

A One-Layer Recurrent Neural Network for Pseudoconvex Optimization Subject to Linear Equality Constraints

Zhishan Guo, *Student Member, IEEE*, Qingshan Liu, *Member, IEEE*, and Jun Wang, *Fellow, IEEE*

Abstract—In this paper, a one-layer recurrent neural network is presented for solving pseudoconvex optimization problems subject to linear equality constraints. The global convergence of the neural network can be guaranteed even though the objective function is pseudoconvex. The finite-time state convergence to the feasible region defined by the equality constraints is also proved. In addition, global exponential convergence is proved when the objective function is strongly pseudoconvex on the feasible region. Simulation results on illustrative examples and application on chemical process data reconciliation are provided to demonstrate the effectiveness and characteristics of the neural network.

Index Terms—Global convergence, linear equality constraints, pseudoconvex optimization, recurrent neural networks.

I. INTRODUCTION

CONSIDER the following constrained nonlinear optimization problem:

$$\begin{aligned} &\text{minimize} && f(x) \\ &\text{s.t.} && Ax = b \end{aligned} \quad (1)$$

where $x \in \mathbb{R}^n$ is the vector of decision variables, $A \in \mathbb{R}^{m \times n}$ is of a coefficient matrix with full row-rank (i.e., $\text{rank}(A) = m \leq n$), and the objective function $f(x) : \mathbb{R}^n \rightarrow \mathbb{R}$ is differentiable, bounded below, locally Lipschitz continuous [1], [2] and pseudoconvex on the feasible region $\{x | Ax - b = 0\}$. In this paper, we assume that problem (1) has at least one finite solution. Constrained optimization with pseudoconvex objective functions has widespread applications, such as fractional programming [3], [4], frictionless contact analysis [5], applications in economics [6], and computer vision [7].

Since Tank and Hopfield's pioneering work on a neural network approach to linear programming [8], the design, and applications of recurrent neural networks for optimization

have been widely investigated. For example, the Lagrangian network for solving nonlinear programming problems with equality constraints [9], the deterministic annealing network for convex programming [10], the Lagrangian network for solving nonlinear programming problems with equality constraints [9], [11], the projection-type neural network for convex programming [12], and a generalized neural network for non-smooth nonlinear programming problems [13] were developed. Recently, several recurrent neural networks with discontinuous activation functions were proposed for solving optimization problems [10], [14]–[23]. Specifically, a nonfeasible gradient projection recurrent neural network was proposed [20] and thoroughly analyzed for convex optimization problems and extended to nonconvex optimization with nonlinear constraints [24]. In particular, a one-layer recurrent neural network for non-smooth convex optimization subject to linear equality constraints was presented [18]. In a recent work, the neural network was applied for constrained sparsity maximization in compressive sensing [25].

In addition to convex optimization, it was shown [26] that pseudomonotone variational inequality and pseudoconvex optimization problems with bound constraints can be solved by using the projection neural network [27], [28]. In this paper, a one-layer recurrent neural network is presented for solving pseudoconvex optimization problems subject to linear equality constraints. The scope of neurodynamic optimization can be expanded from convex optimization problems to pseudoconvex ones.

The remainder of this paper is organized as follows. The related preliminaries and model descriptions are presented in Section II. In Section III, we discuss the stability of the one-layer recurrent neural network. The global convergence, global asymptotic stability, and global exponential stability of the neurodynamic system are delineated under different conditions. Two numerical examples are presented in Section IV. In Section V, an application for chemical process data reconciliation is discussed based on pseudoconvex performance criterion and the present recurrent neural network. Finally, Section VI concludes this paper.

II. PRELIMINARIES

In [18], a one-layer recurrent neural network was proposed for non-smooth convex optimization

$$\epsilon \frac{dx}{dt} \in -Px - (I - P)\partial f(x) + q, x_0 = x(t_0) \quad (2)$$

Manuscript received July 17, 2010; accepted September 18, 2011. Date of publication October 31, 2011; date of current version December 1, 2011. This work was supported in part by the Research Grants Council of the Hong Kong Special Administrative Region in China, under Grant CUHK417209E and Grant CUHK417608E.

Z. Guo is with the Department of Computer Science, University of North Carolina at Chapel Hill, Chapel Hill, NC 27599 USA (e-mail: zsguo@cs.unc.edu).

Q. Liu is with the School of Automation, Southeast University, Nanjing 210096, China (e-mail: qslu@seu.edu.cn).

J. Wang is with the Department of Mechanical and Automation Engineering, Chinese University of Hong Kong, New Territories, Hong Kong (e-mail: jwang@mae.cuhk.edu.hk).

Color versions of one or more of the figures in this paper are available online at <http://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/TNN.2011.2169682

where x is the state vector, ϵ is a positive scaling constant, I is an identity matrix, $P = A^T(AA^T)^{-1}A$, $q = A^T(AA^T)^{-1}b$, and $\partial f(x)$ is the sub-differential of $f(x)$.

In particular, when f in (1) is differentiable, $\nabla f(x)$ is used instead of $\partial f(x)$ as the gradient of $f(x)$

$$\epsilon \frac{dx}{dt} = -Px - (I - P)\nabla f(x) + q, \quad x_0 = x(t_0).$$

However, its global convergence results are established for convex optimization problems only, and theoretically its state reaches the feasible region only when time approaches to infinity. To achieve the finite-time convergence to the feasible region and global convergence to optimal solutions for pseudoconvex optimization problems, the model is modified to

$$\epsilon \frac{dx}{dt} = -(I - P)\nabla f(x) - A^T g(Ax - b), \quad x_0 = x(t_0) \quad (3)$$

where $g = (g(x_1), g(x_2), \dots, g(x_m))^T$ and its component is defined as

$$g(x_i) = \begin{cases} 1, & \text{if } x_i > 0, \\ 0, & \text{if } x_i = 0, \\ -1, & \text{if } x_i < 0. \end{cases} \quad (i = 1, 2, \dots, m) \quad (4)$$

For the convenience of later discussions, several definitions and theorems on pseudoconvex optimization are introduced below.

Definition 1: A differentiable function $f: \mathbb{R}^n \rightarrow \mathbb{R}$ is said to be pseudoconvex on a set Ω if $\forall x, y \in \Omega, x \neq y$

$$\nabla f(x)^T(y - x) \geq 0 \Rightarrow f(y) \geq f(x).$$

The function f is said to be strictly pseudoconvex on Ω if $\forall x \neq y \in \Omega$

$$\nabla f(x)^T(y - x) \geq 0 \Rightarrow f(y) > f(x)$$

and strongly pseudoconvex on Ω if there exist a constant $\beta > 0$ such that $\forall x \neq y \in \Omega$

$$\nabla f(x)^T(y - x) \geq 0 \Rightarrow f(y) > f(x) + \beta \|x - y\|_2^2$$

where $\|\cdot\|_2$ is the L_2 -norm, which will be written as $\|\cdot\|$ hereafter.

Definition 2: A function $F: \mathbb{R}^n \rightarrow \mathbb{R}^n$ is said to be pseudomonotone on a set Ω if $\forall x, x' \in \Omega, x \neq x'$

$$F(x)^T(x' - x) \geq 0 \Rightarrow F(x')^T(x' - x) \geq 0. \quad (5)$$

A very important result on pseudoconvex optimization is given by the following lemma, and its proof follows directly from Theorem 4.3.8 in the reference.

Lemma 1 [29]: For (1), if the Karush–Kuhn–Tucker (KKT) conditions hold at a feasible solution \bar{x} , i.e., $\exists y \in \mathbb{R}^m, \nabla f(\bar{x}) - A^T y = 0$, then \bar{x} is a global optimal solution to (1).

