A new delayed projection neural network for solving quadratic programming problems with equality and inequality constraints

Chunlin Sha, Hongyong Zhao *, Fengli Ren

Department of Mathematics, Nanjing University of Aeronautics and Astronautics, Nanjing 210016, People’s Republic of China

Article Info

A R T I C L E   I N F O

Article history:
Received 16 December 2014
Received in revised form 13 March 2015
Accepted 4 May 2015
Communicated by Haijun Jiang
Available online 14 May 2015

Keywords:
Projection neural network
Quadratic programming
Positive semidefinite
Time delay
Globally exponentially stable

1. Introduction

Constrained quadratic programming problems have been extensively studied in the past decades and widely applied in scientific and engineering areas, such as signal processing, robot control, image fusion, filter design, pattern recognition, regression analysis [1–4]. In practical applications, these optimization problems have a time-varying characteristic, so it is essential to solve the optimum solution in real time. However, most of conventional algorithms based on general-purpose digital computers may not be very efficient since the computing time required for a solution broadly relies on the dimension and structure of these optimization problems. One promising approach for handling real-time optimization is to employ artificial neural networks based on circuit implementation [5–8]. As a result of the inherent massive parallelism, the neural network approach can solve optimization problems in running time much faster than those of the most traditional optimization algorithms executed on digital computers [9].

The introduction of artificial neural networks that can be utilized to a closed-loop circuit was first proposed by Tank and Hopfield [10] in 1986. Thereafter, the neural networks for solving different kinds of quadratic programming problems have been studied extensively and significant research results have been achieved [11–16]. For example, Kennedy and Chua [11] presented a neural network which contains finite penalty parameters and generates approximate solution for solving nonlinear programming problems. Zhang and Constantinides [12] proposed the Lagrangian network with two-layer structure and can be used for strictly convex programming problems. In [13], a dual neural network was proposed for convex quadratic programming subject to linear equality and inequality constraints. Based on projection methods, several projection neural networks [17,18] were used for solving linear and quadratic programming problems, their approaches which deal with inequality constraints indirectly convert the inequality constraints into equality constraints by adding slack or surplus variables. By utilizing the Lagrangian coefficients, Effati and Ranjbar [14] proposed a new neural network model with a simple form and a less number calculation operation.

In 2014, Yang [15] and Nazemi [16] developed a new neural networks for solving the following quadratic programming problems with equality and inequality constraints

\[
\begin{align*}
\text{minimize} & \quad \frac{1}{2}x^TQx + c^Tx \\
\text{subject to} & \quad Ax = b, \\
& \quad Bx \leq d.
\end{align*}
\]

(1)

where \(x \in \mathbb{R}^n\), \(Q \in \mathbb{R}^{n \times n}\) is a symmetric and positive semi-definite matrix (\(0 < m < n\)), \(c \in \mathbb{R}^n\), \(A \in \mathbb{R}^{m \times n}\), \(\text{rank}(A) = m\) (\(0 < m < n\)), \(B \in \mathbb{R}^{p \times n}\), \(b \in \mathbb{R}^p\), \(d \in \mathbb{R}^p\).

It is noted that the above existing neural networks made a critical assumption that neurons communicate and respond instantaneously without any time delay in the practical circuit.
implementation. In reality, as is well known, time delay inevitably occurs in the signal transmission of hardware implementation between the neurons which will actually affect the dynamical behavior and may lead to the oscillation phenomenon or instability of networks [19,20]. In these years, several recurrent neural networks with time delays have been proposed for solving optimization problems. For example, Liu et al. [21] solved a class of linear projection equations by drawing the transmission delay into projection neural network [6] for the first time. And when the delayed neural network is applied to this optimization problem, the delay-independent stability criteria can be obtained. In consideration of the discrete delays, Yang and Cao [22] presented another delayed neural networks for solving convex optimization problem where the time delay occurred in the nonlinear projection transformation component. Niu and Liu [23] extended to consider a projection neural network with two different time delays for solving quadratic programming problems subject to linear constraints. Recently, Huang et al. [24] developed a projection neural network with discrete delays and distributed delays to solve linear variational inequality. By the theory of functional differential equation, the global exponential stability of the delayed neural network is obtained. Based on the proposed linear matrix inequality method, the monotonicity assumption on the linear variational inequality is no longer necessary. However, to the best of the author’s knowledge, only few author has considered time delay for solving quadratic programming problems [1].

Motivated by the aforementioned discussion, in this paper, we investigate the quadratic programming problems (1) which can be solved by drawing the transmission delay in the nonlinear projection [1]. The structure of this paper is arranged as follows. In Section 2, a new projection delayed neural network is proposed for solving the quadratic programming problems (1) with lower structure rather than existing neural network [9,16,17]. In a real-world application, by drawing into delay τ, the proposed neural network for solving the quadratic programming problems (1) is more meaningful than the neural network [15,16] and can be implemented by a circuit with a one-layer structure in Fig. 1.

(iii) According to differential inequality technique [25,26], which is different from previous traditional ways [19,27], the proposed projection neural network is proved to be a global exponential convergence.

The structure of this paper is arranged as follows. In Section 2, a delayed projection neural network with time delays is presented to solve a quadratic programming problem. In Section 3, we demonstrate the existence and uniqueness of the continuous solution. In Section 4, the globally exponential stability of the proposed neural network is investigated. In Section 5, some numerical simulations are given to demonstrate the effectiveness and performance of the proposed neural network. Finally, Section 6 concludes this paper.

