IEEE TRANSACTIONS ON NEURAL NETWORKS

IEEE TRANSACTIONS ON NEURAL NETWORKS is published by the IEEE Computational Intelligence Society. Members may subscribe to this TRANSACTIONS for $56.00 per year. IEEE student members may subscribe for $28.00 per year. Nonmembers may subscribe for $1,645.00. For additional subscription information visit http://www.ieee.org/nns/pubs. For information on receiving this TRANSACTIONS, write to the IEEE Service Center at the address below. *Member copies of Transactions/Journals are for personal use only*. For more information about this TRANSACTIONS see http://www.ieee-cis/org/pubs/tnn.

IEEE TRANSACTIONS ON NEURAL NETWORKS (ISSN 1045-9227) is published monthly by The Institute of Electrical and Electronics Engineers, Inc. Responsibility for the contents rests upon the authors and not upon the IEEE, the Society/Council, or its members. **IEEE Corporate Office:** 3 Park Avenue, 17th Floor, New York, NY 10016-5997. **IEEE Operations Center:** 445 Hoes Lane, Piscataway, NJ 08854-4141. **NJ Telephone:** +1 732 981 0060. **Price/Publication Information:** Individual copies: IEEE Members $20.00 (first copy only), nonmembers $137.00 per copy. (Note: Postage and handling charge not included.) Member and nonmember subscription prices available upon request. Available in microfiche and microfilm. **Copyright and Reprint Permissions:** Abstracting is permitted with credit to the source. Libraries are permitted to photocopy for private use of patrons, provided the per-copy fee indicated in the code at the bottom of the first page is paid through the Copyright Clearance Center, 222 Rosewood Drive, Danvers, MA 01923. For all other copying, reprint, or republication permission, write to Copyrights and Permissions Department, IEEE Publications Administration, 445 Hoes Lane, Piscataway, NJ 08854-4141. Copyright © 2010 by The IEEE, Inc. All rights reserved. Periodicals Postage Paid at New York, NY and at additional mailing offices. **Postmaster:** Send address changes to IEEE TRANSACTIONS ON NEURAL NETWORKS, IEEE, 445 Hoes Lane, Piscataway, NJ 08854-4141. GST Registration No. 125634188. CPC Sales Agreement #40013087. Return undeliverable Canada addresses to: Pitney Bowes IMEX, P.O. Box 4332, Stanton Rd., Toronto, ON M5W 3J4, Canada. IEEE prohibits discrimination, harassment and bullying. For more information visit http://www.ieee.org/nondiscrimination. Printed in U.S.A.

Digital Object Identifier 10.1109/TNN.2010.2042018

Brief Papers

Quasi-Lagrangian Neural Network for Convex Quadratic Optimization

Giovanni Costantini, Renzo Perfetti, and Massimiliano Todisco

***Abstract*—A new neural network for convex quadratic optimization is presented in this brief. The proposed network can handle both equality and inequality constraints, as well as bound constraints on the optimization variables. It is based on the Lagrangian approach, but exploits a partial dual method in order to keep the number of variables at minimum. The dynamic evolution is globally convergent and the steady-state solutions satisfy the necessary and sufficient conditions of optimality. The circuit implementation is simpler with respect to existing solutions for the same class of problems. The validity of the proposed approach is verified through some simulation examples.**

***Index Terms*—Analog circuits, Lagrangian networks, mathematical programming, quadratic optimization, recurrent neural networks.**

I. Introduction

The idea of using analogue circuits to solve mathematical programming problems can be traced back to the works of Pyne [1] and Dennis [2]. A canonical nonlinear programming circuit was proposed by Chua and Lin [3], later extended by Wilson [4]. Kennedy and Chua [5] recast the canonical circuit in a neural network framework and prove the stability. All the networks in [3]–[5] are based on the penalty function method, which gives exact solutions only if the penalty parameter tends to infinity, a condition impossible to meet in practice. To avoid the penalty functions, Zhang and Constantinides [6] proposed a Lagrangian approach to solve quadratic programming (QP) problems with equality constraints. The method can be extended to problems including both equality and inequality constraints converting inequalities into equalities by introducing slack variables. Also bound constraints on the variables, often arising in practical problems, can be treated in the same way at the expense of a huge number of variables. To overcome the penalty method, handling both equality and inequality constraints as well as bounds on the variables, a primal-dual approach [7] and a gradient-based neural network [8] have been proposed. Recently, Xia and Feng [9] proposed a one-layer neural network for convex QP, based on the projection method. To the best of our knowledge, the scheme in [9] represents the simplest available neural network for real-time quadratic optimization.

