_1\tilde{R}_1^T)$$
$$F_2(P,\tilde{R}_2) = F_2(P - \tilde{R}_2\tilde{R}_2^T)$$

Remark: For the discrete Lyapunov matrix inequality described by

$$L(P) = A^T PA - P < 0, \quad P > 0$$

where $P = P^T \in \Re^{m\times m}$ is the variable. Suppose that A is asymptotically stable in the discrete-time sense. Therefore, P must be positive definite. The corresponding neural dynamic equations solving for the solution candidates are given by

$$\frac{dP}{dt} = -\eta_p[AF(P,\tilde{R})A^T - F(P,\tilde{R})]$$
$$\frac{dR}{dt} = -\eta_r F(P,\tilde{R})\tilde{R}$$

where $\tilde{R}$ is defined as in (4), the activation matrix is

$$F(P,\tilde{R}) = F(A^T PA - P + \tilde{R}\tilde{R}^T)$$

3. STABILITY ANALYSIS

The elements of the recurrent neural networks (6a)-(6c) can be described as follows:

$$\frac{dp_{ij}(t)}{dt} = -\eta_p\left\{\sum_{k=1}^{n}\sum_{l=1}^{n}\frac{\partial g_{1,kl}(P,R_1)}{\partial P_{ij}} f_{1,ij}[g_{1,ij}(P,R_1)]\right\} \quad (10a)$$

$$\frac{dr_{s,ij}(t)}{dt} = -\eta_{rs}\left\{\sum_{k=1}^{n}\sum_{l=1}^{n}\frac{\partial g_{s,kl}(P,R_s)}{\partial \tilde{r}_{s,ij}} f_{s,ij}[g_{s,ij}(P,R_s)]\right\}, s = 1,2 \quad (10b)$$

The architecture of the recurrent neural network consists of three output layers and two hidden layers which are bidirectionally connected with the output layers of neurons. There is a functional transformation $f_{s,ij}(\cdot)$ for each neuron in the hidden layer, an integral transformation for each neuron of p_{ij} in the output layer, and an integral-limiting transformation for each neuron of $r_{s,ij}$ in the output layer. Each layer is composed of an array of neurons. The activation state matrix $P(t)$ corresponds to the solution matrix. Since $g_{s,kl}(\cdot,\cdot)$ is a linear function, $\frac{\partial g_{s,kl}}{\partial p_{ij}}$ and $\frac{\partial g_{s,kl}}{\partial \tilde{r}_{s,ij}}$ are constant. The connection weights from the (k,l)-th hidden neuron to the (i,j)-th output neuron of $P(t)$ and $R_s(t)$ are defined as $-\eta_p\frac{\partial g_{s,kl}}{\partial p_{ij}}$ and $-\eta_s\frac{\partial g_{s,kl}}{\partial \tilde{r}_{s,ij}}$, respectively. The connection weights from the (k,l)-th output neuron of $P(t)$ and $R_s(t)$ to the (i,j)-th hidden neuron are defined by the coefficients of $p_{kl}(t)$ and $\tilde{r}_{s,kl}(t)$ in $g_{s,ij}(P,R_s)$, respectively. There is no lateral connection among the neurons in either layer.

It should be emphasized that the effectiveness and convergence of the presented neural networks depend significantly on the values of the learning rates. The optimal choice usually depends on the scheme being solved. This problem seems to be similar to other gradient-based optimization task. Larger gains accelerate the convergence. However, they may overshoot the solution.