2. A projection neural network

In this section, we are concerned with a quadratic programming problem of the form (1) and a corresponding projection delayed neural network with fewer state variables and the lower structure is presented.

For convenience of later discussion, it is necessary to introduce a few notations. Throughout this paper, $\mathbb{R}^n$ denotes the space of $n$-dimensional real column vectors, $[\mu]^T = ([\mu_1]^T, ..., [\mu_p]^T)^T \in \mathbb{R}^p$, $[\mu_1]^T = \max(0, \mu_1)$, and $[A]_{p \times p}$ represents the first $p$ rows of the matrix $A$. In what follows, we suppose that feasible domain $\Omega = \{x \in \mathbb{R}^n | Ax = b, Bx \leq d\}$ is not empty, which assures there exists a unique optimal solution $x^*$ to the problem (1).

According to problem (1), we define the Lagrange function as follows:

$$L(x, \lambda, \mu) = \frac{1}{2}x^TQx + c^T x - \lambda^T (Ax - b) - \mu^T (d - Bx),$$

where $\lambda \in \mathbb{R}^m$ and $\mu \in \mathbb{R}^p$ are referred to as the Lagrange multipliers.

To simplify the network architecture, the following Theorems based on the well-known Karush–Kuhn–Tucker (KKT) [28] condition of the convex optimization problems are given.

**Theorem 2.1.** $x^*$ is an optimal solution of (1) if and only if there exists $\mu^*$ such that $(x^*, \mu^*)$ satisfies the following equation set:

$$(I - U)(Qx + c + B^T \mu) + V(Ax - b) = 0,$$  

where $U = A^T(AA^T)^{-1}A$, $V = A^T(AA^T)^{-1}A$, $\alpha > 0$ and $\delta$ is a constant.

**Proof.** According to the KKT condition for convex optimization [28], we can see if $x^*$ is an optimal solution of (1) if and only if there exist $\lambda \in \mathbb{R}^m$ and $\mu \in \mathbb{R}^p$ such that $(x^*, \lambda^*, \mu^*)$ satisfies the following equation set:

$$Qx + c - A^T \lambda + B^T \mu = 0,$$  

$$Ax = b,$$  

$$d - Bx \geq 0, \quad \mu \geq 0, \quad \mu^T (d - Bx) = 0.$$  

Based on the proof of paper [15], the solutions of (4) and (5) are equivalent to the solutions of (2).

Next, by the projection theorem [3], it can be obtained that $d - Bx \geq 0, \mu \geq 0, \mu^T (d - Bx) = 0$ if and only if $[(\mu + \alpha(\mu^T d - d))] - \mu = 0$.

That is, equation set of (4)–(6) are reduced to equation set of (2) and (7).

Making use of the properties of elementary line transformation, we reformulate equation set of (2) and (7) as an equivalent equation set of (2) and (3). This completes the proof.

**Theorem 2.2.** $x^*$ is an optimal solution of (1) if and only if there exists $\mu^*$ such that $y^* = (x^*, \mu^*)^T$ satisfies projection equation

$$y = P(y - \alpha(My + q)).$$  

where $\alpha > 0$ is a constant,

$$y = \begin{pmatrix} x_{n+1} \\ \mu_{p+1} \\ \end{pmatrix}, \quad q = \begin{pmatrix} (I_n - U_{n \times n}^C)V_{n \times m}b \\ d + \delta(I_n - U_{n \times n}^C)V_{n \times m}b_p \\ \end{pmatrix},$$  

$$M = \begin{pmatrix} (I_n - U_{n \times n}^C)Q_{n \times n} + U_{n \times n} & (I_n - U_{n \times n})B_{n \times p} \\ \delta(I_n - U_{n \times n}^C)Q_{n \times n} + U_{n \times n} - B_{p \times p} & \delta(I_n - U_{n \times n}^C)B_{n \times p} \\ \end{pmatrix}.$$  

**Proof.** We denote $X_t = \{x \in \mathbb{R}^n | -\infty \leq x \leq +\infty\}$. Using the projection method, (2) can be reformulated as the following linear projection equation:

$$x = P_X(x - \alpha((I - U)(Qx + c + B^T \mu) + V(Ax - b))).$$

That is

$$x = P_X\left(\begin{pmatrix} (I_n - U_{n \times n})^C \\ 0_{n \times p} \end{pmatrix} \begin{pmatrix} x \\ \mu \end{pmatrix} - \alpha \begin{pmatrix} (I_n - U_{n \times n})^CQ_{n \times n} + U_{n \times n} & (I_n - U_{n \times n})B_{n \times p} \\ \delta(I_n - U_{n \times n})^CQ_{n \times n} + U_{n \times n} - B_{p \times p} & \delta(I_n - U_{n \times n})B_{n \times p} \end{pmatrix} \begin{pmatrix} x \\ \mu \end{pmatrix} + (I_n - U_{n \times n}^C)V_{n \times m}b \right).$$  

(9)
We also denote \( X_2 = \{ x \in \mathbb{R}^p \mid 0 \leq x \leq +\infty \} \). Using the projection method, (3) can be reformulated as the following linear projection equation:

\[
\mu = P_{X_1} [\mu + \alpha ((8x - d) - \delta (l - U)(Qx + c + B^T \mu) + V(Ax - b)]_p),
\]

where \( P_{X_1}(u_i) = [u_i]^+ \).

