A new variant of the Zhang neural network for solving online time-varying linear inequalities

BY DONGSHENG GUO AND YUNONG ZHANG*

School of Information Science and Technology, Sun Yat-sen University, Guangzhou 510006, Guangdong, People’s Republic of China

Since March 2001, a special class of recurrent neural networks termed the Zhang neural network (ZNN) has been proposed by Zhang and co-workers for solving online a rich repertoire of time-varying problems. By extending Zhang et al.’s design formula (or say, the ZNN design formula), a (new) variant of the ZNN design formula is proposed and investigated in this paper, which is also based on a matrix/vector-valued indefinite error function. In addition, by employing such a novel design formula, a new variant of the ZNN (NVZNN) is proposed, developed and investigated for online solution of time-varying linear inequalities (LIs). The resultant NVZNN models are depicted in implicit dynamics and methodologically exploit the time-derivative information of time-varying coefficients. Computer simulation results further demonstrate the novelty, efficacy and superiority of the proposed NVZNN models for solving online time-varying LIs.

Keywords: new variant of the Zhang neural network (NVZNN); time-varying linear inequalities; design formula; implicit dynamics; solution set

1. Introduction

In recent years, online solution of linear inequalities (LIs) has been considered to be an important problem encountered in numerous fields of science and engineering applications (Yang et al. 1992; Xia 1996; Cichocki & Bargiela 1997; Labonte 1997; Xia et al. 1999; Lin et al. 2000; Zhang 2006). For example, an inequality-based criterion/constraint has been recently proposed, introduced and investigated by Zhang et al. for obstacle avoidance of redundant robot manipulators (Zhang & Wang 2004), which is used to generate escape velocities of variable magnitude, and then drive the affected links away from obstacles. Thus, robot manipulators can avoid obstacles successfully by solving such linear matrix–vector inequalities (LMVIs) in real time $t$ (Zhang & Wang 2004).

Generally speaking, a MATLAB toolbox (Gahinet et al. 1995; Lin et al. 2000) can be used to solve linear matrix inequalities (LMIs), which is viewed as a sub-case in the research of LI solving. Because of the limitation of such a toolbox, two classes of well-developed approaches have been proposed and investigated for solving LIs. One of the classes transforms LIs into optimization problems (e.g. linear programmes), which are then solved by using classical methods, e.g. the simplex method and the penalty method (Xia et al. 1999; Lin et al. 2000).

Received 8 November 2011
Accepted 16 February 2012

This journal is © 2012 The Royal Society
However, these methods require matrix operations that are often impractical for large-scale systems. The second class of approaches is based on iterative methods, one of which is derived from the relaxation method. Most do not need matrix manipulations, thus, the basic computation step is simple and easy to code. Unfortunately, the amount of work for the iterative approach is proportional to the cube of the number of constraints in the group (Yang et al. 1992). Consequently, when solving a large-scale system that may have thousands of constraints, it may not be effective and leads to slow convergence (Yang et al. 1992).

Because of the parallel distributed nature and the suitability for hardware realization, neural networks have been proposed, developed and studied for scientific research and engineering applications (Ramasamy et al. 1995; Cichocki & Bargiela 1997; Labonte 1997; Xia et al. 1999; Lin et al. 2000; Liao et al. 2002; Zhang et al. 2002; Novo et al. 2005; Zhang & Ge 2005; Chetwynd et al. 2006; Yu et al. 2007; Liu 2009; Zhang & Li 2009; Li & Zhang 2010; Guo et al. 2011; Kaslik & Sivasundaram 2011), drawing extensive interest and investigation by researchers. Recently, owing to in-depth research on neural networks, the artificial neural-dynamic approach based on recurrent neural networks has been viewed as a powerful alternative to online solution of LMIs as well as LMVIs. For example, Lin et al. presented a recurrent neural-network approach solving a class of LMIs that is commonly encountered in robust control-system analysis and design (Lin et al. 2000). Cichocki & Bargiela (1997) developed three continuous-time neural networks for solving LI systems. It is worth pointing out that most reported neural-network models are related to gradient-based methods and/or designed theoretically for solving time-invariant (or termed static) LIs. The gradient/gradient-based neural network (GNN) is based on a gradient-descent method to minimize a norm-based lower-bounded energy function, which has already been employed comprehensively to solve such static problems. However, when such a GNN model is applied to solving LIs in the situation of having time-varying coefficients (e.g. the aforementioned time-varying LMVIs for obstacle avoidance of redundant robot manipulators (Zhang & Wang 2004)), a faster convergence rate is often required when compared with the variational rate of time-varying coefficients. This may thus impose stringent restrictions on the GNN’s physical realization and/or sacrifice the solution precision.