4. NEURAL DYNAMICS FOR A CLASS OF LMIS

(i) Simultaneous Lyapunov inequality problem

The problem arises in determining Lyapunov stability for linear inclusion systems of the form:

$$x = A(t)x \tag{11}$$

The system is stable for $A(t)$ in the convex set [2]

$$A(t) \in C_o \equiv \text{convex hull}(A_1, A_2, \quad , A_L)$$

where $A_i \in \Re^{n\times n}, \forall i$ are given and stable, if, and only if, there exists a global variable $P>0$ such that

$$A^T P + PA < 0, A \in C_o \tag{12}$$

or equivalently,

$$A_i^T P + PA_i < 0, \quad i = 1, \quad , L \tag{13}$$

Determining quadratic stability for the system is an LMI problem in the variable P. To determine the variable P, the LMIs are converted into a group of linear matrix equalities:

$$A_i^T P + PA_i + \tilde{R}_i \tilde{R}_i^T = 0, \quad i = 1, \quad , L$$

where $\tilde{R}_i, i = 1, \quad , L$ are defined as in (4). Since $A_i \in \Re^{n\times n}, \forall i$ have been assumed to be stable, thus P must be positive definite. The remaining work is to find the permissible matrix P satisfying (12).

To find the neural dynamics, the following objective matrices are defined:

$$G_i(P, R_i) = A_i^T P + PA_i + \tilde{R}_i \tilde{R}_i^T, \quad i = 1, \quad , L \tag{14}$$

Based on these objective matrices and using Lemma 3, the neural dynamics solving for the problem can be easily formulated as follows

$$\frac{dP}{dt} = -\eta_P \frac{\partial E}{\partial P} = -\eta_P \sum_{i=1}^{L} [A_i F_i(P, \tilde{R}_i) + F_i(P, \tilde{R}_i) A_i^T] \tag{15a}$$

$$\frac{dR_i}{dt} = -\eta_{r_i} \frac{\partial E}{\partial \tilde{R}_i} = -\eta_{r_i} F_i(P, \tilde{R}_i) \tilde{R}_i, i = 1, \quad , L \tag{15b}$$

where the activation matrix is

$$F_i(P, \tilde{R}_i) = F_i(A_i^T P + PA_i + \tilde{R}_i \tilde{R}_i^T), i = 1, \quad , L$$

(ii) Stabilizingtabilizing control problem

Consider the linear time-invariant system

$$x = Ax + Bu \tag{16}$$

where $A \in \Re^{n\times n}, B \in \Re^{n\times m}$. The state feedback control law is given by

$$u = Kx \tag{17}$$

The system is said to be quadratically stabilizable if there exists a state-feedback gain K such that the closed-loop system $x = (A+BK)x$ is quadratically stable, or equivalently, there exists $P = P^T > 0$ such that

$$(A+BK)^T P + P(A+BK) < 0 \tag{18}$$

An alternative equivalent condition is that, there exists $Q = Q^T > 0$ such that

$$Q(A+BK)^T + (A+BK)Q < 0 \tag{19}$$

Define $X = KQ$, so that $K = XQ^{-1}$. Substituting this into (24) yields

$$AQ + QA^T + BX + X^T B^T < 0 \tag{20}$$

The LMI includes two variables in Q and X. Thus, the closed-loop system is quadratically stabilizable if and only if there exist $Q>0$ and X such that the LMI (20) holds.

Using the elimination procedure for matrix variables Q and X, (20) can be equivalently expressed as

$$\tilde{B}^T (AQ + QA^T) \tilde{B} < 0 \tag{21}$$

where $\tilde{B}$ is an orthogonal complement of B, i.e.

$$\tilde{B}^T B = 0$$

and $[\tilde{B} \quad B]$ is nonsingular. Now using Finsler's lemma [3], (21) is also equivalent to

$$AQ + QA^T - \sigma BB^T < 0, \quad \sigma \in \Re \tag{22}$$

where σ is some scalar. Clearly, the variable X has been eliminated from (20). We can always assume $\sigma > 0$, and since (22) is homogeneous in Q and σ, we may set $\sigma = 1$ without loss of generality. If $Q>0$ satisfies (22), a stabilizing state-feedback gain is given by

$$K = -\frac{1}{2}\sigma B^T Q^{-1} \tag{23}$$

Note that the simplified LMI (22) remains only one variable Q. To solve this equation, let us define the objective matrix

$$G(Q, R) = AQ + QA^T - \sigma BB^T + \tilde{R}\tilde{R}^T = 0 \tag{24}$$

where the slack matrix $\tilde{R}$ is defined as in (4). The corresponding dynamic equations of the recurrent neural network can be obtained by applying (6) and Lemma 3