That is

\[
\mu = P_{X_1} \left\{ \left( 0_{p \times n} \ I_p \right) \left( x \right) \right\} - \alpha \left\{ \left( \delta(l_n - U_{n \times n})Q_{n \times n} + U_{n \times n} \right)_p - B_{p \times n} \ \delta(l_n - U_{n \times n}) c - V_{n \times m} b \right\}.
\]

Denote \( X = \{ x \in \mathbb{R}^{n+p} \mid l \leq x \leq h \} \), where

\[
l = \begin{pmatrix}
-\infty_{n \times 1} \\
0_{p \times 1}
\end{pmatrix}, \quad h = \begin{pmatrix}
+\infty_{n \times 1} \\
+\infty_{p \times 1}
\end{pmatrix}.
\]

Equations (9) and (10) can be rewritten as follows:

\[
\left( \frac{d}{dt} x \right) = P_X \left( \left( x \right) - \alpha \left( \left( \delta(l_n - U_{n \times n})Q_{n \times n} + U_{n \times n} \right)_p - B_{p \times n} \ \delta(l_n - U_{n \times n}) c - V_{n \times m} b \right) \right).
\]

Defined by

\[
M = \begin{pmatrix}
(l_n - U_{n \times n})Q_{n \times n} + U_{n \times n} & (l_n - U_{n \times n})B_{p \times n}^T \\
\delta(l_n - U_{n \times n})Q_{n \times n} + U_{n \times n}_p - B_{p \times n} & \delta(l_n - U_{n \times n})B_{p \times n}^T
\end{pmatrix},
\]

\[
y = \left( \begin{array}{c}
x \\ \mu
\end{array} \right)
\]

and

\[
q = \left( \begin{array}{c}
(l_n - U_{n \times n}) c - V_{n \times m} b \\ d + \delta(l_n - U_{n \times n}) c - V_{n \times m} b
\end{array} \right),
\]

the above projection equation can be reformulated as

\[
y = \mathcal{P}_X (y - \alpha (My + q)).
\]

This completes the proof. □

From Theorem 2.2, we propose a new delayed projection neural network for solving (1) as follows:

\[
\begin{cases}
\frac{dy}{dt} = -\beta y(t) + (\beta - 1) P_X (y(t - \tau) - \alpha (My(t - \tau) + q)) \\
+ P_X (y(t) - \alpha (My(t) + q)),
\end{cases}
\]

\[
y(t) = \varphi(t), \quad t \in [-\tau, 0].
\]

Fig. 1. Architecture of the delayed projection network (11).
where $P_{X} : R^{n+p} \rightarrow X$ is a projection operator defined by $P_{X}(y) = \arg \min_{y \in X} \|y - v\|$, $\forall y \in R^{n+p}$, $\beta > 0$ is a scale parameter, $\tau > 0$ denotes the transmission delay. It is easy to see that the architecture of the delayed projection neural network in Fig. 1 can be implemented by a circuit with a one-layer structure, where the vector $y$ is the network output, $aq = (\bar{a}_{1}, \ldots, \bar{a}_{n+p})^{T}$ is the network input vector, $I - \alpha M = (\bar{a}_{i})_{i \in R^{n+p}}$ is weight connection and $\phi = (\phi_{1}, \ldots, \phi_{n+p})^{T}$ is network initial state.

For further discussing existence and stability of (11), we introduce the following definitions and lemmas:

**Definition 2.1.** The point $y^{*}$ is said to be an equilibrium point of delayed projection neural network (11), if $y^{*}$ satisfies $y^{*} = P_{X}(y^{*} - \alpha(My^{*} + q))$.

**Definition 2.2.** The equilibrium point $y^{*}$ of the delayed projection neural network defined by (11) is said to be globally exponentially stable if there exist constants $K > 0$ and $\lambda > 0$, such that the output trajectory of this network satisfies

$$\|y(t) - y^{*}\| \leq K\|y^{*}\|e^{-\lambda t} \quad \text{for all } t \geq 0,$$

where $\|y^{*}\| = \sup_{-\tau \leq s \leq 0} \|y - y^{*}\|$.

**Lemma 2.1** (Kinderlehrer and Stampacchia [29]). Assume the set $\Omega \subset R^{n}$ is a closed convex set, then for any $x, y \in R^{n}$, $P_{\Omega}$ satisfies the following inequalities: $P_{\Omega}(x) - P_{\Omega}(y) \leq \|x - y\|$ and $(P_{\Omega}(x) - P_{\Omega}(y))'P_{\Omega}(y) - P_{\Omega}(y)'(x - y)$.

**Lemma 2.2** (Kinderlehrer and Stampacchia [29]). Assume that $Q$ is a positive-semidefinite matrix, then the set of equilibrium point of neural network (11) is nonempty.

**Remark 2.1.** It is easy to see that the equilibrium point of the projection delayed neural network (11) which is equal to the solution of quadratic programming problems (1). Therefore, from Theorems 2.1 and 2.2 we can derive that when the delayed projection neural network converges to its equilibrium point, the state trajectory $y(t)$ converges to the optimal solution of quadratic programming problems (1). That is, if the equilibrium point of proposed neural network is exponentially stable, then the steady output of neural network will be the solution of quadratic programming problems (1).