III. GLOBAL CONVERGENCE

In this section, we analyze the global convergence of the recurrent neural network (3). The dynamical system is described by an ordinary differential equation with a discontinuous right-hand side, and Filippov solution is considered

in this paper. First of all, the definition of global convergence is given. Afterward, Theorem 1 discusses the finite-time convergence of the states to the feasible region of (1). In Theorems 2 and 3, the Lyapunov stability of the proposed neural network is proved, based on which the globally convergence to the optimal solution of (1) is then shown. Theorem 4 reveals the exponential convergence of the neural network when the gradient of the optimal function $\nabla f(x)$ is strongly pseudomonotone.

The state vector of the neural network (3) is said to be globally convergent to an optimal solution of (1) if for any $x(t)$ of the neural network with initial point $x_0 \in \mathbb{R}^n$, such that $\lim_{t \rightarrow +\infty} x(t) = x^*$, where x^* is an optimal solution. The existence of the solution can be derived from the locally Lipschitz continuity of the objective function $f(\cdot)$ and Proposition 3 in [30]. The solution for discontinuous system may not be unique [31], and the LaSalle invariant set theorem does not require the uniqueness of the solution.

Denote the feasible region as $\mathcal{S} = \{x | Ax = b\}$.

Theorem 1: The state vector of the recurrent neural network (3) is globally convergent to the feasible region \mathcal{S} in finite time by $t_S = \epsilon \|Ax_0 - b\|_1 / \lambda_{\min}(AA^T)$ and stays there thereafter, where x_0 is the initial value, and λ_{\min} is the minimum eigenvalue of the matrix.

Proof: Note that $\mathcal{B}(x) = \|Ax - b\|_1$, which is convex and regular, by using the chain rule [32], [13], we have

$$\frac{d}{dt} \mathcal{B}(x) = \zeta^T \frac{dx(t)}{dt} \quad \forall \zeta \in \partial \mathcal{B}(x(t)) = A^T K[g(Ax - b)]$$

where $K(\cdot)$ denotes the closure of the convex hull, i.e., the Filippov set-valued map [30], and $\dot{x}(t)$ is given by (3).

From the definition of P , we know that $A(I - P) = A - AA^T(AA^T)^{-1}A = 0$. Thus for any $x_0 \in \mathbb{R}^n$, when $x(t) \in \mathbb{R}^n \setminus \mathcal{S}$, we have

$$\exists \eta \in K[g(Ax - b)] \text{ such that } \frac{d}{dt} \mathcal{B}(x) = -\frac{1}{\epsilon} \|A^T \eta\|^2.$$

For any $x \in \mathbb{R}^n \setminus \mathcal{S}$, $Ax - b \neq 0$. So at least one of the components of η is 1 or -1 . On one hand, since A has full row-rank, AA^T is invertible. It follows that:

$$\|(AA^T)^{-1}AA^T\eta\| = \|\eta\| \geq 1.$$

Since AA^T is positive definite, we have

$$\|A^T \eta\|^2 = \eta^T AA^T \eta \geq \lambda_{\min}(AA^T) \|\eta\|^2 \geq \lambda_{\min}(AA^T).$$

Thus

$$\frac{d\mathcal{B}(x(t))}{dt} \leq -\frac{1}{\epsilon} \lambda_{\min}(AA^T) < 0. \quad (6)$$

Integrating the both sides of (6) from $t_0 = 0$ to t , we have

$$\|Ax(t) - b\|_1 \leq \|Ax_0 - b\|_1 - \frac{1}{\epsilon} \lambda_{\min}(AA^T) t.$$

Thus, $Ax(t) - b = 0$ as $t = \epsilon \|Ax_0 - b\|_1 / \lambda_{\min}(AA^T)$. That is, the state vector of neural network (3) reaches \mathcal{S} in finite time and an upper bound of the hit time is $t_S = \epsilon \|Ax_0 - b\|_1 / \lambda_{\min}(AA^T)$.

Next, we prove that, when $t \geq t_S$, the state vector of neural network (3) remains inside \mathcal{S} thereafter. If not so, assume that

the trajectory leaves \mathcal{S} at time t_1 and stays outside of \mathcal{S} for almost all $t \in (t_1, t_2)$, where $t_1 < t_2$. Then, $\|A(x(t_1)) - b\|_1 = 0$, and from above analysis, $\|A(x(t)) - b\|_1 < 0$ for almost all $t \in (t_1, t_2)$ which is a contradiction. That is, the state vector of neural network (3) reaches the equality feasible region \mathcal{S} by $t_{\mathcal{S}}$ at the latest and stays there thereafter. ■

Theorem 2: Let $f(x)$ be pseudoconvex on \mathcal{S} . The state vector of the neural network (3) is stable in the sense of Lyapunov and globally convergent to the equilibrium point set for any $x_0 \in \mathbb{R}^n$. In particular, assume that $f(x)$ is strictly pseudoconvex on \mathcal{S} , then the neural network (3) is globally asymptotically stable.

Proof: Denote \bar{x} as an equilibrium point of (3), i.e., $0 \in A^T K[g(A\bar{x} - b)] + (I - P)\nabla f(\bar{x})$. Since by Theorem 1, any trajectory $x(t)$ will convergent to the feasible region \mathcal{S} in finite time $t_{\mathcal{S}} = \epsilon \|A(x_0) - b\|_1 / \lambda_{\min}(AA^T)$, and will remain in \mathcal{S} forever, i.e., $\forall t \geq t_{\mathcal{S}}, x(t) \in \mathcal{S}$. As a result, it suffices to show the stability of the system with $x(t) \in \mathcal{S}$.

Consider the following Lyapunov function:

$$V_1(x) = f(x) - f(\bar{x}) + \frac{1}{2} \|x - \bar{x}\|^2. \quad (7)$$

Clearly, $\forall x \in \mathcal{S}$ and $x \neq \bar{x}$, $V_1(x) > 0$, $0 = \eta \in K[g(Ax - b)]$, and $Ax - b = 0$. So $A(\bar{x} - x) = 0$, and $(x - \bar{x})^T P = (x - \bar{x})^T A^T (AA^T)^{-1} A = [A(x - \bar{x})]^T (AA^T)^{-1} A = 0$, as well as $P^T(x - \bar{x}) = 0$. Since $x = P^T x + (I - P)^T x$ and $(I - P)\nabla f(\bar{x}) = 0$, as a result

$$\begin{aligned} \nabla f(\bar{x})^T (x - \bar{x}) &= \nabla f(\bar{x})^T [P^T(x - \bar{x}) + (I - P)^T(x - \bar{x})] \\ &= [(I - P)\nabla f(\bar{x})]^T (x - \bar{x}) = 0. \end{aligned} \quad (8)$$

By the pseudoconvexity of $f(x)$ on \mathcal{S} , we know that $\nabla f(x)$ is a pseudomonotone mapping on \mathcal{S} [33]. Thus from (8), we know that for any $x \in \mathcal{S}$ and $x \neq \bar{x}$, $\nabla f(x)^T (x - \bar{x}) \geq 0$

$$\begin{aligned} \frac{dV_1(x)}{dt} &= \nabla V_1(x)^T \cdot \frac{dx}{dt} \\ &= -(\nabla f(x) + x - \bar{x})^T ((I - P)\nabla f(x) + A^T \eta) \\ &= -\nabla f(x)^T (I - P)\nabla f(x) - (x - \bar{x})^T \nabla f(x) \\ &\quad + (x - \bar{x})^T P \nabla f(x) \\ &\leq -\|(I - P)\nabla f(x)\|^2 \leq 0. \end{aligned} \quad (9)$$

Furthermore, $d(V_1(x))/dt = 0$ if and only if $(I - P)\nabla f(x) = 0$, since $f(\cdot)$ is locally Lipschitz continuous, from LaSalle invariant set theorem [30], [34], [35] $x(t) \rightarrow \bar{\Omega} = \{x | d(V_1(x))/dt = 0\}$.