Since March 2001, a special class of recurrent neural networks termed the Zhang neural network (ZNN) has been proposed by Zhang and co-workers for solving time-varying problems (Zhang et al. 2002; Zhang & Ge 2005; Zhang & Li 2009; Li & Zhang 2010; Guo et al. 2011), such as time-varying Sylvester equation solving and time-varying matrix inversion. The design of such a ZNN (i.e. the so-called ZNN design formula) is based on a matrix/vector-valued lower-unbounded error function (instead of the conventional scalar-valued lower-bounded energy function that is usually involved in the gradient-descent method and Hopfield neural networks), which aims at making every element of the error function globally and exponentially converge to zero. The resultant ZNN models depicted in implicit dynamics exploit methodologically the time-derivative information of time-varying coefficient(s), and thus solve the time-varying problems effectively and efficiently (Zhang et al. 2002; Zhang & Ge 2005; Zhang & Li 2009; Li & Zhang 2010; Guo et al. 2011). Note that the abovementioned ZNN design formula is presented for solving time-varying equations, but it cannot
be used directly to solve time-varying LIs by defining the corresponding simple matrix/vector-valued error functions. Therefore, it is necessary to investigate a (new) variant of the ZNN design formula used to solve time-varying LIs directly.

By extending Zhang et al.’s design formula (i.e. the ZNN design formula), a (new) variant of the ZNN design formula is proposed and investigated in this paper, which is also based on a matrix/vector-valued lower-unbounded error function. In addition, by employing such a novel design formula, the new variant of ZNN (NVZNN) is proposed, developed and investigated for online solution of time-varying LIs. The resultant NVZNN models are depicted in implicit dynamics and methodologically exploit the time-derivative information of time-varying coefficients. Thus, the elements of the error function with their initial errors being greater than zero can globally and exponentially converge to zero, while the rest of the elements of the error function (i.e. the elements with their initial errors being less than or equal to zero) can always equal their initial errors. Simply put, the proposed NVZNN models can generate the exact time-varying solution of time-varying LIs. Computer simulation results further demonstrate the novelty, efficacy and superiority of the proposed NVZNN models for solving time-varying LIs online.

The remainder of this paper is organized into four sections. In §2, the NVZNN design formula is proposed and investigated for time-varying LIs. Section 3 presents the NVZNN models for solving time-varying LMIs, together with the corresponding theoretical analysis and simulation results. In §4, the NVZNN models for solving time-varying LMVIs are presented and the related simulation results are illustrated for verification. Section 5 concludes this paper with final remarks. Before ending this section, it is worth pointing out the main contributions of this paper as follows.

— This paper investigates the online solution of time-varying LIs, rather than the conventionally investigated solving of static LIs. To the best of the authors’ knowledge, there are almost no researches on time-varying LI solving, especially time-varying LMI solving.
— A novel NVZNN design formula is proposed and investigated in this paper, which is based on a matrix/vector-valued lower-unbounded error function (instead of the conventional scalar-valued lower-bounded energy function).
— By exploiting the NVZNN design formula, the NVZNN models are proposed, developed and investigated in this paper for online solution of time-varying LMIs/LMVIs.
— Theoretical analysis guarantees that the NVZNN models can generate the exact time-varying solution of time-varying LIs.
— Simulative results are illustrated to demonstrate the novelty, efficacy and superiority of the proposed NVZNN models for solving online time-varying LMIs/LMVIs.

2. The new variant of the Zhang neural network design formula for time-varying linear inequalities

In this section, the NVZNN design formula is proposed and investigated for solving online time-varying LIs.
(a) Scalar-valued new variant of the Zhang neural network design formula

For better readability and understanding, let us consider the scalar-valued time-varying LI in the form of \( f(t) \leq 0 \in \mathbb{R} \), with \( f(\cdot) \) being a differentiable function. In the NVZNN design methodology, we firstly define the scalar-valued lower-unbounded error function as

\[
e(t) \triangleq f(t) \in \mathbb{R}, \quad t \in [0, +\infty).
\]

Then, based on the ZNN design formula (Zhang et al. 2002; Zhang & Ge 2005; Zhang & Li 2009; Guo et al. 2011), the time derivative \( \dot{e}(t) \) of such an error function \( e(t) \) is chosen via the following form, which is the NVZNN design formula proposed in this paper for solving online \( f(t) \leq 0 \):

\[
\dot{e}(t) = -\gamma \text{sgn}(e(0)) e(t), \tag{2.1}
\]

where \( \gamma > 0 \in \mathbb{R} \), the reciprocal of a capacitance parameter, is used to scale the convergence rate of the solution. Generally speaking, such a design parameter \( \gamma \) should be set as large as the hardware system would permit, or selected appropriately for simulative/experimental purposes (Mead 1989). In addition, \( e(0) \) denotes the initial error, and the unipolar signum function \( \text{sgn}(\cdot) \) is defined as

\[
\text{sgn}(e(0)) = \begin{cases} 
1, & \text{if } e(0) > 0, \\
0, & \text{if } e(0) \leq 0.
\end{cases}
\]

**Theorem 2.1.** For the proposed scalar-valued NVZNN design formula (2.1): (i) if the initial error \( e(0) > 0 \), the error function \( e(t) \) is globally and exponentially convergent to zero and (ii) if the initial error \( e(0) \leq 0 \), the error function \( e(t) \) is always equal to the initial error \( e(0) \).