**Remark 2.2.** Let us compare our delayed projection neural network with some previous networks [9,17,16]. Based on the Karash–Kuhn–Tucker conditions, Effati and Nazemi [9] proposed a neural network for solving this problem in need of the positive definiteness of $Q$. According to saddle point theorem, Xue [17] proposed a project neural network which ease the conditions that $Q$ is a positive semi-definite matrix. By modifying the multipliers associated with inequality constraints, Nazemi [16] presented a neural network model to solve the convex quadratic programming problem requiring $Q$ to be a positive matrix. These above neural networks need $n+p+m$ neurons. Comparatively, our projection network not only has a simple structure with $n+p$ neurons but also state trajectory that can globally converge to the equilibrium point under weaker condition, i.e., $Q$ is a positive semi-definite matrix.

### 3. Existence and uniqueness

In this section, we will present the existence and uniqueness of the continuous solution to the projection delayed neural network (11).

Throughout this paper, we always suppose that

$$H_{1} : \frac{1 + |\beta - 1|}{\beta} ||I - \alpha M|| < 1,$$

where $\alpha$ and $\beta$ are defined by Theorem 2.1.

For convenience, denote $Y(t) = y(t) - y^{*}$, $F(Y(t)) = P_{X}(y(t) - \alpha(My(t) + q)) - P_{X}(y^{*} - \alpha(My^{*} + q))$, $\phi(t) = \phi(t) - y^{*}$, then delayed projection neural network (11) can be rewritten as

$$\begin{align*}
\frac{dY}{dt} &= -\beta Y(t) + F(Y(t)) + (\beta - 1)\phi(t), \\
\phi(t) &= \phi(t), \quad t \in [-\tau, 0].
\end{align*}$$

(12)

where $F(Y) = (f_{1}, \ldots, f_{n+p})^{T}$ and $Y(t) = (Y_{1}, \ldots, Y_{n+p})^{T}$.

Clearly, the solution to the delayed projection neural network (11) is existent and unique if and only if the solution to the delayed projection neural network (12) is existent and unique.

**Theorem 3.1.** Assume that the condition $H_{1}$ holds, then for any given $\phi \in C([-\tau, 0], R^{n+p})$, there exists a large enough number $D > 0$, such that

$$\|Y(t)\| < D.$$  

(13)

**Proof.** By the condition $H_{1}$, we have $1 - \frac{|1 + |\beta - 1|}{\beta} ||I - \alpha M|| > 0$. For any given $\phi \in C([-\tau, 0], R^{n+p})$, there exists a large enough number $D > 0$, such that

$$\|Y(t)\| < D \quad \text{for } t \geq -\tau.$$  

(14)

By the variation-of-constants formula, projection delayed neural network (12) can be written as

$$Y(t) = e^{-\beta t} Y(0) + \int_{0}^{t} e^{-\beta(t-s)} F(Y(s)) ds + (\beta - 1) \int_{0}^{t} e^{-\beta(t-s)} \phi(Y(s - \tau)) ds.$$  

(17)

It follows from (17) that

$$\|Y(t_{1})\| \leq e^{-\beta t_{1}} \|\phi\| + \frac{1}{\beta} \int_{0}^{t_{1}} \|I - \alpha M\| \|Y(s)\| e^{-\beta(t_{1} - s)} ds + (\beta - 1) \int_{0}^{t_{1}} e^{-\beta(t_{1} - s)} \|Y(s - \tau)\| e^{-\beta(t_{1} - s)} ds.$$  

$$\leq e^{-\beta t_{1}D} \frac{1 + ||I - \alpha M||}{\beta} \frac{1}{\beta} \int_{0}^{t_{1}} ||I - \alpha M|| \|Y(s - \tau)\| e^{-\beta(t_{1} - s)} ds.$$  

$$= e^{-\beta t_{1}D} \frac{1 + ||I - \alpha M||}{\beta} \int_{0}^{t_{1}} ||I - \alpha M|| D e^{-\beta(t_{1} - s)} ds.$$  

$$= e^{-\beta t_{1}D} \frac{1 + ||I - \alpha M||}{\beta} \int_{0}^{t_{1}} ||I - \alpha M|| D e^{-\beta(t_{1} - s)} ds.$$  

$$< D,$$

which contradicts the equality (15), and so inequality (14) holds. That is, the solution is uniformly bounded.

**Theorem 3.2.** If the condition $H_{1}$ holds, then for any given $\phi \in C([-\tau, 0], R^{n+p})$, there exists a unique continuous solution $Y(t)$ for the proposed neural network (12) in the global time interval $t \in [0, +\infty)$.

**Proof.** By the theory of functional differential equation and Theorem 3.1, the existence of the continuous solution to projection delayed neural network (12) can be inferred. Now we need to show that the solution $Y(t)$ is unique.