$$\frac{dQ}{dt} = -\eta_q \frac{\partial E}{\partial Q} = -\eta_q (A^T F(Q, \tilde{R}) + F(Q, \tilde{R}) A) \tag{25a}$$

$$\frac{dR}{dt} = -\eta_r \frac{\partial E}{\partial \tilde{R}} = -\eta_r F(Q, \tilde{R}) \tilde{R} \tag{25b}$$

where the activation matrix is

$$F(Q, \tilde{R}) = F(AQ + QA^T - \sigma BB^T + \tilde{R}\tilde{R}^T) \tag{25c}$$

As Q is obtained from (25a), the control gain is recovered as $K = -0.5\sigma B^T Q^{-1}$. For the actual applications, however, computation for the matrix inverse Q^{-1} should be avoided. Under this consideration, the neural network given in [12] can also be used to realize the gain.

(iii) Algebraic Riccati matrix inequality problem

The class of algebraic Riccati matrix inequalities are commonly used in quadratic stabilizing control for determining a robust stabilizing control law. It is typically described by

$$A^T P + PA + (PB - C^T)(D + D^T)^{-1}(B^T P - C) + Q < 0, \tag{26}$$

where $P = P^T > 0$ is the variable, $D + D^T > 0$, and A, B, C, D, $Q = Q^T$ are given matrices of appropriate sizes. This is a quadratic matrix inequality in the variable P. We assume for simplicity that A is stable and the system (A,B,C) is minimal. Using Schur theorem [8], it can be equivalently expressed as an LMI:

$$\begin{bmatrix} PA+A^TP+Q & PB-C^T \\ B^TP-C & -D^T-D \end{bmatrix} < 0, \quad P>0 \qquad (27)$$

Or equivalently

$$\overline{A}P\overline{B}+\overline{B}^TP\overline{A}^T+\overline{D}<0, \quad P>0 \qquad (28)$$

where

$$\overline{A}=\begin{bmatrix} I \\ 0 \end{bmatrix}, \overline{B}=\begin{bmatrix} A & B \end{bmatrix}, \ \overline{D}=\begin{bmatrix} Q & -C^T \\ -C & -D^T-D \end{bmatrix}$$

A Riccati equation has many solutions. However, if the equation has a symmetric positive definite solution then it is unique. Suppose that there is a positive definite solution to (26). Now to avoid the solution obtained converges to the one which is not positive definite, an additional constraint as in (3b) can be imposed:

$$G_1(P,R_1)=\overline{A}P\overline{B}+\overline{B}^TP\overline{A}^T+\overline{D}+\tilde{R}_1\tilde{R}_1^T \qquad (29a)$$

$$G_2(P,R_2)=P-\tilde{R}_2\tilde{R}_2^T \qquad (29b)$$

Applying the derivation presented in (6) and Lemma 3, the recurrent neural dynamics for solving the Riccati matrix inequality is given by

$$\frac{dP}{dt}=-\eta_p[\overline{A}^TF_1(P,\tilde{R}_1)\overline{B}^T+\overline{B}F_1(P,\tilde{R}_1)\overline{A}+F_2(P,\tilde{R}_2)] \qquad (30a)$$

$$\frac{dR_1}{dt}=-\eta_{r1}F_1(P,\tilde{R}_1)\tilde{R}_1 \qquad (30b)$$

$$\frac{dR_2}{dt}=-\eta_{r2}F_2(P,\tilde{R}_2)\tilde{R}_2 \qquad (30c)$$

where the activation matrices are

$$F_1(P,\tilde{R}_1)=F_1(\overline{A}P\overline{B}+\overline{B}^TP\overline{A}^T+\overline{D}+\tilde{R}_1\tilde{R}_1^T)$$

$$F_2(P,\tilde{R}_2)=F_2(P-\tilde{R}_2\tilde{R}_2^T)$$

4. ILLUSTRATIVE EXAMPLES

In the following examples, elements of the convex energy function were defined as