**Proof.** Based on the unipolar signum function \( \text{sgn}(\cdot) \), the NVZNN design formula (2.1) is rewritten as

\[
\dot{e}(t) = \begin{cases} 
-\gamma e(t), & \text{if } e(0) > 0, \\
0, & \text{if } e(0) \leq 0.
\end{cases}
\]

Evidently, (i) when \( e(0) > 0 \), \( \dot{e}(t) = -\gamma e(t) \), of which the solution is analytically \( e(t) = \exp(-\gamma t) e(0) \), and (ii) when \( e(0) \leq 0 \), \( \dot{e}(t) = 0 \), of which the solution is analytically \( e(t) = e(0) \). Therefore, if the initial error \( e(0) > 0 \), then the error function \( e(t) = \exp(-\gamma t) e(0) \) globally and exponentially converges to zero (i.e. as \( t \to +\infty \), \( e(t) \to 0 \) globally and exponentially); if the initial error \( e(0) \leq 0 \), then the error function \( e(t) \) is always equal to \( e(0) \) (i.e. \( e(t) = e(0) \) no matter how time \( t \) evolves). The proof is thus completed. \( \blacksquare \)

By exploiting the scalar-valued NVZNN design formula (2.1), the resultant NVZNN model (e.g. the one shown in appendix 1) can thus be developed for online solution of the scalar-valued time-varying LI \( f(t) \leq 0 \). In addition, it follows from theorem 2.1 that the corresponding error function \( e(t) \) globally and exponentially converges to zero for any initial state (of the NVZNN model) outside the solution set of \( f(t) \leq 0 \) (i.e. \( f(0) = e(0) > 0 \)). Otherwise (i.e. starting with any initial state inside
the solution set), the error function \( c(t) \) remains \( c(0) \leq 0 \). These imply that the resultant NVZNN model (by exploiting (2.1)) can generate an exact time-varying solution of the scalar-valued time-varying LI \( f(t) \leq 0 \).

(b) Matrix/vector-valued new variant of the Zhang neural network design formula

On the basis of the above analysis, corresponding to the problem of time-varying LMIs/LMVIs \( F(t) \leq 0 \in R^{m \times n} \) (with all elements of \( F(\cdot) \) being differentiable) to be solved, the following NVZNN design formula is further proposed, with the matrix/vector-valued error function defined as \( E(0) \triangleq F(t) \in R^{m \times n} \),

\[
\dot{E}(t) = -\gamma \text{SGN}(E(0)) \odot E(t),
\]

where \( \dot{E}(t) \) is the time derivative of the error function \( E(t) \), \( E(0) \) denotes the initial error and \( \gamma \) is defined the same as before. In addition, \( \text{SGN}(\cdot): R^{m \times n} \rightarrow R^{m \times n} \) denotes the unipolar signum function array, and the \( ij \)th element of \( \text{SGN}(E(0)) \) is \( \text{sgn}(e_{ij}(0)) \) \( (i \in \{1, 2, \ldots, m\} \) and \( j \in \{1, 2, \ldots, n\} \) \) with \( e_{ij}(0) \) being the \( ij \)th element of \( E(0) \) and \( \text{sgn}(\cdot) \) defined the same as before. Furthermore, the matrix–matrix multiplication operator \( \odot \) is the Hadamard product (Liu & Trenkler 2008) and is defined as

\[
U \odot W = \begin{bmatrix}
   u_{11}w_{11} & u_{12}w_{12} & \cdots & u_{1n}w_{1n} \\
   u_{21}w_{21} & u_{22}w_{22} & \cdots & u_{2n}w_{2n} \\
   \vdots & \vdots & \ddots & \vdots \\
   u_{m1}w_{m1} & u_{m2}w_{m2} & \cdots & u_{mn}w_{mn}
\end{bmatrix} \in R^{m \times n}.
\]

**Theorem 2.2.** For the proposed matrix/vector-valued NVZNN design formula (2.2), (i) if the initial error \( E(0) > 0 \in R^{m \times n} \) (i.e. each element of \( E(0) \) is greater than zero), the error function \( E(t) \) is always equal to their initial errors, and (ii) if the initial error \( E(0) \leq 0 \) (i.e. each element of \( E(0) \) is less than or equal to zero), the error function \( E(t) \) always equals the initial error \( E(0) \), and (iii) the elements of the error function \( E(t) \) with their initial errors being greater than zero are globally and exponentially convergent to zero, while the rest of the elements of \( E(t) \) (i.e. with their initial errors being less than or equal to zero) are always equal to their initial errors.