Suppose that $Y(t)$ and $\tilde{Y}(t)$ are two arbitrary solutions of (12), then by the variation-of-constants formula, it can be obtained that

$$Y(t) = e^{-\beta t} Y(0) + \int_{0}^{t} e^{-\beta(t-s)} F(Y(s)) ds + (\beta - 1) \int_{0}^{t} e^{-\beta(t-s)} \phi(Y(s - \tau)) ds.$$  

(18)
and
\[
Y(t) = e^{-\beta t}Y(0) + \int_0^t e^{-\beta s - \lambda}F(Y(s)) ds + (\beta - 1) \int_0^t e^{-\beta s - \lambda}F(Y(s) - r) ds. \quad (19)
\]

It follows from (18) and (19) that
\[
\begin{align*}
\sup_t \|Y(t) - \bar{Y}(t)\| & \leq \sup_t \left( \int_0^t \|F(Y(s)) - F(\bar{Y}(s))\| e^{-\beta s - \lambda} ds + \langle \beta - 1 \rangle \int_0^t e^{-\beta s - \lambda}F(\bar{Y}(s - r)) ds \right) \\
& \leq \sup_t \left( \int_0^t \|I - \alpha M\| \|Y(s) - \bar{Y}(s)\| e^{-\beta s - \lambda} ds + \langle \beta - 1 \rangle \int_0^t e^{-\beta s - \lambda}F(\bar{Y}(s - r)) ds \right) \\
& \leq \|I - \alpha M\| \sup_t \|Y(t) - \bar{Y}(t)\| \sup_t \left( \int_0^t e^{-\beta s - \lambda} ds \right) \\
& \leq \|I - \alpha M\| \sup_t \|Y(t) - \bar{Y}(t)\| \frac{1 + \|\beta - 1\|}{\beta} \sup_t \left( \int_0^t e^{-\beta s - \lambda} ds \right),
\end{align*}
\]
which implies that
\[
\left( 1 - \frac{1 + \|\beta - 1\|}{\beta} \right) \|I - \alpha M\| \sup_t \|1 - e^{-\beta r}\| \left( \sup_t \|Y(t) - \bar{Y}(t)\| \right) \leq 0.
\]

It is easy to observe that
\[
1 - \frac{1 + \|\beta - 1\|}{\beta} \|I - \alpha M\| \sup_t \|1 - e^{-\beta r}\| > 0.
\]

According to above two inequalities, we can obtain
\[
\sup_t \|Y(t) - \bar{Y}(t)\| \leq 0.
\]

Hence, \(Y(t) = \bar{Y}(t)\) which indicates there exists a unique continuous solution \(Y(t)\) for neutral network (12). The proof is completed.\(\blacksquare\)

4. Global exponential stability

In this section, we will prove the global exponential stability of the delayed network (11) under \(H_1\) condition.

Obviously, the equilibrium point \(y^*\) of the delayed projection neural network (11) is globally exponentially stable if and only if zero equilibrium point of delayed neural network (12) is globally exponentially stable.

**Theorem 4.1.** If the condition \(H_1\) holds, then the zero equilibrium point of the delayed projection neural network defined by (12) is globally exponentially stable.

**Proof.** According to the condition \(H_1\), we can choose some constant \(\lambda\) with \(0 < \lambda < \beta\), such that
\[
\frac{1 + \|\beta - 1\|}{\beta - \lambda} e^{-\beta |r|} < 1. \quad (20)
\]

For any \(\phi \in C((-\tau, 0], R^{n+p})\), we shall prove that
\[
\sup_t \|Y(t)\| \leq K\|\phi\| e^{-\beta t} \quad \text{for } t \geq -\tau,
\]
where \(K \geq \frac{1}{(1 + \|\beta - 1\|/\beta)e^{\beta |r|} |I - \alpha M|}.\)

To prove above, we first show, for any \(\gamma > 1\), the following inequality holds
\[
\|Y(t)\| < \gamma K\|\phi\| e^{-\beta t} \quad \text{for } t \geq -\tau. \quad (22)
\]

If (22) does not hold, then there must be some \(t_2 > 0\), such that
\[
\|Y(t_2)\| = \gamma K\|\phi\| e^{-\beta t_2},
\]
and
\[
\|Y(t)\| < \gamma K\|\phi\| e^{-\beta t_2} \quad \text{for } -\tau \leq t < t_2. \quad (24)
\]

By neural network system (12) and the variation-of-constants formula, we have
\[
\begin{align*}
\|Y(t_2)\| & \leq e^{-\beta t_2}\|\phi\| + \int_0^{t_2} e^{-\beta s - \lambda} \|I - \alpha M\| Y(s - \gamma) ds \\
& \leq e^{-\beta t_2}\|\phi\| + \int_0^{t_2} e^{-\beta s - \lambda} (1 + \|\beta - 1\| e^{\tau}) \|I - \alpha M\| Y(s - \gamma)\|\phi\| e^{-\beta s} ds \\
& = e^{-\beta t_2}\|\phi\| + \int_0^{t_2} e^{-\beta s - \lambda} (1 + \|\beta - 1\| e^{\tau}) \|I - \alpha M\| Y(s - \gamma)\|\phi\| e^{-\beta s} ds \\
& = e^{-\beta t_2}\|\phi\| e^{-\beta t_2} (1 + \|\beta - 1\| e^{\tau}) \|I - \alpha M\| Y(0)\|\phi\| e^{-\beta t_2} ds \\
& = e^{-\beta t_2}\|\phi\| e^{-\beta t_2} \left( \frac{1}{\beta - \lambda} \right) \left( 1 + \|\beta - 1\| e^{\tau} \|I - \alpha M\| \right),
\end{align*}
\]
which contradicts the equality of (23), so (22) holds. Letting \(\gamma \rightarrow 1\), then (21) holds. Combing (21) with definition of the exponential stability, we derive that the zero equilibrium point of projection neural network system (12) is globally exponentially stable and the proof is completed.\(\blacksquare\)