Now we show that $\{x | d(V_1(x))/dt = 0\}$ is the same set as $\{x | dx/dt = 0\}$. From (9), it is obvious that $d(V_1(x))/dt = 0 \Rightarrow (I - P)\nabla f(x) = 0$. Since the assumption that $x(t) \in \mathcal{S}$ has been made at the beginning of the proof, we have $0 \in K[g(Ax - b)]$. Thus $d(V_1(x))/dt = 0 \Rightarrow (I - P)\nabla f(x) = 0 \Rightarrow dx/dt = 0$. For any x that satisfies $dx/dt = 0$, it is clear that $d(V_1(x))/dt = d(V_1(x))/dx \cdot dx/dt = 0$. As a result, $x(t) \rightarrow \bar{\Omega} = \{x | d(V_1(x))/dt = 0\} = \{x | dx/dt = 0\}$, thus the neural network is stable in the sense of Lyapunov and globally convergent to the equilibrium points set.

If $f(x)$ is strictly pseudoconvex on \mathcal{S} , $\nabla f(x)$ is a strictly pseudomonotone mapping on \mathcal{S} [33]. From, we know that thus

from (8), then $\forall x \in \mathcal{S}$ and $x \neq \bar{x}$, $(x - \bar{x})^T \nabla f(x) > 0$. From (9), we know that $\forall x \in \mathcal{S}$ and $x \neq \bar{x}$, $dV_1(x)/dt < 0$, and $dV_1(x)/dt = 0$ if and only if $x = \bar{x}$. Also $f(x) > f(\bar{x})$ can be derived from $\nabla f(x)^T (x - \bar{x}) = 0$ for any $x \in \mathcal{S}$ since $f(x)$ is strictly pseudoconvex. As a result, \bar{x} is a unique equilibrium point. Thus if $f(x)$ is strictly pseudoconvex on \mathcal{S} , the neural network (3) is globally asymptotically stable. ■

Theorem 3: Let $f(x)$ be pseudoconvex on \mathcal{S} . The state vector of the neural network (3) is globally convergent to optimal solution set of (1) for any $x_0 \in \mathbb{R}^n$. In addition, when $f(x)$ is strictly pseudoconvex on \mathcal{S} , the neural network (3) is globally convergent to the unique optimal solution x^* of (1).

Proof: From Theorem 2, we know that the (3) is stable in the sense of Lyapunov, an globally convergent to equilibrium point set $\bar{\Omega} = \{x | dx/dt = 0\}$. As $0 \in -(I - P)\nabla f(\bar{x}) - A^T K[g(A\bar{x} - b)]$ holds for any \bar{x} and $P = A^T (AA^T)^{-1} A$, we have

$$0 \in \nabla f(\bar{x}) - A^T (AA^T)^{-1} A \nabla f(\bar{x}) + A^T K[g(A\bar{x} - b)].$$

Let $y \in (AA^T)^{-1} A \nabla f(\bar{x}) - K[g(A\bar{x} - b)]$. Then $\nabla f(\bar{x}) - A^T y = 0$, which means \bar{x} satisfies KKT condition of (1). Considering Lemma 1, we can conclude that any equilibrium point \bar{x} of (3) is an optimal solution x^* of (1). Thus the neural network (3) is globally convergent to the optimal solution set of (1).

For strictly pseudoconvex optimization, since the solution x^* is unique, it is obvious that the neural network (3) is convergent to the optimal solution of (1). ■

Theorem 4: Let $\nabla f(x)$ be strongly pseudomonotone on \mathcal{S} . For any initial point $x_0 \in \mathbb{R}^n$, the state vector of the neural network (3) is exponentially convergent to the optimal solution x^* of (1) after $t \geq t_{\mathcal{S}}$.

Proof: By the strongly pseudomonotone of $\nabla f(x)$ on \mathcal{S} , since in (8) we have $\nabla f(\bar{x})^T (x - \bar{x}) = 0$, $\exists \gamma > 0$, such that $\forall t > t_{\mathcal{S}}$

$$\nabla f(x)^T (x - \bar{x}) \geq \gamma \|x - \bar{x}\|^2$$

where \bar{x} is an equilibrium point that satisfies $0 \in -A^T K[g(A\bar{x} - b)] - (I - P)\nabla f(\bar{x})$.

Consider the following Lyapunov function:

$$V_2(x) = \frac{1}{2} \|x - \bar{x}\|^2. \quad (10)$$

We have

$$\frac{dV_2(x)}{dt} \leq -(x - \bar{x})^T \nabla f(x) \leq -\gamma \|x - \bar{x}\|^2 = -2\gamma V_2(x).$$

As a result, $\forall t > t_{\mathcal{S}}$

$$V_2(x(t)) \leq V_2(x(t_{\mathcal{S}})) \exp(-2\gamma(t - t_{\mathcal{S}})).$$

From Lemma 1 and the proof in Theorem 3, as $\nabla f(x)$ is strongly pseudomonotone on \mathcal{S} , $f(x)$ is pseudoconvex on \mathcal{S} . Thus we know that \bar{x} satisfies KKT condition and is the optimal solution x^* . Because $V_2(x) = 0$ if and only if $x = \bar{x}$, the neural network (3) is exponentially convergent to the optimal solution x^* of (1) after $t \geq t_{\mathcal{S}}$. ■

Note that any strictly convex quadratic function is also strictly pseudoconvex. Thus the state vector of the neural

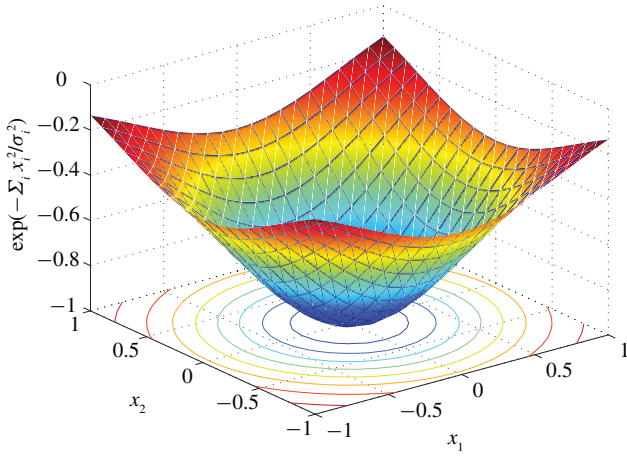


Fig. 1. Isometric of inverted 2-D non-normalized Gausssian function with $\sigma = [1, 1]^T$.

network (3) is also exponentially convergent to the optimal solution for strictly convex quadratic optimization subject to linear equality constraints.

IV. NUMERICAL EXAMPLES

To demonstrate the performance of the one-layer neural network in solving pseudoconvex optimization problems with linear equality constraints, two illustrative examples are given in this section. Many functions in nature are pseudoconvex, such as Butterworth filter functions, fractional functions, and some density functions in probability theory. Among them, the Gaussian function as shown in Fig. 1 is chosen in Example 1, and quadratic fractional function is chosen for Example 2. In the following simulations, the differential equation defined by (3) is solved using MATLAB r2008a ode45 algorithm on a 2.4 GHZ Intel CoreTM2 Qrad PC running Windows Vista with 2.0 GB main memory.