**Proof.** For the NVZNN design formula (2.2), its compact form of a set of \( mn \) decoupled differential equations can be written as

\[
\dot{e}_{ij}(t) = -\gamma \text{sgn}(e_{ij}(0))e_{ij}(t),
\]

for any \( i \in \{1, 2, \ldots, m\} \) and \( j \in \{1, 2, \ldots, n\} \). Evidently, it follows from theorem 2.1 that \( e_{ij}(t) \) globally and exponentially converges to zero when \( e_{ij}(0) > 0 \), or if \( e_{ij}(0) \leq 0 \). Therefore, (i) if \( e_{ij}(0) > 0 \) for all the \( i \) and \( j \) (i.e. \( E(0) > 0 \)), the error function \( E(t) \) is globally and exponentially convergent to zero, (ii) if \( e_{ij}(0) \leq 0 \) for all the \( i \) and \( j \) (i.e. \( E(0) \leq 0 \)), the error function \( E(t) \) is always equal to the initial error \( E(0) \), and (iii) the elements of the error function \( E(t) \) with their initial errors being greater than zero (i.e. corresponding to the case with \( e_{ij}(0) > 0 \)) are globally and exponentially convergent to zero, while the rest of the elements of \( E(t) \) (i.e. corresponding to the case with \( e_{ij}(0) \leq 0 \)) are always equal to their initial errors. The proof is thus completed.

\[\blacksquare\]
As we know, unexpected nonlinearities and implementation errors always exist. Even if the linear activation functions are used, the nonlinear phenomena may appear in the hardware implementations. Thus, based on the previous work (Zhang et al. 2002; Zhang & Ge 2005; Zhang & Li 2009; Li & Zhang 2010; Guo et al. 2011), the (nonlinear) activation function array $\Phi(\cdot): R^{m \times n} \rightarrow R^{m \times n}$ is introduced and used for the construction of the NVZNN design formula. The generalized NVZNN design formula is then proposed as

$$\dot{E}(t) = -\gamma \text{SGN}(E(0)) \odot \Phi(E(t)).$$  

(2.3)

Generally speaking, the function $\phi(\cdot)$, being an element of $\Phi(\cdot)$, can be any monotonically increasing odd activation function, such as linear, power-sigmoid and power-sum activation functions that the authors have discussed and investigated since 2001. In this sense, (2.3) reduces to (2.2) when using a linear activation function array. In other words, (2.2) is viewed as a special case of (2.3) (which is the reason why we mentioned ‘generalized NVZNN design formula’ in the previous paragraph). Hereafter, in this paper, (2.3) is called the NVZNN design formula for convenience of presentation (note that (2.1) is also viewed as a special case of (2.3) when we focus on the scalar-valued case). By exploiting the NVZNN design formula (2.3), the corresponding NVZNN models can be established for solving time-varying LMIs/LMVIs (as well as the scalar-valued time-varying LI).

**Remark 2.3.** It is worth pointing out that, when the initial error $E(0) > 0 \in R^{m \times n}$, the proposed NVZNN design formula (2.3) reduces to $\dot{E}(t) = -\gamma \Phi(E(t))$, which is the so-called ZNN design formula for solving online time-varying matrix/vector/scalar equations (Zhang et al. 2002; Zhang & Ge 2005; Zhang & Li 2009; Li & Zhang 2010; Guo et al. 2011). In this situation, different choices of activation function arrays may lead to different convergence performances of the error function $E(t)$. For example, when a linear activation function array is used, (2.3) reduces to $\dot{E}(t) = -\gamma E(t)$ (with $E(0) > 0$), for which the solution is analytic $E(t) = \exp(-\gamma t)E(0)$. Evidently, as $t \to \infty$, $E(t) \to 0$ globally and exponentially with the convergence rate $\gamma > 0$. Based on the previous work (Zhang et al. 2002; Zhang & Ge 2005; Zhang & Li 2009; Li & Zhang 2010; Guo et al. 2011), superior convergence properties of the error function $E(t)$ can be achieved by using nonlinear activation functions (e.g. power-sigmoid activation functions). In addition, the convergence property of the error function $E(t)$ can be further improved by increasing the value of the design parameter $\gamma$. Thus, the global and exponential convergence property of the error function $E(t)$ is guaranteed, when the initial error $E(0) > 0$. Note that, when the initial error $E(0) \leq 0$, (2.3) reduces to $\dot{E}(t) = 0$, and thus there is no need to investigate the convergence performance of the error function $E(t)$ by using different activation functions and different values of $\gamma$.