**Remark 4.1.** If condition \(H_1\) holds, then the output trajectory of our proposed projection neural network globally exponentially converges to a unique optimal solution of quadratic programming problems (1) within a finite time. Moreover, it has the following convergence rate \(\lambda\) satisfying
\[
\|y(t) - y^*\| \leq K \|q(t) - y^*\| e^{-\beta t} \quad \text{for } t \geq -\tau,
\]
where \(y(t) = (x(t))^T \mu(t)^T, 0 < \lambda < \beta, \frac{1 + \|\beta - 1\|}{\beta - \lambda} e^{-\beta |r|} |I - \alpha M| < 1\) and \(1 \leq K < \frac{1 + \|\beta - 1\|}{\beta - \lambda} e^{-\beta |r|} |I - \alpha M|\).

**Remark 4.2.** In proof of Theorems 3.1 and 4.1 of our paper, we present our approach of differential inequality technique which is different from previous traditional ways. So far as is known to the author, almost none of the paper is applied this way to solve quadratic programming problems.

5. Numerical simulation

In this section, we will show three illustrative computer-simulation examples so as to demonstrate the effectiveness of
the projection delayed neural network. The simulations are conducted by the Matlab R2012b.

**Example 1.** Consider the following quadratic program with equality constraints:

$$\min \ f(x) = 0.5x_1^2 + x_2^2 + 2x_3^3 + 1.25x_1x_2 + 1.25x_1x_3$$

subject to

$$x_1 + x_2 + 2x_3 = 7,$$

$$x_1 + x_2 - x_3 = 3.$$  

Let

$$Q = \begin{pmatrix} 1 & 1.25 & 1.25 \\ 1.25 & 2 & 1.25 \\ 1.25 & 1.25 & 4 \end{pmatrix}, \quad c = \begin{pmatrix} -11 \\ 0 \\ -5 \end{pmatrix},$$

$$A = \begin{pmatrix} 1 & 1 & 2 \\ -1 & 1 & -1 \end{pmatrix}, \quad b = \begin{pmatrix} 7 \\ 3 \end{pmatrix}.$$  

Then this quadratic programming problem with equality constraints can be rewritten as the quadratic programming problem with the form (1). This quadratic programming problem has an optimal solution $x^* = (2.6724, 5.2241, -0.4483)^T$. For the above problem, we use the delayed neural network (11) of degradation with the initial function on the interval $[-\tau, 0]$ as follows:

$$\frac{dx}{dt} = -\beta x(t) + (\beta - 1)P x(t - \alpha(Mx(t - \tau) + q)) + P x(t) - \alpha(Mx(t) + q),$$

where $M = (I_n - U_{m,n})Q_{n,n} + U_{m,n} \cdot q = (I_n - U_{m,n})c - V_{n,m}b$, $[\varphi(t)]_n = (\varphi_1, \dots, \varphi_n)^T$ and $\alpha > 0$ is a constant.

(i) **Globally exponentially converge to the optimal solution:** Consider the delayed projection neural network (25) with the following parameters $\lambda = 0.1$, $\alpha = 0.4$ and $\beta = 2$, then $1 + |\beta - 1| / \beta = 0.8768$ is $< 1$. Based on Theorem 4.1 and (20), when $0 \leq \tau < \frac{\ln(1 + |\beta - 1| / \beta)}{\ln 4} = 0.8076$, the equilibrium point is globally exponentially stable. Fig. 2 depicts the transient behavior of our delayed projection neural network (25) with 10 random initial functions when $\tau = 0.5$. From this figure we see that all the state trajectories globally exponentially converge to the optimal solution.

(ii) **Impact of delay on the convergence rate of neural network:** We consider the delayed projection neural network (25) with the following parameters $\alpha = 0.5$, $\beta = 2$ unchanged, and $\tau = 0.5, 2, 4, 6$, respectively. From Fig. 3 it is obvious that with $\tau$ increasing the delayed projection neural network (25) takes more time to converge to the optimal solution. That is to say, the transmission delay of the circuit neuron directly influences the global exponential convergence rate of the neural network (25).

**Example 2.** Consider the following quadratic program with equality and inequality constraints:

minimize $f(x) = 0.4x_1^2 + 0.3x_2^2 - 0.2x_1 - 0.4x_2 + 0.7x_3$

subject to

$x_1 - x_2 + x_3 = 5,$

$0.9x_1 + 0.2x_2 - 0.2x_3 \leq 4,$

$0.2x_1 + 0.7x_2 - 0.1x_3 \leq 10.$

Let

$$Q = \begin{pmatrix} 0.8 & -0.1 & 0 \\ -0.1 & 0.6 & 0 \\ 0 & 0 & 0 \end{pmatrix}, \quad c = \begin{pmatrix} -0.2 \\ 0.4 \\ 0.7 \end{pmatrix}, \quad A = (1 - 1 1),$$

$$b = 5, \quad B = \begin{pmatrix} 0.9 & 0.2 & -0.2 \\ 0.2 & 0.7 & -0.1 \end{pmatrix}, \quad d = \begin{pmatrix} 4 \\ 10 \end{pmatrix}.$$  

Then this quadratic programming problem with equality and inequality constraints can be rewritten as the quadratic programming problem with the form (1). It is obvious that $Q$ is a positive semi-definite matrix with the minimal eigenvalue 0 and the maximal eigenvalue 0.8414. This quadratic program has an optimal solution $x^* = (1.0851, -0.3191, 3.5957)^T$.