$$e_{ij}=g_{ij}\tan^{-1}(g_{ij})-\ln\sqrt{1+g_{ij}^{\ 2}}, \quad \forall i,j$$

Therefore the corresponding activation functions are

$$f_{ij}=\tan^{-1}(g_{ij}), \quad \forall i,j$$

Example 1. Consider the perturbed system of the form

$$x=Ax+Bu+\tilde{\Delta}(x,u,t), \quad (A,B)\in C_o\{(A_1,B_1), \quad ,(A_L,B_L)\} \qquad (31)$$

where $B_i=B(\sigma_i)$, $A_i=A(\sigma_i), i=1, \quad ,L$ are the stable system matrices evaluated at the i-th operating point σ_i, $\tilde{\Delta}(x,u,t)$ is the nonlinear time-varying perturbation satisfying the matching condition [2]

$$\tilde{\Delta}(x,u,t)=B\Delta(x,u,t)$$

and the perturbation upper bound is estimated as

$$\|\Delta(x,u,t)\|\le\alpha\|x\|+\beta\|u\|, \ \beta<1$$

over all operating conditions. In practical applications, $\tilde{\Delta}(x,u,t)$ can be viewed as the linearization error with A and B being the nominal system matrices. Since $A_i, i=1, \quad ,L$ have been assumed to stable, the control law is mainly designed to ensure the overall stability while there is in the presence of the perturbation.

Since $A_i, i=1, \quad ,L$ are stable we know that there exist $P_i>0, \forall i$ such that

$$P_iA_i+A_i^TP_i<0, \forall i \qquad (32)$$

The LMIs can be converted into the linear matrix equalities by imposing slack matrices $\tilde{R}_i$:

$$P_iA_i+A_i^TP_i+\tilde{R}_i\tilde{R}_i^T=0, \quad i=1, \quad ,L \qquad (33)$$

Based on the preliminary result given in [2], it can be proven that the perturbed system would be stable provided that the control law at the i-th operating condition is given by

$$u=-K_ix$$

where $K_i=K(\sigma_i)=\gamma_iB^TP_i, \gamma_i>0$, the parameter γ_i satisfies the following condition

$$\gamma_i>\frac{\alpha^2}{2\lambda_{\min}(\tilde{R}_i\tilde{R}_i^T)(1-\beta)}, i=1, \quad ,L \qquad (34)$$

where $\lambda_{\min}(\cdot)$ denotes the minimum eigenvalue. The control law for intermediate operating conditions are linearly interpolated by computing the function $\hat{K}(\sigma_i)$ such that

$$\hat{K}(\sigma_i)=K(\sigma_i)$$

Then the actual control law is implemented as

$$u(t)=-K(\sigma(t))x(t)$$

Now consider a perturbed system of the form of (1) with the stable coefficient matrix A and vector B

$$A=\begin{bmatrix} -1-\left|\sin\left(\frac{10\pi\sigma_i}{4}\right)\right| & -1-\left|\cos\left(\frac{10\pi\sigma_i}{4}\right)\right| & -2 \\ 0 & -2-\left|\cos\left(\frac{10\pi\sigma_i}{4}\right)\right| & -3 \\ 0 & 0 & -3 \end{bmatrix}, \quad B=\begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$$

where $\sigma_i=0.1, 0.2, 0.3,$ are consistent with the operating time. The nonlinear time-varying disturbance Δ is supposed to be