Example 1: Consider the following pseudoconvex optimization problem with linear equality constraints:

$$\begin{aligned} & \text{minimize} && -\exp\left(-\sum_{i=1}^2 \frac{x_i^2}{\sigma_i^2}\right) \\ & \text{s.t.} && Ax = b \end{aligned} \quad (11)$$

where $x \in \mathbb{R}^2$, $\sigma = (1, 1)^T$, the elements of $A = [0.787, 0.586]$ and $b = 0.823$ randomly drawn from the uniform distribution over $(0, 1)$. Obviously, the objective function is locally Lipschitz continuous and strictly pseudoconvex on \mathbb{R}^2 .

Since the conditions in Theorems 1–3 hold, the one-layer recurrent neural network (3) is globally asymptotically stable and capable of solving this optimization problem. Fig. 2 is the state phase plot of the neural network (3) from 20 random initial points converge to the feasible set $\mathcal{S} = \{x | Ax - b = 0\}$ in finite time, and then converge to x^* . It is also obvious that the state variables stay in the feasible region \mathcal{S} once into it. Fig. 3 shows the transient states of the neural network (3) with $\epsilon = 10^{-6}$ in Example 1, where 20 random initial points are generated from the uniform distribution over $(-1, 1)$.

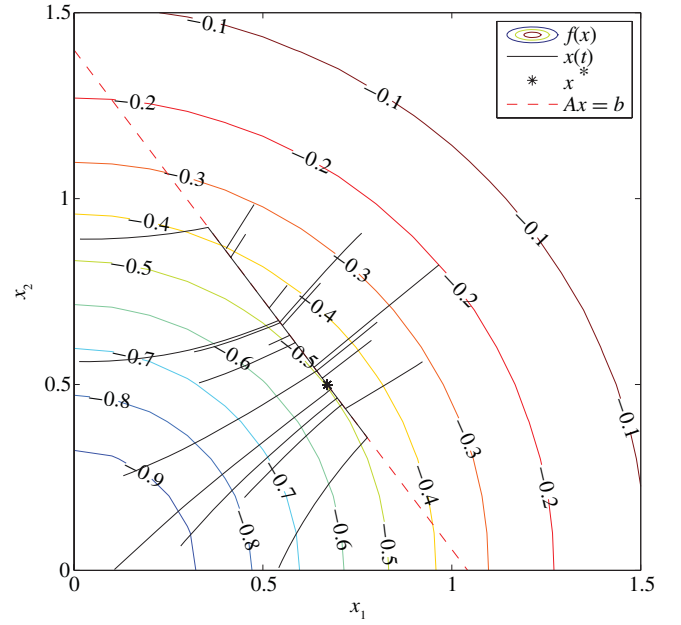


Fig. 2. Transient behaviors of the neural network (3) with 20 random initial points in Example 1.

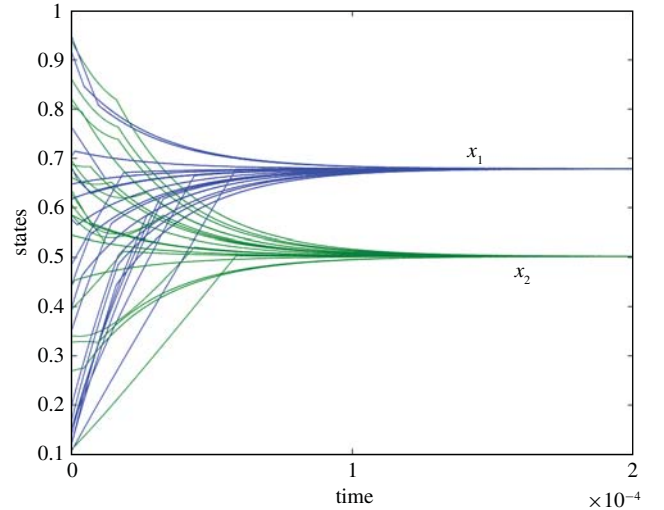


Fig. 3. Transient states of the neural network (3) in Example 1.

The projection neural network [27] and the two-layer recurrent neural network [21] are also used for solving the same problem (11). As there is no bound constraints, the projection neural network will degeneration to the Lagrangian network [9], [36] which is given by the following equations where x is the output state vector and y is the hidden state vector:

$$\begin{aligned} \frac{dx}{dt} &= -\nabla f(x) + A^T y \\ \frac{dy}{dt} &= -Ax + b. \end{aligned} \quad (12)$$

The global convergence of the Lagrangian network for convex optimization was studied in [11]. However, global convergence is not guaranteed for pseudoconvex problems.

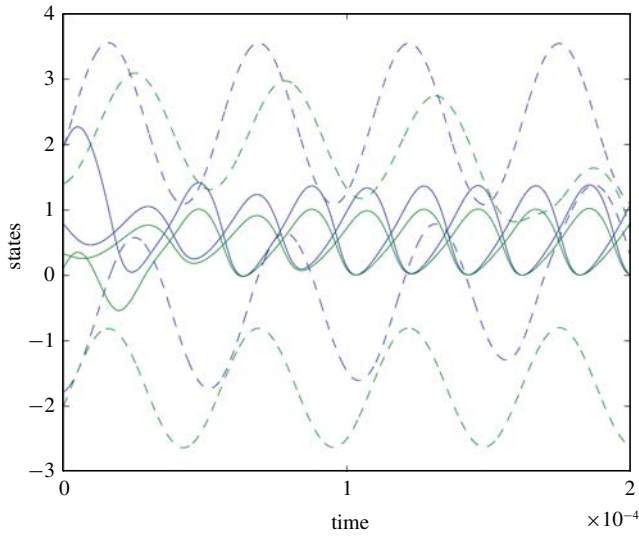


Fig. 4. Transient states of both the Lagrangian network (in dashed line) and the two-layer recurrent neural network (in continues line) in Example 1.

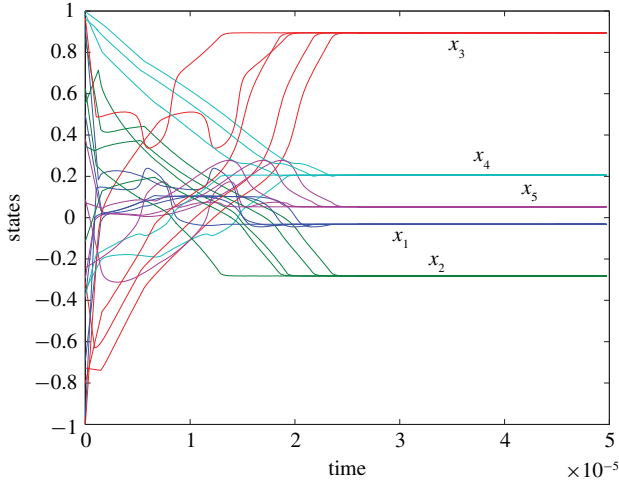


Fig. 5. Transient states of the one-layer recurrent neural network (3) from 5 random initial points ($n = 5$) in Example 1.

Fig. 4 shows the transient states of both the Lagrangian network (in dashed line) and the two-layer recurrent neural network (in continues line) in Example 1, where two random initial points are generated from the uniform distribution over $(-2, 2)$ for models. It is obvious that the state vectors of both the Lagrangian network and the two layer recurrent neural network oscillate and do not converge to x^* for this example.

Furthermore, consider (11) in a higher-dimension case with $n = 5$, where $\sigma = [1, 1/2, 1/4, 1/2, 1]^T$, $A \in \mathbb{R}^{3 \times 5}$ and $b \in \mathbb{R}^3$ are drawn from the uniform distribution over $(0, 1)^5$. Fig. 5 depicts the transient states of the one-layer recurrent neural network (3) with $\epsilon = 10^{-6}$, where five random initial points are generated from the uniform distribution over $(-1, 1)^5$. It also shows the global convergence of the states to the unique optimal solution of the problem.