(i) **Globally exponentially converge to the optimal solution:** We choose $\lambda = 0.05$, $\alpha = 0.5$, $\delta = 1$ and $\beta = 2$, by calculating

$$U = \begin{pmatrix} 0.3333 & -0.3333 & 0.3333 \\ -0.3333 & 0.3333 & -0.3333 \\ -0.3333 & -0.3333 & 0.3333 \end{pmatrix}, \quad V = \begin{pmatrix} 0.3333 \\ -0.3333 \\ -0.3333 \end{pmatrix},$$

$$M = \begin{pmatrix} 0.8333 & -0.2000 & 0.3333 & 0.7333 & 0.4000 \\ -0.1333 & 0.7000 & -0.3333 & 0.3667 & 0.5000 \\ -0.0333 & -0.1000 & 0.3333 & -0.3667 & 0.1000 \\ -0.0667 & -0.4000 & 0.5333 & 0.7333 & 0.4000 \\ -0.3333 & 0 & -0.2333 & 0.3667 & 0.5000 \end{pmatrix}.$$  

$$1 + |\beta - 1| / \beta \cdot ||l - aM|| = 0.9651 < 1.$$  

Thus, by Theorem 4.1 and (20), when $0 \leq \tau < \frac{\ln(1 + |\beta - 1| / \beta)}{\ln 4} = 6.0119$, the equilibrium point of the delayed projection neural network (11) is globally exponentially stable. Fig. 4 illustrates state trajectories of the proposed delayed projection neural network (11) starting from 10 random initial functions when $\tau = 0.5$. It is shown that all the state trajectories globally exponentially converge to $[1.0851, -0.3191, 3.5957]^T$, which corresponds to the optimal solution.

(ii) **Impact of delay on the convergence rate of neural network:** Fix the parameters $\alpha = 0.5$, $\beta = 2, \delta = 1$ and choose $\tau = 1, 4, 7$, respectively. As shown in Fig. 5, it is obvious that with $\tau$ increasing the delayed projection neural network (11) takes more time to converge to the optimal solution. That is, the transmission delay of the circuit neuron directly influences the globally exponential convergence rate of the delayed projection neural network (11).
The advantage of multiple adjustable parameters: A criteria of possessing multiple adjustable parameters is prime of importance for designing a neural network. For example, in [15], a neural network for solving (1) was proposed as follows:

\[
\begin{align*}
\frac{dx}{dt} &= -(I - P)(W x + c + B^T (y + B x - d)^+) - Q(A x - b), \\
\frac{dy}{dt} &= -2y + \frac{1}{2}(y + B x - d)^+.
\end{align*}
\]

where \( P = A^T (A A^T)^{-1} A \), \( Q = A^T (A A^T)^{-1} \).

Compared to the above neural network, our delayed projection neural network has introduced multiple parameters such as \( \delta \) and \( \beta \). On one hand, take \( \alpha = 1 \) and \( \delta = 0 \), then Theorem 2.1 becomes Lemma 2 proposed by Yang [15]. From Theorem 2.2, matrix

\[
M = \begin{pmatrix} (I - U)Q + U & (I - U)B^T \\ B & 0 \end{pmatrix}
\]

could be derived. It is noted that \( M \) corresponding to our projection network is not positive definite, by Remark 2 of paper [30], we obtain \( \| I - \alpha M \| \geq 1 \), for any \( \alpha > 0 \). Thus condition \( H_1 \) fails. In order

Fig. 3. The transient behaviors of the delayed projection neural network vary with time delay increasing.

Fig. 4. Ten group of transient behaviors of the delayed projection neural network when \( \tau = 0.5 \) and \( \alpha = 0.5 \).

Fig. 5. The transient behaviors of the delayed projection neural network vary with time delay increasing.

Fig. 6. The transient behaviors vary with different neural networks.
to obtain a sufficient condition ensuring global exponential stability, by drawing $\alpha = 0.5$ and $\delta = 1$ into our neural network, we improve the approach to modify the matrix $M$ so that our condition $H_1$ holds. On the other hand, choose $\alpha$ and $\delta$ unchanged, fix $\tau = 1.8$, as shown from Fig. 6, it illustrates that the output transient behaviors of our neural network with appropriate $\beta = 1.4$ is faster than neural network (26).