**Remark 2.4.** By understanding the nature of time-varying LIs, the SGN(\cdot) function array is exploited in the original ZNN design formula (Zhang et al. 2002; Zhang & Ge 2005; Zhang & Li 2009; Li & Zhang 2010; Guo et al. 2011), and the NVZNN design formula is thus proposed and investigated in this paper for solving online time-varying LIs. Note that, by exploiting the SGN(\cdot) function array, the resultant NVZNN design formula (2.3) is also based on a matrix/vector-valued
lower-unbounded error function. More importantly, the (nonlinear) activation function array is introduced and used for the construction of such a NVZNN design formula (to improve the convergence performances of the corresponding NVZNN models). In addition, the proposed NVZNN design formula (2.3) is not only suitable for the particular case (i.e. LI solving), but is also applicable for nonlinear matrix inequalities solving (which is an interesting direction for future work).

3. Time-varying linear matrix inequalities

Recently, LMI approaches have been proposed, developed and studied for solving lots of problems arising in numerous fields of science and engineering (Liao et al. 2002; Yu et al. 2007; Liu 2009). That is, these science and engineering problems can be reformulated as the corresponding LMI problems, which are then solved via numerical algorithms and/or neural networks. They have been viewed as powerful formulation and design techniques for solving a variety of problems (Lin et al. 2000). Thus, it is worth investigating the online solution of the LMI problem. In this section, the proposed NVZNN design formula (2.3) is exploited to solve online the LMI problem with time-varying coefficients.

(a) The new variant Zhang neural network model

Let us consider the following problem of time-varying LMIs:

\[ A(t)X(t) \leq B(t), \quad (3.1) \]

where \( A(t) \in R^{n \times n} \) and \( B(t) \in R^{n \times n} \) are smoothly time-varying matrices, and \( X(t) \in R^{n \times n} \) is the unknown matrix to be obtained. In addition, let us define the time-varying solution set \( \mathcal{Q}(t) = \{ X(t)| X(t) \in R^{n \times n} \text{ solves } (3.1) \} \). It is worth mentioning that (3.1) is a representative time-varying LMI problem that is studied here. The proposed NVZNN design formula (2.3) can be directly extended to solve other types of time-varying LMIs (e.g. time-varying Lyapunov matrix inequalities).

To monitor and control the process of solving the time-varying LMI (3.1), we firstly define the following matrix-valued error function:

\[ E(t) = A(t)X(t) - B(t) \in R^{n \times n}. \]

Then, by following the proposed NVZNN design formula (2.3), the NVZNN model is established as

\[ A(t)\dot{X}(t) = - \dot{A}(t)X(t) + \dot{B}(t) - \gamma\text{SGN}(E(0)) \odot \Phi(E(t)), \quad (3.2) \]

where \( X(t) \in R^{n \times n} \), starting from any initial state \( X(0) \in R^{n \times n} \), denotes the neural-network state matrix, and the initial error \( E(0) = A(0)X(0) - B(0) \). In addition, \( \gamma, \text{SGN}(\cdot), \Phi(\cdot) \) and \( \odot \) are defined the same as before. Furthermore, \( \dot{A}(t), \dot{B}(t) \) and \( \dot{X}(t) \) denote the time derivatives of \( A(t), B(t) \) and \( X(t) \), respectively.

For better readability and potential hardware implementation, the block diagram
of the NVZNN model (3.2) is illustrated in figure 1. It is worth mentioning that, if a linear activation function array is used, the NVZNN model (3.2) reduces to

$$A(t)\dot{X}(t) = -\dot{A}(t)X(t) + \dot{B}(t) - \gamma \text{SGN}(E(0)) \odot E(t).$$

(b) Theoretical analysis

In this subsection, theoretical analysis of the NVZNN model (3.2) for solving the time-varying LMI (3.1) is given via the following theorem.

**Theorem 3.1.** Given a smoothly time-varying non-singular coefficient matrix $A(t)$ and a smoothly time-varying coefficient matrix $B(t)$ in (3.1), the proposed NVZNN model (3.2) generates an exact time-varying solution of $A(t)X(t) \leq B(t)$.

**Proof.** Consider the NVZNN model (3.2), which is derived from the proposed NVZNN design formula (2.3). Therefore, there are three situations as follows.

— If the randomly generated initial state $X(0) \in \mathbb{R}^{n \times n}$ is outside the initial solution set $\mathcal{Q}(0)$ of (3.1), i.e. $E(0) = A(0)X(0) - B(0) > 0$, based on theorem 2.2 and remark 2.3, the matrix-valued error function $E(t)$ can globally and exponentially converge to zero. This implies that the neural state $X(t)$ of the NVZNN model (3.2) globally and exponentially converges to the solution of matrix equation $A(t)X(t) - B(t) = 0$ (since $E(t) = A(t)X(t) - B(t)$). Note that $A(t)X(t) - B(t) = 0$ is a special case of the time-varying LMI (3.1). Therefore, the proposed NVZNN model (3.2) generates an exact time-varying solution, i.e. the neural state $X(t)$ globally and exponentially converges to the solution set $\mathcal{Q}(t)$ of (3.1).