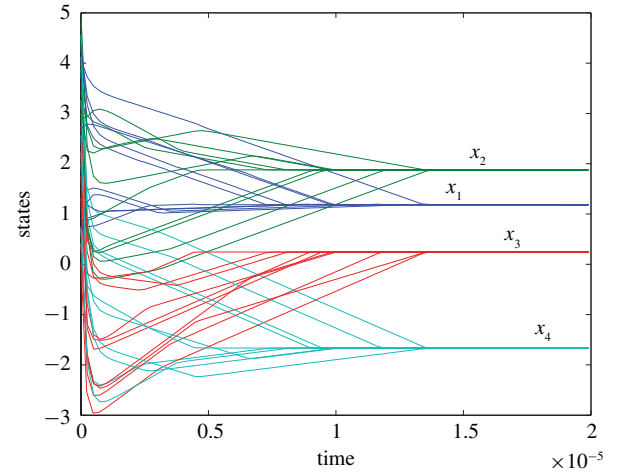


Fig. 6. Transient behaviors of the one-layer recurrent neural network (3) with ten random initial points in Example 2.

Example 2: One of the important classes of pseudoconvex optimization problems is the quadratic fractional programming problem

$$\begin{aligned} & \text{minimize} && \frac{x^T Q x + a^T x + a_0}{c^T x + c_0} \\ & \text{s.t.} && A x = b \end{aligned} \quad (13)$$

where Q is an $n \times n$ positive semidefinite matrix, $a, c \in \mathbb{R}^n$, and $a_0, c_0 \in \mathbb{R}$. It is known that the objective function is pseudoconvex on the half space $\{x | c^T x + c_0 > 0\}$.

Let $n = 4$

$$Q = \begin{pmatrix} 5 & -1 & 2 & 0 \\ -1 & 5 & -1 & 3 \\ 2 & -1 & 3 & 0 \\ 0 & 3 & 0 & 5 \end{pmatrix}, \quad a = \begin{pmatrix} 1 \\ -2 \\ -2 \\ 1 \end{pmatrix}, \quad c = \begin{pmatrix} 2 \\ 1 \\ -1 \\ 0 \end{pmatrix},$$

$$A = \begin{pmatrix} 2 & 1 & -1 & 0 \\ 1 & 0 & 2 & -2 \end{pmatrix}, \quad b = \begin{pmatrix} 4 \\ 5 \end{pmatrix}, \quad a_0 = -2, \quad c_0 = 5.$$

As Q is symmetric and positive definite in \mathbb{R}^4 , the objective function is pseudoconvex on the feasible region $\{x | A x = b\}$ [3]. Fig. 6 depicts the transient states of the one-layer recurrent neural network (3) with $\epsilon = 10^{-6}$, where ten random initial points are generated from the uniform distribution over $(0, 5)^4$. It shows the global convergence to the unique optimal solution of the problem.

Fig. 7 shows the transient states of both the Lagrangian network (in dashed line) and the two-layer recurrent neural network (in continues line) in Example 2, with a random initial point generated from the uniform distribution over $(0, 5)$.

V. DATA RECONCILIATION

Measured process data usually contain several types of errors. It is important to understand what is wrong with the values obtained by the measurement and how they can be adjusted [37]. Data reconciliation is a means to adjust process data measurements by minimizing the error and ensuring constraint satisfaction, which is a way to improve the quality

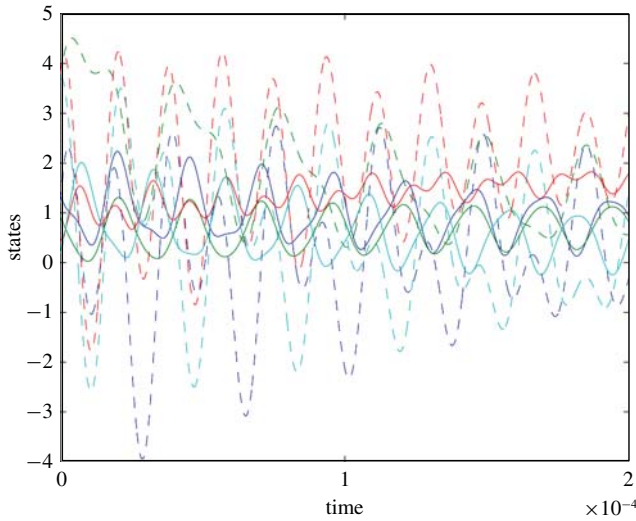


Fig. 7. Transient states of both the Lagrangian network (dashed line) and the two-layer recurrent neural network (continues line) in Example 2.

of distributed control systems. A good estimation is usually defined as the optimal solution to a constrained maximum likelihood objective function subject to data flow balance constraints. Real-time data reconciliation is necessary to make properly use of the large amount of available process information.

This section reports the results of the proposed neurodynamic optimization approach to data reconciliation. It is shown that the problem can be formulated as pseudoconvex optimization problem with linear equality constraints. Based on a performance index, simulation results on industrial applications in the literature are shown.

Consider a series of measured data with errors

$$y_i(k) = z_i(k) + e_i(k), \quad i = 1, \dots, n, \quad k = 1, 2, \dots$$

where $y_i(k)$ is the k th measured value of element i , $z_i(k)$ is the true value of the element, and $e_i(k)$ is the identically independent distribution error that usually consists of three different types of errors: small random Gaussian errors, Cauchy distributed systematic biases and drift, and gross errors which are usually large resulting from instrument malfunction. It is shown in literature that Cauchy (Lorentzian) function is the most effective generalized maximum likelihood objective function with higher data reconciliation performance [38]. Thus data reconciliation can be stated in the following form for each given k :

$$\begin{aligned} & \text{maximize} \quad \Pi_i \left\{ \frac{1}{\pi \sigma_i (1 + (y_i - x_i)^2 / \sigma_i^2)} \right\} \\ & \text{s.t.} \quad \sum_{j=1}^n a_{ij} x_j = b_i, \quad i = 1, \dots, n \end{aligned} \quad (14)$$

where y_i is the measurement of variable i , x_i is the reconciled estimate, σ_i is a scaler statistical parameter of the error.

Now we will show that the Cauchy function $g(x) = \Pi_i 1 / [\pi \sigma_i (1 + (y_i - x_i)^2 / \sigma_i^2)]$ is strictly pseudoconcave on \mathbb{R}^n . It is shown in [33] that a differentiable function is strictly pseudoconcave if and only if its negative gradient is a strictly pseudomonotone mapping where the definition is given below.

From the definition of Cauchy function, we have

$$\frac{\partial g}{\partial x_i} = g(x) \frac{2(x_i - y_i)}{\sigma_i^2 [1 + (y_i - x_i)^2 / \sigma_i^2]}.$$

Since $g(x) \geq 0$ holds always, $\forall x, x' \in \mathbb{R}^n$, from $-\nabla g(x)^T (x' - x) \geq 0$, we have

$$\sum_{i=1}^n \frac{2(x_i - y_i)(x'_i - x_i)}{[1 + (y_i - x_i)^2 / \sigma_i^2]} \geq 0$$

which simply leads to $\nabla g(x')^T (x' - x) > 0$. Thus $g(x)$ is strictly pseudoconvex on \mathbb{R}^n , and (14) is a pseudoconvex optimization problem with linear equality constraints. As a result, the proposed neurodynamic optimization method can be used for solving the data reconciliation problem in chemical processes. The benefit for using neural networks for data reconciliation is that the proposed neural dynamic system can achieve the optimal solution in very little time, which makes real-time data reconciliation possible.