**Example 3.** Consider the following quadratic program with equality and inequality constraints:

\[
\text{minimize } f(x) = 0.7x_1^2 + 0.6x_2^2 + 0.5x_3^2 + 0.35x_1x_2 + 0.45x_1x_3 + 0.25x_2x_3 + x_1 - 0.7x_2 + 0.9x_3 \\
\text{subject to } x_1 + 0.5x_2 - x_3 = 5, \\
x_1 + 0.2x_2 - 0.3x_3 \leq 6, \\
-0.2x_1 + x_2 + 0.1x_3 \leq 3.
\]

Let

\[
Q = \begin{pmatrix} 1.4 & 0.35 & 0.45 \\
0.35 & 1.2 & 0.25 \\
0.45 & 0.25 & 1 \end{pmatrix}, \quad c = \begin{pmatrix} 1 \\
-0.7 \end{pmatrix}.
\]

(A = (1 0.5 -1), $b = 5$)

\[
B = \begin{pmatrix} 1 & 0.2 & -0.3 \\
-0.2 & 1 & 0.1 \end{pmatrix}, \quad d = \begin{pmatrix} 6 \\
3 \end{pmatrix}.
\]

Then this quadratic programming problem with equality and inequality constraints can be rewritten as the quadratic programming problem with the form (1). It is obvious that $Q$ is a positive definite matrix with the minimal eigenvalue 0.7064. This quadratic program has an optimal solution $x^* = (0.9846, 1.5855, -3.2226)^T$.

(i) **Globally exponentially converge to the optimal solution:** Choose the parameters as follows $\lambda = 0.01$, $\alpha = 0.75$, $\delta = 1$ and $\beta = 2$, by calculating

\[
U = \begin{pmatrix} 0.4444 & 0.2222 & -0.4444 \\
0.2222 & 0.1111 & -0.2222 \end{pmatrix}, \quad V = \begin{pmatrix} 0.4444 \\
-0.4444 & -0.2222 & 0.4444 \end{pmatrix}, \quad M = \begin{pmatrix} 1.3444 & 0.2611 & 0.1944 & 0.3778 & -0.2889 \\
0.3222 & 1.1556 & 0.1222 & -0.1111 & 0.9556 \\
0.5056 & 0.3389 & 1.2556 & 0.3222 & 0.1889 \\
0.3444 & 0.0611 & 0.4944 & 0.3778 & -0.2889 \\
0.5222 & 0.1556 & 0.0222 & -0.1111 & 0.9556 \end{pmatrix},
\]

\[
1 + \frac{|\beta - 1|}{\beta} ||I - \alpha M|| = 0.9390 < 1.
\]
It follows from Theorem 4.1 and (20) that the equilibrium point of the delayed projection neural network (11) is globally exponentially stable when $0 \leq \tau < \frac{\ln \left(1 - \frac{1}{e} \right)}{\beta_1 - \beta_2} = 11.2682$. Further, Fig. 7 shows that all state trajectories $x(t)$ starting from ten different initial functions globally exponentially converge to the unique optimal solution when $\tau = 1$.5.

(ii) Impact of delay on the convergence rate of neural network: To observe the impact of delay on the convergence of our neural network, we consider $\alpha = 0.75$, $\beta = 2$, $\delta = 1$ unchanged, $\tau = 0$ and 1, respectively. As seen from Fig. 8, it illustrates the output transient behavior of our neural network with delay will converge faster than the corresponding neural network without delay which implies that the delayed neural network is more effective with the introduction of proper delay $\tau$. Further, Choose $\delta = 1$, $\tau = 1$, respectively, and other parameters unchanged, we can observe from Fig. 9 that the delayed projection neural network (11) needs more time to converge to the optimal solution with $\delta = 5$ increasing. That is to say, the transmission delay of the circuit neuron directly influences the global exponential convergence rate of the neural network (11).

6. Conclusion

In this paper, we propose a new recurrent neural network modeled by project differential equation for resolving quadratic programming problems with general linear constraints containing mixed constraints. Comparing with the existing neural network for quadratic optimization problems, the proposed projection neural network is designed with fewer neurons, meanwhile simpler architecture and consequently complexity and difficulty of computation can be reduced. In addition, utilizing the differential inequality technique which is different from previously known methods, we further propose the global exponential stability of the presented neural network under a sufficient condition. Finally, several simulation results clearly demonstrate the convergence behavior and the superior performance of the proposed neural network.

References


Chunlin Sha received the B.S. and M.S. degrees in Nanjing University of Aeronautics and Astronautics, Nanjing, China, in 2001 and 2006, respectively. Since 2011, she has been studying the Doctoral Degree in Mathematics at Nanjing University of Aeronautics and Astronautics. He has been a lecturer at the College of Science, Nanjing University of Aeronautics and Astronautics, Nanjing, China, since 2009. His research interests include stability theory, nonlinear systems, neural networks, optimization theory and applications.

Hongyang Zhao received the Ph.D. degree from Sichuan University, Chengdu, China, and the Post-Doctoral Fellow in the Department of Mathematics at Nanjing University, Nanjing, China. He was with Department of Mathematics at Nanjing University of Aeronautics and Astronautics, Nanjing, China. He is currently a Professor of Nanjing University of Aeronautics and Astronautics, Nanjing, China. He is also the author or coauthor of more than 60 journal papers. His research interests include nonlinear dynamic systems, neural networks, control theory, and applied mathematics.

Fengli Ren received the B.S. degree in Mathematics from Zhengzhou University, Zhengzhou, China, in 2001, and the M.S. and Ph.D. degrees in Mathematics from Southeast University, Nanjing, China, in 2006 and 2009 respectively. She has been an Associate Professor at the College of Science, Nanjing University of Aeronautics and Astronautics, Nanjing, China, since 2012. Her research interests include stability theory, nonlinear systems, neural networks, chaos synchronization and gene regulatory networks.