$$\Delta(x,u,t)=0.1x_1\sin(x_1t)+0.2e^{-2t}u\cos(ut)$$

Let the initial states of the neural state P and the plant state x were, respectively, $P(0)=[0.1]_{3\times3}$ and $x(0)=[0.1 \ \ 0.1 \ \ 0.1]^T$. Learning rates for the neural networks were chosen as $\eta_p=80$ and $\eta_r=500$. The upper and lower limits of the nonlinear transformations in (8a) were chosen as $p_{s,\max}=2$ and $\varepsilon=0.1$. We used the neural dynamics (9) to solve the Lyapunov matrix inequality (32) at every operating point. Figures 2(a)-2(c) illustrate, respectively, the convergent behavior of the activation states to the solution matrix P, the slack matrix $\tilde{R}_1$ and the objective matrix G. For clarity, only lower triangular elements of these matrices and the responses within 0. to 0.5 seconds were shown. Note that the coefficient matrix A changes at every 0.1 seconds. The solution matrix P and the slack matrix R reach exact steady state values (i.e. the matrix values such that (33) holds) less than 0.03 seconds. Figures 3(a) and 3(b) illustrate, respectively, the transient responses of the system state x and the control signal u. The parameter γ determined by the robust stability condition (34) is illustrated in Fig. 4. It was found from the transient response of x that, the influence caused by the nonlinear time-varying perturbation has been successfully restrained by the control signal. Clearly, the feedback control system is robustly stabilized. It is can also be seen that the proposed neural network solving for the solution of the LMI can follow the slow variation of the system parameters to generate stabilizing control signals.

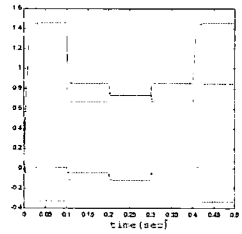

Fig. 2(a) Transient of P

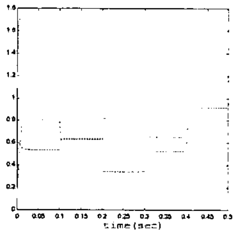

Fig. 2(b) Transient of $\tilde{R}$

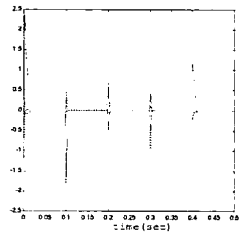

Fig. 2(c) Transient of G

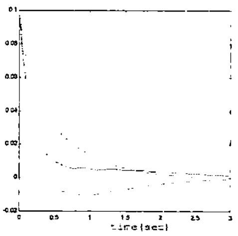

Fig. 3(a) Transient of x

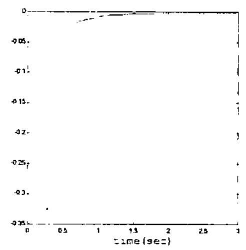

Fig. 3(b) Transient of

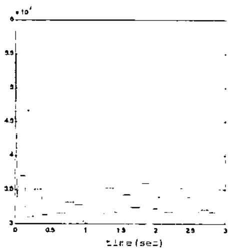

Fig. 4 Change of the parameter γ

Example 2. Consider the linear time-invariant system (16) with

$$A=\begin{bmatrix}-1 & -1 & -1\\ -1 & -2 & -1\\ -1 & -1 & -3\end{bmatrix},\quad B=\begin{bmatrix}-1 & 0 & 0\\ -1 & -1 & 0\\ -1 & -1 & -1\end{bmatrix}$$

The neural dynamics (25) was used to find the solution of the transformed LMI (22). Let $Q(0)=[0.1]_{3\times 3}$, $\eta_p=5000$, $\eta_r=20000$, $p_{\max}=2$, $\varepsilon=0.1$, $\sigma=1$. Figures 5(a)-5(c) illustrate, respectively, the convergent behavior of the activation states to the solution matrices Q, the slack matrix $\tilde{R}$ and the objective matrix G (only lower triangular elements were shown). The steady states of Q and $\tilde{R}$ were approximately obtained as

$$Q=\begin{bmatrix}0.983082 & -0.00443528 & -0.0188904\\ -0.00443528 & 0.963767 & -0.217216\\ -0.0188904 & -0.217216 & 0.295135\end{bmatrix}$$

$$\tilde{R}=\begin{bmatrix}1.70866 & 0 & 0\\ 1.57869 & 1.70866 & 0\\ 1.15939 & 0.186802 & 1.70866\end{bmatrix}$$

Clearly, the matrix Q is symmetric and positive definite, and satisfy the LMI (22). The steady control gain is then obtained as

$$K=\begin{bmatrix}0.562230 & 1.0925567 & 2.534235\\ 0.052782 & 1.0809352 & 2.493074\\ 0.041161 & 0.458701 & 2.034373\end{bmatrix}$$