— If the randomly generated initial state $X(0) \in \mathbb{R}^{n \times n}$ is inside the initial solution set $\mathcal{Q}(0)$ of (3.1), i.e. $E(0) = A(0)X(0) - B(0) \leq 0$, based on theorem 2.2, the matrix-valued error function $E(t)$ remains $E(0)$. In other words, no matter how time $t$ evolves, $E(t) = E(0) \leq 0$. This implies that...
the neural state $X(t)$ of the proposed NVZNN model (3.2) will always stay inside the time-varying solution set $\Omega(t)$ of (3.1).

— If some elements of the randomly generated initial state $X(0) \in \mathbb{R}^{n \times n}$ are inside the initial solution set $\Omega(0)$ of (3.1), while the others are outside $\Omega(0)$, i.e. some elements of $E(0) = A(0)X(0) - B(0)$ are greater than zero while the rest of the elements of $E(0)$ are less than or equal to zero, based on theorem 2.2, the elements of the error function $E(t)$ with their initial errors being greater than zero can globally and exponentially converge to zero, while the rest elements of $E(t)$ can always equal their initial errors. This implies that the proposed NVZNN model (3.2) generates an exact time-varying solution of (3.1) in the sense that each element of the error function $E(t)$ is less than or equal to zero.

In summary, based on the above analysis, the proposed NVZNN model (3.2) generates an exact time-varying solution of $A(t)X(t) \leq B(t)$. The proof is thus completed.

(c) Simulation verification

In this subsection, computer simulations are performed to verify the efficacy of the proposed NVZNN model (3.2) (as well as the NVZNN design formula (2.3)) for online solution of the time-varying LMI (3.1). It is worth pointing out that the power-sigmoid activation function array (Zhang et al. 2002; Zhang & Ge 2005; Zhang & Li 2009) is exploited in the simulations.

For illustration and verification, let us consider the time-varying LMI (3.1) with the following time-varying coefficient matrices:

$$A(t) = \begin{bmatrix} \sin(10t) & \cos(10t) \\ -\cos(10t) & \sin(10t) \end{bmatrix} \in \mathbb{R}^{2 \times 2}$$

and

$$B(t) = \begin{bmatrix} \cos(10t) + 1 & \sin(10t) + 1.5 \\ -\sin(10t) + 1.5 & -\cos(10t) + 1 \end{bmatrix} \in \mathbb{R}^{2 \times 2}.$$

Considering that different initial states of the proposed NVZNN model (3.2) may result in different convergent performances, we investigate the following three cases.

Case 1. If the initial state $X(0) \in \mathbb{R}^{2 \times 2}$ is outside the initial solution set $\Omega(0)$ of (3.1), by applying the NVZNN model (3.2) with $\gamma = 1$ to solving the time-varying LMI (3.1), the corresponding simulation results are illustrated in figure 2. As seen from figure 2a, starting from a randomly generated initial state $X(0) \notin \Omega(0)$, the neural state $X(t)$ of the NVZNN model (3.2) converges to a time-varying solution. In addition, as shown in figure 2b, each element of the error function $E(t) = A(t)X(t) - B(t)$ converges to zero, which implies that the neural state $X(t)$ of the NVZNN model (3.2) (illustrated in figure 2a) is exactly convergent to the theoretical solution of the time-varying linear matrix equation $A(t)X(t) - B(t) = 0$ (in the sense that $E(t) = A(t)X(t) - B(t)$). Note that such a time-varying linear matrix equation is a special case of the time-varying LMI (3.1). These demonstrate that the proposed NVZNN model (3.2) can generate an exact time-varying solution of (3.1), though the initial state of (3.2) is outside the initial solution set $\Omega(0)$.
For further investigation, the proposed NVZNN model (3.2) is exploited to solve the time-varying LMI (3.1) by using different values of \( g \) (i.e. \( g = 10 \) and \( g = 100 \)), and the corresponding simulation results are shown in figure 3. As seen from figure 3 (as well as figure 2b), the performance of the NVZNN model (3.2) is improved very effectively by increasing the value of \( g \). These demonstrate the efficacy of the proposed NVZNN model (3.2) for solving online the time-varying LMI (3.1).

Case 2. If the initial state \( X(0) \) is inside the initial solution set \( \Omega(0) \), from the previous analysis, we know that the error function \( E(t) = A(t)X(t) - B(t) \) remains \( E(0) = A(0)X(0) - B(0) \leq 0 \) without an appreciable convergence process. Figure 4 illustrates the simulation results synthesized by the proposed NVZNN model (3.2) with \( \gamma = 1 \) and \( X(0) \in \Omega(0) \). As shown in figure 4, the solution of the
NVZNN model (3.2) is time varying, while the corresponding error function is always equal to the initial error $E(0)$. This further demonstrates the efficacy of the NVZNN model (3.2) for solving online the time-varying LMI (3.1), i.e. (3.2) can generate an exact time-varying solution of (3.1).