In this brief, which extends a previous work on the solution of the optimization problem arising in the context of support vector machine learning [10], we propose a new neural network solver for general QP problems. It is inspired by the Lagrangian approach of Zhang and Constantinides but exploits a partial dual method to keep the number of variables at minimum, and the idea of Bouzerdoum and Pattison [11] to implement the bounds both on optimization variables and Lagrange multipliers. The proposed network does not require slack variables, or transformation of inequality constraints into equality constraints. The circuit implementation is simpler than existing solutions; in particular, it requires the same number of connections but less components with respect to the network in [9].

The rest of this brief is organized as follows. In Section II, the optimization problem is formulated. In Section III, the proposed neural network is introduced and illustrated. In Section IV, convergence and stability are proved. Section V illustrates the circuit implementation. Section VI presents some simulation examples. Finally, some comments conclude this brief.

Manuscript received January 8, 2008; accepted May 28, 2008. First published September 26, 2008; current version published October 8, 2008.

R. Perfetti is with the Department of Electronic and Information Engineering, University of Perugia, I-06125 Perugia, Italy (e-mail: perfetti@diei.unipg.it).

G. Costantini and M. Todisco are with the Department of Electronic Engineering, University of Rome "Tor Vergata," Rome I-00100, Italy (e-mail: costantini@uniroma2.it; massimiliano.todisco@uniroma2.it).

Digital Object Identifier 10.1109/TNN.2008.2001183

II. Optimization Problem

We consider the following QP problem:

$$\min_{\mathbf{x}} J(\mathbf{x}) = \frac{1}{2}\mathbf{x}^{\mathrm{T}}\mathbf{Q}\mathbf{x} + \mathbf{p}^{\mathrm{T}}\mathbf{x} \tag{1a}$$

subject to

$$\mathbf{A}\mathbf{x} - \mathbf{d} \leq \mathbf{0} \tag{1b}$$

$$\mathbf{B}\mathbf{x} - \mathbf{c} = \mathbf{0} \tag{1c}$$

$$\mathbf{l} \leq \mathbf{x} \leq \mathbf{h} \tag{1d}$$

where $\mathbf{Q} \in \Re^{n\times n}$ is a symmetric and positive-definite matrix, $\mathbf{A} \in \Re^{r\times n}$, $\mathbf{B} \in \Re^{m\times n}$, $\mathbf{x}, \mathbf{p}, \mathbf{h}, \mathbf{l} \in \Re^{n}$; $\mathbf{c} \in \Re^{m}$, and $\mathbf{d} \in \Re^{r}$. Constraints (1b–d) can be recast as follows:

$$g_i(\mathbf{x}) \leq 0, \qquad i = 1, \ldots, r \tag{2}$$

$$f_i(\mathbf{x}) = 0, \qquad i = 1, \ldots, m \tag{3}$$

$$l_i \leq x_i \leq h_i, \qquad i = 1, \ldots, n \tag{4}$$

being

$$g_i(\mathbf{x}) = \sum_{j=1}^{n} a_{ij}x_j - d_i, \qquad i = 1, \ldots, r \tag{5}$$

$$f_i(\mathbf{x}) = \sum_{j=1}^{n} b_{ij}x_j - c_i, \qquad i = 1, \ldots, m. \tag{6}$$

III. Proposed Neural Network

The basic idea in Lagrangian duality is to take the constraints in (1) into account by augmenting the objective function with a weighted sum of the constraint functions [11]. To limit the complexity of the resulting neural network, we follow a *partial dual* approach defining the following Lagrangian function:

$$L(\mathbf{x}, \boldsymbol{\alpha}, \boldsymbol{\beta}) = J(\mathbf{x}) + \sum_{j=1}^{r} \alpha_j g_j(\mathbf{x}) + \sum_{j=1}^{m} \beta_j f_j(\mathbf{x}) \tag{7}$$