Total error reductions (TER) [39] is often used to evaluate the data validation performance

$$TER = \max \left\{ 0, \frac{\sqrt{\sum_{i=1}^n ((y_i - z_i) / \sigma_i)^2} - \sqrt{\sum_{i=1}^n ((x_i^* - z_i) / \sigma_i)^2}}{\sqrt{\sum_{i=1}^n ((y_i - z_i) / \sigma_i)^2}} \right\}. \quad (15)$$

The range of TER is $[0, 1]$ and it reaches its maximum when the optimal solution x^* is exactly the same as the true value z .

In the following experiments, the measurement sets y_i are generated for each variable by adding noise from Cauchy and normal distributions with equal probability to true value z_i . For the gross errors, outliers are created in ten percent randomly selected measurements by adding or subtracting 10–100 percent of the true values. The lower bounds on the measurement variables are set to 50 percent of the true values and the upper bounds to twice of the true values.

Example 3: Consider a chemical reactor with two entering and two leaving mass flows [40]. The four variables are related by three linear mass balance equations, where

$$\begin{aligned} A &= \begin{pmatrix} 0.1 & 0.6 & -0.2 & -0.7 \\ 0.8 & 0.1 & -0.2 & -0.1 \\ 0.1 & 0.3 & -0.6 & -0.2 \end{pmatrix}, \quad b = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}, \\ \sigma &= \text{diag}(0.00289, 0.0025, 0.00576, 0.04), \\ z &= (0.1850, 4.7935, 1.2295, 3.880)^T. \end{aligned}$$

Figs. 8 and 9 show, respectively, the transient states of the neural network (3) and the performance index value TER with five random initial states and the same errors. It shows the global convergence of the neurodynamic optimization approach.

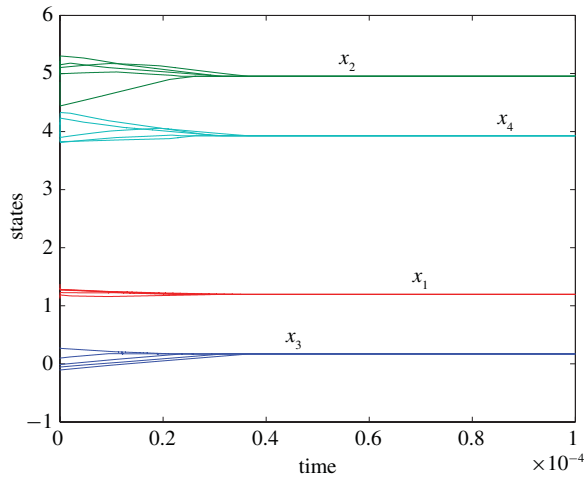


Fig. 8. Transient states of the neural network (3) for data reconciliation with five random initial states in Example 3.

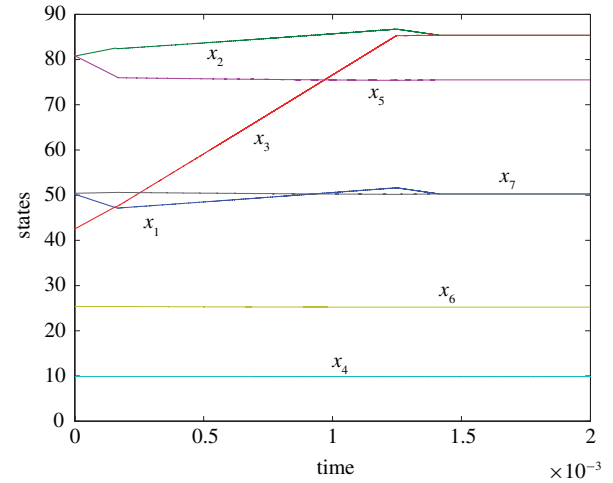


Fig. 10. Transient states of the neural network (3) for data reconciliation in Example 4.

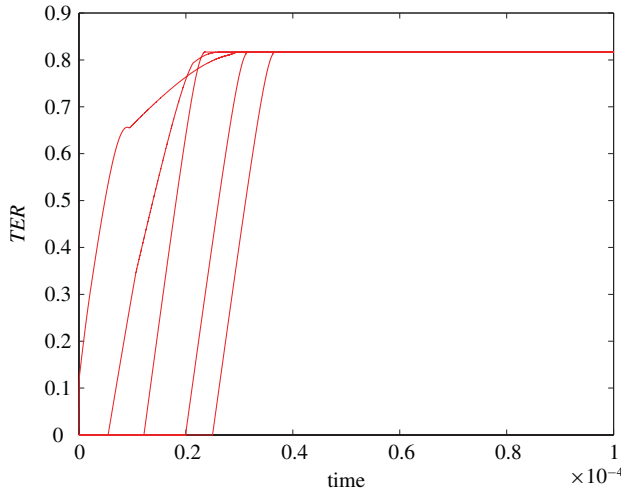


Fig. 9. Transient behaviors of the performance index TER in Example 3.

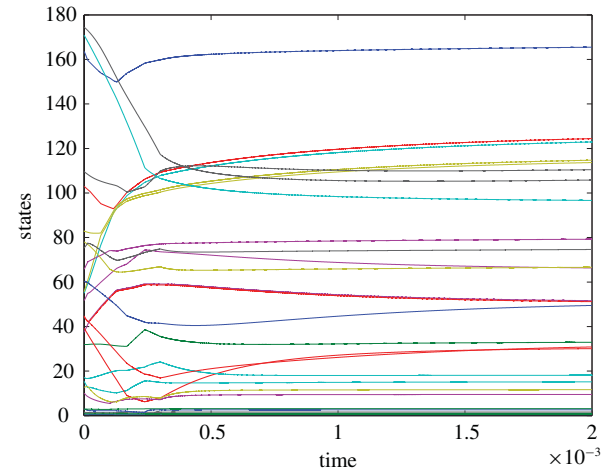


Fig. 11. Transient states of the neural network (3) for data reconciliation in Example 5.

Example 4: Consider a recycle process network, where seven streams are identified with overall material balance as four linear equality constraints [41], where

$$A = \begin{pmatrix} 1 & -1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & -1 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & -1 & -1 \end{pmatrix}, b = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

$$\sigma = \text{diag}(1.5625, 4.5156, 4.5156, 0.0625, 3.5156, 0.3906, 0.3906)$$

$$z = (49.5, 81.5, 85.3, 10.1, 72.9, 25.7, 50.7)^T.$$

Fig. 10 shows the transient states of the neural network.

Example 5: Consider a steam metering system with 28 measured variables and twelve linear equality constraints [42] (detailed information for this example is not listed here since space is limited). Fig. 11 shows transient states of the neural network.

Fig. 12 depicts the performance index TER during the convergent processes in Examples 4 and 5. It shows that there

TABLE I
PERFORMANCE OF MONTE CARLO TESTS IN TERMS OF TER IN
EXAMPLES 3–5

Example	Gaussian	Cauchy		
	average	average	max	min
3	0.751 [38]	0.757	0.992	0.424
4	0.764 [43]	0.789	0.898	0.260
5	0.466 [38]	0.526	0.558	0.205

are mainly two parts in this transient behaviors: at first, the state vector x converges to a feasible point (that satisfies linear constraints) in a very short time, during which the TER value may even decrease, and then converges to the optimal solution of the problem, where the TER value increases and reaches its maximum value.