Case 3. Generally speaking, it may be difficult to know whether the initial state $X(0)$ used for simulation/application is outside $\Omega(0)$ or not. Thus, it is worth investigating the performance of the proposed NVZNN model (3.2) for online solution of the time-varying LMI (3.1) when some elements of $X(0)$ are outside $\Omega(0)$, while the others are inside $\Omega(0)$. In this situation, some elements of the initial error $E(0)$ are greater than zero, while the rest are less than or equal
to zero. The corresponding simulation results are illustrated in figure 5. From figure 5a, we can see again that the neural state $X(t)$ of the NVZNN model (3.2) is time varying. In addition, as shown in figure 5b, the errors $e_{11}(t)$ and $e_{21}(t)$ converge to zero, and the errors $e_{12}(t)$ and $e_{22}(t)$ are always equal to $e_{12}(0) < 0$ and $e_{22}(0) < 0$. These imply that the proposed NVZNN model (3.2) generates an exact time-varying solution of (3.1).

In summary, the above simulation results (i.e. figures 2–5) have demonstrated the novelty, efficacy and superiority of the proposed NVZNN model (3.2) (as well as the NVZNN design formula (2.3)) for online solution of time-varying LMIs (e.g. (3.1)).

4. Time-varying linear matrix–vector inequalities

In recent years, the online solution of LMVIs (viewed as a special case of LMIs) has been considered to be an important problem encountered in the science and engineering fields (Xia et al. 1999; Zhang & Wang 2004; Zhang 2006) (e.g. online solution of time-varying inequality constraint for obstacle avoidance of redundant robot manipulators). In this section, we consider the following time-varying LMVIs:

$$A(t)x(t) \leq b(t), \quad (4.1)$$

where $A(t) \in \mathbb{R}^{n \times n}$ and $b(t) \in \mathbb{R}^n$ are, respectively, the smoothly time-varying matrix and vector, and $x(t) \in \mathbb{R}^n$ is the unknown vector to be obtained.

Based on the analysis in the previous sections, we can firstly define the vector-valued error function $E(t) \in \mathbb{R}^n$ as

$$E(t) = A(t)x(t) - b(t).$$

Then, by exploiting the NVZNN design formula (2.3), we have the following NVZNN model for online solution of the time-varying LMVI (4.1):

$$A(t)\dot{x}(t) = -\dot{A}(t)x(t) + \dot{b}(t) - \gamma \text{SGN}(E(0)) \odot \Phi(E(t)), \quad (4.2)$$

where $\dot{A}(t)$, $\dot{b}(t)$ and $\dot{x}(t)$ denote the time derivatives of $A(t)$, $b(t)$ and $x(t)$, respectively. In addition, the initial error $E(0) = A(0)x(0) - b(0)$. Furthermore, $\gamma$, $\text{SGN}(\cdot)$, $\Phi(\cdot)$ and $\odot$ are defined the same as before. For better understanding and potential hardware implementation, the block diagram of the NVZNN model (4.2) is illustrated in figure 6. Note that, as the online solution of the time-varying LMVI (4.1) can be viewed as a special case of that of the time-varying LMI (3.1), the block diagram shown in figure 6 is thus quite similar to that shown in figure 1. In addition, by using a linear activation function array, the NVZNN model (4.2) reduces to

$$A(t)\dot{x}(t) = -\dot{A}(t)x(t) + \dot{b}(t) - \gamma \text{SGN}(E(0)) \odot E(t).$$

Based on the previous analysis, the following theorem on the performance/property of the NVZNN model (4.2) for online solution of the time-varying LMVI (4.1) can be summarized and presented.

**Theorem 4.1.** Given a smoothly time-varying non-singular coefficient matrix $A(t)$ and a smoothly time-varying coefficient vector $b(t)$ in (4.1), the proposed NVZNN model (4.2) generates an exact time-varying solution of $A(t)x(t) \leq b(t)$. 

Proof. The proof can be generalized from the proof of theorem 3.1 given in the previous section (and thus omitted here due to space limitation).