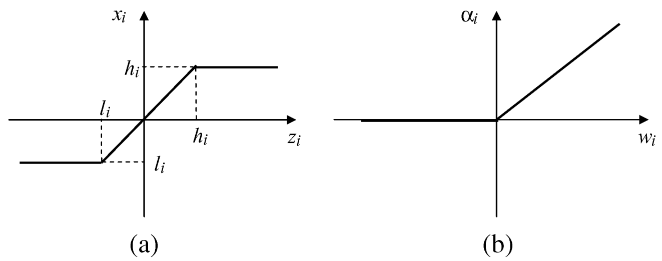

Fig. 1. Piecewise linear function P_1 and P_2.

where α_i is the Lagrange multiplier associated with the ith inequality constraint, satisfying

$$\alpha_i \geq 0 \qquad \forall\, i \tag{8}$$

and β_i is the Lagrange multiplier associated with the ith equality constraint. Note that constraints (4) are not included in (7), avoiding $2n$ additional dual variables. The solution of problem (1) corresponds to the saddle point of the Lagrangian function (7), which must be maximized with respect to $\boldsymbol{\alpha}, \boldsymbol{\beta}$ and minimized with respect to $\mathbf{x}$. Following [6], a gradient dynamical system can be used such that, along a trajectory, function (7) is increasing with each α_i and β_i and decreasing with each x_i. To ensure that constraints (4) and (8) are satisfied, avoiding the drawbacks of the penalty approach, we extend the method suggested in [7] to take into account bound constraints. To this end, we introduce the state variables $z_i \in \Re, i = 1, \ldots, n$, and $w_i \in \Re, i = 1, \ldots, r$, being

$$x_i = \mathrm{P}_1(z_i) = \begin{cases} l_i, & z_i \leq l_i \\ z_i, & l_i < z_i < h_i \\ h_i & z_i, \geq h_i \end{cases} \tag{9}$$

$$\alpha_i = \mathrm{P}_2(w_i) = \begin{cases} 0, & w_i < 0 \\ w_i & w_i \geq 0. \end{cases} \tag{10}$$

The piecewise linear functions P_1 and P_2, depicted in Fig. 1, guarantee the fulfillment of constraints (4) and (8).

The proposed dynamical system is described by the following state equations:

$$\begin{aligned} \tau \dot{z}_i &= -\frac{\partial L}{\partial x_i} + \lambda_i (x_i - z_i) \\ &= -\sum_{j=1}^{n} q_{ij} x_j - p_i - \sum_{j=1}^{r} \alpha_j a_{ji} - \sum_{j=1}^{m} \beta_j b_{ji} + \lambda_i (x_i - z_i) \end{aligned} \tag{11a}$$

$$\begin{aligned} \tau \dot{w}_i &= \frac{\partial L}{\partial \alpha_i} + \mu_i (\alpha_i - w_i) \\ &= g_i(\mathbf{x}) + \mu_i (\alpha_i - w_i) \end{aligned} \tag{11b}$$

$$\tau \dot{\beta}_i = \frac{\partial L}{\partial \beta_i} = f_i(\mathbf{x}) \tag{11c}$$

where $\lambda_i, \mu_i > 0$, and $\tau > 0$ is a time scaling factor. Equations (11) represent a gradient dynamical system seeking for the saddle point of the Lagrangian, with the additional terms $\lambda_i(x_i - z_i)$ and $\mu_i(\alpha_i - w_i)$, which guarantee the existence of a finite equilibrium point, as it will be shown in Section IV.

Equation (11) can be realized by the recurrent neural network shown in Fig. 2. It is composed of $n + m + r$ integrators, n limiters realizing function (9), and r limiters with transfer function (10).