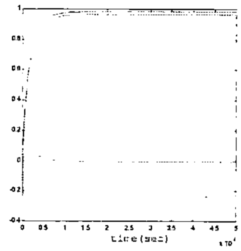

Fig. 5(a) Transient of Q

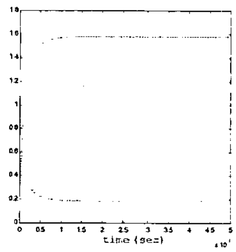

Fig. 5(b) Transient of $\tilde{R}$

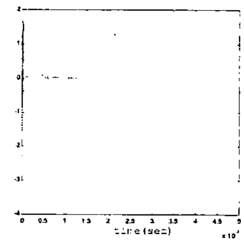

Fig. 5(c) Transient of G

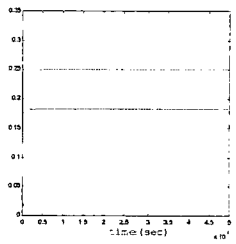

Fig. 6(a) Transient of P

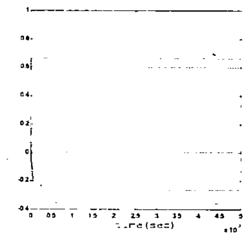

Fig. 6(b) Transient of $\tilde{R}_1$

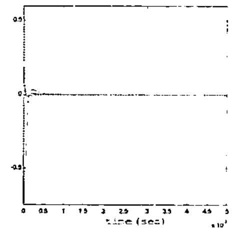

Fig. 6(c) Transient of G_1

Example 3. Consider the algebraic Riccati matrix inequality (26) with

$$A=\begin{bmatrix}-4 & 1\\ 0 & -7\end{bmatrix},\ B=\begin{bmatrix}1 & 1\\ 1 & 1\end{bmatrix},\ C=\begin{bmatrix}0 & 0\\ 0 & 0\end{bmatrix},$$

$$D^T+D=\begin{bmatrix}1 & 0\\ 0 & 1\end{bmatrix},\ Q=\begin{bmatrix}1 & 0\\ 0 & 3\end{bmatrix}$$

The algebraic Riccati matrix inequality is converted to the LMI (28) with

$$\bar{A}=\begin{bmatrix}1 & 0\\ 0 & 1\\ 0 & 0\\ 0 & 0\end{bmatrix},\ \bar{B}=\begin{bmatrix}-4 & 1 & 1 & 1\\ 0 & -7 & 1 & 1\end{bmatrix},\ \bar{D}=\begin{bmatrix}1 & 0 & 0 & 0\\ 0 & 3 & 0 & 0\\ 0 & 0 & -1 & 0\\ 0 & 0 & 0 & -1\end{bmatrix}$$

Let $P(0)=[0.1]_{2\times 2}$, $\eta_p=5000$, $\eta_r=50000$, $p_{s,\max}=1$, $\varepsilon=0.01$. The neural dynamics (28) was used to solve the algebraic Riccati matrix inequality (26). Figures 6(a)-6(c) illustrate, respectively, the convergent behavior of the activation states to the solution matrix P, the slack matrix $\tilde{R}_1$, and the objective matrix G_1. The steady states of P and $\tilde{R}_1$ were approximately obtained as

$$P=\begin{bmatrix}0.183057 & 0.0159725\\ 0.0158725 & 0.250297\end{bmatrix}$$

$$\tilde{R}_1 = \begin{bmatrix} 0.66343 & 0 & 0 & 0 \\ 0.00421687 & 0.66343 & 0 & 0 \\ -0.116196 & -0.143594 & 0.66343 & 0 \\ -0.264492 & -0.358474 & 0.601167 & 0.66343 \end{bmatrix}$$

Clearly, P is symmetric, positive definite, and satisfies the LMI (28).

5. CONCLUSIONS

We present a recurrent neural network approach solving for a class of LMIs that are commonly encountered in robust control system analysis and designs. The proposed networks have been proven to be stable and their steady states are the solution candidate to the LMIs. The solution process is parallel and distributed in neural computation. Therefore, it possesses the potential to be used for real-time control applications. It is shown that the developed approach can be easily extended to solve a wide class of LMIs. Illustrative examples are provided to demonstrate the operating characteristics of the neural network.