Table I summarizes the results of Monte Carlo tests with random errors of 100 runs. The average, maximum (max), and minimum (min) values of TER with Cauchy errors and

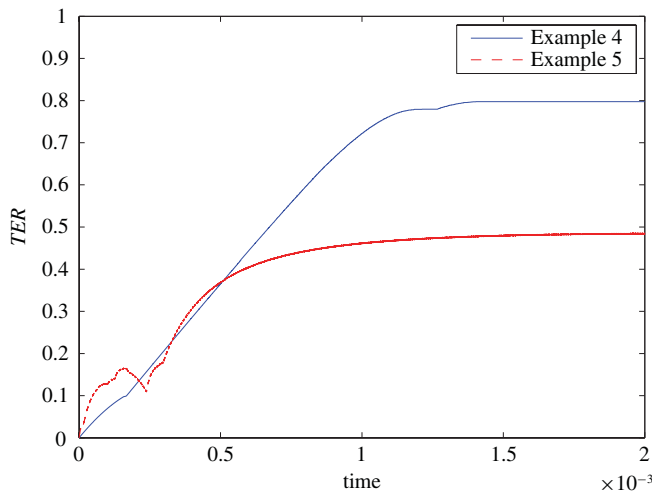


Fig. 12. Transient behaviors of the performance index TER in Examples 4 and 5.

also average TER of Gaussian ones are compared. Obviously, the results with Cauchy errors are better than those with Gaussian ones.

VI. CONCLUSION

In this paper, a single-layer recurrent neural network for solving pseudoconvex optimization problems with linear equality constraints was proposed based on an existing model for convex optimization. The reconstructed recurrent neural network was proven to be globally stable in the sense of Lyapunov, globally asymptotically stable, and global exponentially stable when the objective function is pseudoconvex, strictly pseudoconvex, and strongly pseudoconvex in the feasible region, respectively. Simulation results on numerical examples and an applications for chemical process data reconciliation were elaborated upon to substantiate the effectiveness and performance of the recurrent neural network.

REFERENCES

- [1] O. L. Mangasarian, "Pseudo-convex functions," *SIAM J. Control*, vol. 3, no. 2, pp. 281–290, 1965.
- [2] J. Ponstein, "Seven kinds of convexity," *SIAM Rev.*, vol. 9, no. 1, pp. 115–119, Jan. 1967.
- [3] W. Dinkelbach, "On nonlinear fractional programming," *Manage. Sci.*, vol. 13, no. 7, pp. 492–498, Mar. 1967.
- [4] L. Carosi and L. Martein, "Some classes of pseudoconvex fractional functions via the Charnes-Cooper transformation," in *Generalized Convexity and Related Topics*. Berlin, Germany: Springer-Verlag, 2006, pp. 177–188.
- [5] J. R. Barber, *Solid Mechanics and Its Applications*, 2nd ed. 2004, ch. 26, pp. 351–357.
- [6] F. Forgo and I. Joó, "Fixed point and equilibrium theorems in pseudoconvex and related spaces," *J. Global Optim.*, vol. 14, no. 1, pp. 27–54, Jan. 1999.
- [7] C. Olsson and F. Kahl, "Generalized convexity in multiple view geometry," *J. Math. Imaging Vis.*, vol. 38, no. 1, pp. 35–51, Sep. 2010.
- [8] D. W. Tank and J. J. Hopfield, "Simple 'neural' optimization networks: An A/D converter, signal decision circuit, and a linear programming circuit," *IEEE Trans. Circuits Syst.*, vol. 33, no. 5, pp. 533–541, May 1986.
- [9] S. Zhang and A. G. Constantinides, "Lagrange programming neural networks," *IEEE Trans. Circuits Syst. II*, vol. 39, no. 7, pp. 441–452, Jul. 1992.
- [10] J. Wang, "A deterministic annealing neural network for convex programming," *Neural Netw.*, vol. 7, no. 4, pp. 629–641, 1994.
- [11] J. Wang, Q. Hu, and D. Jiang, "A Lagrangian network for kinematic control of redundant robot manipulators," *IEEE Trans. Neural Netw.*, vol. 10, no. 5, pp. 1123–1132, Sep. 1999.
- [12] Y. Xia and J. Wang, "A recurrent neural network for solving nonlinear convex programs subject to linear constraints," *IEEE Trans. Neural Netw.*, vol. 16, no. 2, pp. 379–386, Mar. 2005.
- [13] M. Forti, P. Nistri, and M. Quincampoix, "Generalized neural network for nonsmooth nonlinear programming problems," *IEEE Trans. Circuits Syst. I*, vol. 51, no. 9, pp. 1741–1754, Sep. 2004.
- [14] S. Liu and J. Wang, "A simplified dual neural network for quadratic programming with its KWT application," *IEEE Trans. Neural Netw.*, vol. 17, no. 6, pp. 1500–1510, Nov. 2006.
- [15] Q. Liu and J. Wang, "A recurrent neural network for non-smooth convex programming subject to linear equality and bound constraints," in *Proc. Int. Conf. Neural Inf. Process.*, vol. 4233, 2006, pp. 1004–1013.
- [16] Q. Liu and J. Wang, "A one-layer recurrent neural network with a discontinuous activation function for linear programming," *Neural Comput.*, vol. 20, no. 5, pp. 1366–1383, May 2008.
- [17] Q. Liu and J. Wang, "A one-layer recurrent neural network with a discontinuous hard-limiting activation function for quadratic programming," *IEEE Trans. Neural Netw.*, vol. 19, no. 4, pp. 558–570, Apr. 2008.
- [18] Q. Liu and J. Wang, "A one-layer recurrent neural network for non-smooth convex optimization subject to linear equality constraints," in *Proc. Int. Conf. Neural Inf. Process.*, vol. 5507, 2009, pp. 1003–1010.
- [19] Q. Liu and J. Wang, "Finite-time convergent recurrent neural network with a hard-limiting activation function for constrained optimization with piecewise-linear objective functions," *IEEE Trans. Neural Netw.*, vol. 22, no. 4, pp. 601–603, Apr. 2011.
- [20] M. P. Barbarosou and N. G. Maratos, "Non-feasible gradient projection recurrent neural network for equality constrained optimization," in *Proc. IEEE Int. Joint Conf. Neural Netw.*, vol. 3, Jul. 2004, pp. 2251–2256.
- [21] Y. Xia, G. Feng, and J. Wang, "A novel recurrent neural network for solving nonlinear optimization problems with inequality constraints," *IEEE Trans. Neural Netw.*, vol. 19, no. 8, pp. 1340–1353, Aug. 2008.
- [22] Y. Xia, G. Feng, and J. Wang, "A recurrent neural network with exponential convergence for solving convex quadratic program and related linear piecewise equations," *Neural Netw.*, vol. 17, no. 7, pp. 1003–1015, Sep. 2004.
- [23] X. Hu, C. Sun, and B. Zhang, "Design of recurrent neural networks for solving constrained least absolute deviation problems," *IEEE Trans. Neural Netw.*, vol. 21, no. 7, pp. 1073–1086, Jul. 2010.
- [24] M. P. Barbarosou and N. G. Maratos, "A nonfeasible gradient projection recurrent neural network for equality-constrained optimization problems," *IEEE Trans. Neural Netw.*, vol. 19, no. 10, pp. 1665–1677, Oct. 2008.
- [25] Z. Guo and J. Wang, "A neurodynamic optimization approach to constrained sparsity maximization based on alternative objective functions," in *Proc. Int. Joint Conf. Neural Netw.*, Barcelona, Spain, Jul. 2010, pp. 1–8.
- [26] X. Hu and J. Wang, "Solving pseudomonotone variational inequalities and pseudoconvex optimization problems using the projection neural network," *IEEE Trans. Neural Netw.*, vol. 17, no. 6, pp. 1487–1499, Nov. 2006.
- [27] Y. Xia, H. Leung, and J. Wang, "A projection neural network and its application to constrained optimization problems," *IEEE Trans. Circuits Syst. I*, vol. 49, no. 4, pp. 447–458, Apr. 2002.
- [28] X. Hu and J. Wang, "Design of general projection neural networks for solving monotone linear variational inequalities and linear and quadratic optimization problems," *IEEE Trans. Syst., Man, Cybern., Part B: Cybern.*, vol. 37, no. 5, pp. 1414–1421, Oct. 2007.
- [29] M. S. Bazaraa, H. D. Sherali, and C. M. Shetty, *Nonlinear Programming: Theory and Algorithms*, 3rd ed. New York: Wiley, 2006.
- [30] J. Cortes, "Discontinuous dynamical systems," *IEEE Control Syst. Mag.*, vol. 28, no. 3, pp. 36–73, Jun. 2008.
- [31] A. Bacciotti and F. Ceragioli, "Stability and stabilization of discontinuous systems and nonsmooth Lyapunov functions," *ESAIM: Control, Optim. Calculus Variat.*, vol. 4, pp. 361–376, Jun. 1999.