For illustration and verification, the NVZNN model (4.2) is exploited to solve the time-varying LMVI (4.1) with coefficients \( A(t) \) and \( b(t) \),

\[
A(t) = \begin{bmatrix}
3 + \sin(5t) & \frac{\cos(5t)}{2} & \frac{\cos(5t)}{2} \\
\frac{\cos(5t)}{2} & 3 + \sin(5t) & \frac{\cos(5t)}{2} \\
\frac{\cos(5t)}{2} & \frac{\cos(5t)}{2} & 3 + \sin(5t)
\end{bmatrix} \in \mathbb{R}^{3 \times 3}
\]

and

\[
b(t) = \begin{bmatrix}
\sin(5t) + 1 \\
\cos(5t) + 2 \\
\sin(5t) + \cos(5t) + 3
\end{bmatrix} \in \mathbb{R}^{3}.
\]

Figures 7 and 8 show the simulation results synthesized by the proposed NVZNN model (4.2) with \( \gamma = 1 \) and using the power-sigmoid activation function array. As shown in figure 7, starting from five randomly generated initial states \( x(0) \), the neural states of the NVZNN model (4.2) are time varying. In addition, as seen from figure 8, the elements of the error function \( E(t) = A(t)x(t) - b(t) \), which are greater than zero, converge to zero, while those less than or equal to zero remain at their initial value. These demonstrate the efficacy of the proposed NVZNN model (4.2) for online solution of time-varying LMVIs. In other words, the proposed NVZNN model (4.2) can generate an exact time-varying solution of time-varying LMVI \( A(t)x(t) \leq b(t) \).

Remark 4.2. As presented previously, both the proposed NVZNN models (3.2) and (4.2) are depicted in implicit dynamics (e.g. \( A(t)\dot{X}(t) = \cdots \)), which coincide well with systems in nature and in engineering practice (e.g. in analogue electronic circuits and mechanical systems owing to Kirchhoff’s and Newton’s
By extending Zhang et al.'s design formula (i.e. the ZNN design formula),
the NVZNN design formula (2.3) has been proposed and investigated in this
paper for solving online time-varying LIs. Based on such a novel design formula,
the NVZNN models have been proposed, developed and investigated for online
solution of time-varying linear (LMI/LMVIs) inequalities (i.e. (3.1) and (4.1)).
The resultant NVZNN models (3.2) and (4.2) depicted in implicit dynamics
have exploited methodologically the time-derivative information of time-varying
coefficients (e.g. $\dot{A}(t)$, $\dot{B}(t)$ and $\dot{b}(t)$). In addition, theoretical analyses have been
given on the convergence properties of the proposed NVZNN models. Computer
simulation results have further demonstrated the novelty, efficacy and superiority

5. Conclusions
of the proposed NVZNN models (as well as the NVZNN design formula) for solving online time-varying LMI/LMVs. That is, the proposed NVZNN models can generate an exact time-varying solution of time-varying linear (LMI/LMVs) inequalities.

Appendix A

For further investigation and completeness, the proposed NVZNN design formula (2.3) is exploited to solve the scalar-valued time-varying LI in the form of

$$a(t)x(t) \leq b(t), \quad (A\ 1)$$

where \(a(t) \in R\) and \(b(t) \in R\) are smoothly time-varying scalars, and \(x(t) \in R\) is the unknown scalar to be obtained. Evidently, by defining the error function \(e(t) = a(t)x(t) - b(t)\) and exploiting the NVZNN design formula (2.3), the following NVZNN model is established for online solution of the scalar-valued time-varying LI (A 1):

$$a(t)\dot{x}(t) = -\dot{a}(t)x(t) + \dot{b}(t) - \gamma \text{sgn}(e(0))\phi(e(t)), \quad (A\ 2)$$

where \(\dot{a}(t), \dot{b}(t)\) and \(\dot{x}(t)\) denote the time derivatives of \(a(t), b(t)\) and \(x(t)\), respectively. In addition, the initial error \(e(0) = a(0)x(0) - b(0)\). Furthermore, \(\gamma\), \(\text{sgn}(\cdot)\) and \(\phi(\cdot)\) are defined the same as before. Moreover, the NVZNN model (A 2) reduces to the following form when using a linear activation function:

$$a(t)\dot{x}(t) = -\dot{a}(t)x(t) + \dot{b}(t) - \gamma \text{sgn}(e(0))e(t).$$

For illustration and verification, the proposed NVZNN model (A 2) is exploited to solve (A 1) with coefficients

\[ a(t) = \cos(5t) + 1.5 \quad \text{and} \quad b(t) = 0.5 \sin(5t). \]
Figure 9 illustrates the simulation results synthesized by the NVZNN model (A2) by the using the power-sigmoid activation function and different values of $\gamma$. As seen from the figure, the proposed NVZNN model (A2) can generate an exact time-varying solution of (A1), no matter what the initial state $x(0)$ of (A2) is set at (inside or outside the time-varying solution set of (A1)). This demonstrates the efficacy of the proposed NVZNN model (A2) for online solution of the scalar-valued time-varying LI (e.g. (A1)).

This work was supported by the National Natural Science Foundation of China under grants 61075121 and 60935001, and also by the Fundamental Research Funds for the Central Universities of China.

References