REFERENCES

[1] V. Balakrishnan and E. Feron (eds.), *Linear Matrix Inequalities in Control Theory and Applications*, special issue of the International Journal of Robust and Nonlinear Control, vol. 6, pp. 869-1099, 1996.

[2] B.R. Barmish, M. Coreless and G. Leitmann, "A new class of stabilizing controllers for uncertain dynamical systems," *SIAM J. Contr.*, vol. 21, pp. 246-255, 1983.

[3] S. Boyd, L.E. Ghaoui, E. Feron and V. Balakrishnan, *Linear Matrix Inequalities in System and Control Theory*, Philadelphia: SIAM, 1994.

[4] M. Chilali and P. Gahinet, "H_∞ design with pole placement constraints: an LMI approach," *Proc. 33rd IEEE Conf. Decision and Control*, pp.1511-1516, 1994.,

[5] A. Cichocki and R. Unbehauen, *Neural Networks for Optimization and Signal Processing*, New York: John Wiley & Sons, 1993.

[6] A. Cichocki, R. Unbehauen, K. Weinzierl and R. Holzel, "A new neural network for solving linear programming problems," *European J. Operational Research*, vol. 93, pp. 244-256, 1996.

[7] P. Gahinet, A. Nemirovskii, A.J. Laub and M. Chilali, *LMI Control Box*, MathWorks, 1995.

[8] R.A. Horn and C.R. Johnson, *Matrix Analysis*, New York: Cambridge University Press, 1985.

[9] H. Li and M. Fu, "A linear matrix inequality approach to robust H_∞ filtering," *IEEE Trans. Signal Processing*, vol. 5, pp. 2338-2350, 1997.

[10] X. Li and C.E. de Souza, "Delay-dependent robust stability and stabilization of uncertain linear delay systems· a linear matrix inequality approach," *IEEE Trans. Automa. Contr.*, vol. 42, pp.1144-1148, 1997.

[11] S. Limanond and J. Si, "Neural-network-based control design: an LMI approach," *IEEE Trans. Neural Networks*, vol. 9, pp. 1422-1429, 1998.

[12] C.L. Lin, "Synthesis of real-time robust flexible structure controllers," *AIAA J. Guid., Contr., Dyna.*, vol. 21, pp. 1001-1004, 1998.

[13] S. Sivrioglu and K. Nonami, "LMI approach to gain scheduled H_∞ control beyond PID control for gyroscopic rotor-magnetic bearing system," *Proc. 35th IEEE Conf. on Decision and Control*, pp. 3694-3699, 1996.

[14] G.W. Stewart, *Introduction to Matrix Computation*, Orlando: Academic Press, 1973.

[15] L. Vandenberghe and V. Balakrishnan, "Algorithms and software for LMI problems in control," *IEEE Contr. Syst. Mag.*, vol. 17, pp. 89-95, 1997.

[16] J. Wang, "Recurrent neural networks for solving linear matrix equalities," *Computers Math. Applic.*, vol. 26, pp. 23-34, 1993.

[17] J. Wang, "A deterministic annealing neural networks for convex programming," *Neural Networks*, vol. 7, pp. 629-641, 1994.

[18] H.O. Wang, K. Tanaka and M. Griffin, "An approach to control of nonlinear systems: stability and design issues," *IEEE Trans. Fuzzy Syst.*, vol. , pp. 14-23, 1996.

[19] L.X. Wang and J.M. Mendel, "Three-dimensional structured networks for matrix equation solving," *IEEE Trans. Computers*, vol. 40, pp. 1337-1346, 1991.

[20] Y. Xia, "A new neural network for solving linear and quadratic programming problems," *IEEE Trans. Neural Networks*, vo.7, pp. 1544-1547, 1996.

[21] J.M. Zurada, *Introduction to Artificial Neural Systems*, St. Paul: West publishing Co., 1992.