- [32] J. P. Aubin and H. Frankowska, *Set-Valued Analysis*. Basel, Switzerland: Birkhäuser, 1990.
- [33] S. Karamardian and S. Schaible, "Seven kinds of monotone maps," *J. Optim. Theory Appl.*, vol. 66, no. 1, pp. 37–46, Jul. 1990.
- [34] J. P. LaSalle, "Some extensions of Liapunov's second method," *IRE Trans. Circuit Theory*, vol. 7, no. 4, pp. 520–527, Dec. 1960.
- [35] H. K. Khalil, *Nonlinear Systems*. Englewood Cliffs, NJ: Prentice-Hall, 1996.
- [36] S. Zhang, X. Zhu, and L. H. Zou, "Second-order neural nets for constrained optimization," *IEEE Trans. Neural Netw.*, vol. 3, no. 6, pp. 1021–1024, Nov. 1992.
- [37] J. Romagnoli and M. Sanchez, *Data Processing and Reconciliation for Chemical Process Operation*. New York: Academic, 1999.
- [38] B. Derya and W. Ralph, "Theory and practice of simultaneous data reconciliation and gross error detection for chemical processes," *Comput. Chem. Eng.*, vol. 28, no. 3, pp. 381–402, 2004.
- [39] R. Serth, C. Valero, and W. Heenan, "Detection of gross errors in nonlinearly constrained data: A case study," *Chem. Eng. Commun.*, vol. 51, nos. 1–6, pp. 89–104, 1987.
- [40] D. Ripps, "Adjustment of experimental data," *Chem. Eng. Progr. Symp. Ser.*, vol. 61, no. 55, pp. 8–13, 1965.
- [41] D. Rollins and J. Davis, "Gross error detection when variance-covariance matrices are unknown," *J. Amer. Inst. Chem. Eng.*, vol. 39, no. 8, pp. 1335–1341, Aug. 1993.
- [42] R. Serth and W. Heenan, "Gross error detection and data reconciliation in steam-metering systems," *J. Amer. Inst. Chem. Eng.*, vol. 32, no. 5, pp. 733–742, 1986.
- [43] T. Edgar, D. Himmelblau, and L. Lasdon, *Optimization of Chemical Processes*, 2nd ed. New York: McGraw-Hill, 2002.
- [44] L. Cheng, Z. Hou, M. Tan, Y. Lin, W. Zhang, and F. Wang, "Recurrent neural network for non-smooth convex optimization problems with application to the identification of genetic regulatory networks," *IEEE Trans. Neural Netw.*, vol. 22, no. 5, pp. 714–726, May 2011.
- [45] N. Hadjisavvas and S. Schaible, "On strong pseudomonotonicity and (semi)strict quasimonotonicity," *J. Optim. Theory Appl.*, vol. 79, no. 1, pp. 139–155, 1993.
- [46] P. Huber, *Robust Statistics*. New York: Wiley, 1981.



Zhishan Guo (S'10) received the B.E. degree in computer science and technology from Tsinghua University, Beijing, China, and the M.Phil. degree in mechanical and automation engineering from the Chinese University of Hong Kong, Shatin, Hong Kong, in 2009 and 2011, respectively. He is currently pursuing the Ph.D. degree with the Department of Computer Science, University of North Carolina at Chapel Hill, Chapel Hill.

His current research interests include computational intelligence, machine learning, and bio-

informatics.



Qingshan Liu (S'07–M'08) received the B.S. degree in mathematics from Anhui Normal University, Wuhu, China, the M.S. degree in applied mathematics from Southeast University, Nanjing, China, and the Ph.D. degree in automation and computer-aided engineering from the Chinese University of Hong Kong, Shatin, Hong Kong, in 2001, 2005, and 2008, respectively.

He joined the School of Automation, Southeast University, Nanjing, in 2008. From August 2009 to November 2009, he was a Senior Research Associate with the Department of Manufacturing Engineering and Engineering Management, City University of Hong Kong, Kowloon, Hong Kong. From February 2010 to August 2010, he was a Post-Doctoral Fellow with the Department of Mechanical and Automation Engineering, Chinese University of Hong Kong. From June 2011 to September 2011, he was a Post-Doctoral Fellow with the Department of Systems Engineering and Engineering Management, City University of Hong Kong. He is currently an Associate Professor with the School of Automation, Southeast University. His current research interests include optimization theory and applications, artificial neural networks, computational intelligence, and nonlinear systems.



Jun Wang (S'89–M'90–SM'93–F'07) received the B.S. degree in electrical engineering and the M.S. degree in systems engineering from the Dalian University of Technology, Dalian, China, in 1982 and 1985, respectively, and the Ph.D. degree in systems engineering from Case Western Reserve University, Cleveland, OH, in 1991.

He is a Professor with the Department of Mechanical and Automation Engineering, Chinese University of Hong Kong, Shatin, Hong Kong. He held various academic positions at the Dalian University of Technology, Case Western Reserve University, and the University of North Dakota, Grand Forks. He also held various short-term visiting positions at U.S. Air Force Armstrong Laboratory, Dayton, OH, in 1995, RIKEN Brain Science Institute, Wako Saitama, Japan, in 2001, Universite Catholique de Louvain, Ottignies, Belgium, in 2001, Chinese Academy of Sciences, Beijing, China, in 2002, Huazhong University of Science and Technology, Hubei, China, from 2006 to 2007, Shanghai Jiao Tong University, Shanghai, China, as a Cheung Kong Chair Professor from 2008 to 2011, and the Dalian University of Technology as a National Thousand-Talent Chair Professor since 2011. His current research interests include neural networks and their applications.

Prof. Wang has served as an Associate Editor of the IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS - PART B since 2003 and an Editorial Advisory Board Member of the *International Journal of Neural Systems* since 2006. He served as an Associate Editor of the IEEE TRANSACTIONS ON NEURAL NETWORKS from 1999 to 2009 and the IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS - PART C from 2002 to 2005. He was a Guest Editor of special issues of the *European Journal of Operational Research* in 1996, the *International Journal of Neural Systems* in 2007, and *Neurocomputing* in 2008. He served as the President of the Asia Pacific Neural Network Assembly (APNNA) in 2006, the General Chair of 13th International Conference on Neural Information Processing in 2006, and the IEEE World Congress on Computational Intelligence in 2008. In addition, he has served on many IEEE committees such as the Fellow Committee. He is an IEEE Distinguished Lecturer from 2010 to 2012. He was the recipient of the Research Excellence Award from the Chinese University of Hong Kong from 2008 to 2009, a Shanghai Natural Science Award (first class) in 2009, the APNNA Outstanding Achievements Award, and the IEEE TRANSACTIONS ON NEURAL NETWORKS Outstanding Paper Award (with Qingshan Liu) in 2011.