The equilibrium conditions of system (11) are

$$\frac{\partial L}{\partial x_i} = \lambda_i (x_i - z_i), \qquad i = 1, \ldots, n \tag{12}$$

$$g_i(\mathbf{x}) = -\mu_i (\alpha_i - w_i), \qquad i = 1, \ldots, r \tag{13}$$

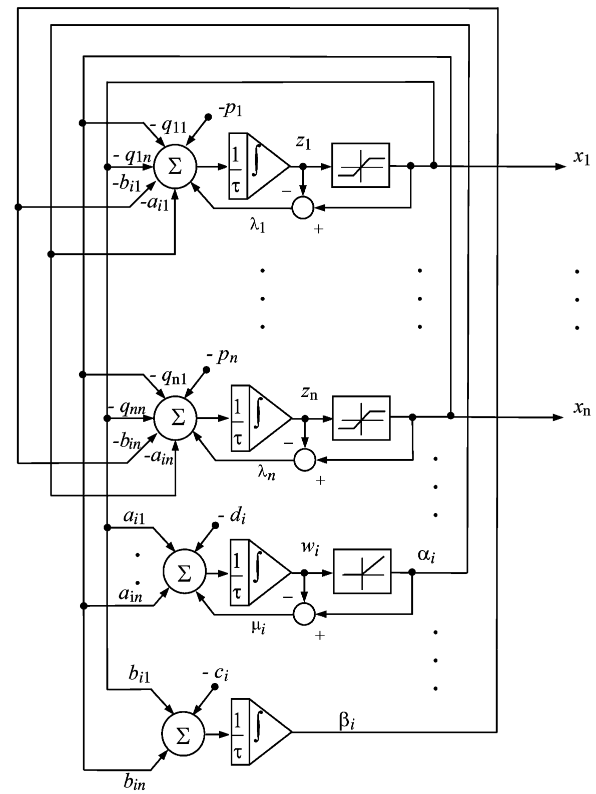

Fig. 2. Scheme of the proposed neural network.

$$f_i(\mathbf{x}) = 0, \qquad i = 1, \ldots, m. \tag{14}$$

Concerning (12), we have three cases

$$l_i < x_i < h_i \qquad \frac{\partial L}{\partial x_i} = 0 \tag{15a}$$

$$x_i = l_i \qquad \frac{\partial L}{\partial x_i} \geq 0 \tag{15b}$$

$$x_i = h_i \qquad \frac{\partial L}{\partial x_i} \leq 0. \tag{15c}$$

Moreover, taking into account the form of $\mathrm{P}_2(.)$, (13) corresponds to the following conditions:

$$g_i(\mathbf{x}) \leq 0, \qquad i = 1, \ldots, r \tag{16a}$$

$$\alpha_i \geq 0, \qquad i = 1, \ldots, r \tag{16b}$$

$$\alpha_i g_i(\mathbf{x}) = 0, \qquad i = 1, \ldots, r. \tag{16c}$$

It is easy to verify that (14)–(16) represent the Karush–Kuhn–Tucker (KKT) conditions of problem (1) [7], [11]. Hence, because the optimization problem is convex, each equilibrium point of system (11) corresponds to an optimal (global) solution of (1).

IV. Convergence Analysis

A dynamical system is said to be *globally convergent* if every trajectory converges to the unique equilibrium point. We will prove that the proposed neural network is globally convergent under the hypothesis that the Hessian matrix $\mathbf{Q}$ is positive definite.

Theorem: If $\mathbf{Q}$ is positive definite, the dynamical system described by (11) is globally convergent.

Proof: By hypothesis, the necessary and sufficient KKT conditions (14)–(16) have a unique solution $(\mathbf{x}^*, \boldsymbol{\alpha}^*, \boldsymbol{\beta}^*)$. The equilibrium points satisfy the KKT conditions. Hence, each equilibrium point has

$\boldsymbol{\beta} = \boldsymbol{\beta}^*$. We will prove that there is only one equilibrium point ($\mathbf{z}^*$, $\mathbf{w}^*$, $\boldsymbol{\beta}^*$) mapped by $\mathrm{P}_1(.)$ and $\mathrm{P}_2(.)$ onto ($\mathbf{x}^*$, $\boldsymbol{\alpha}^*$, $\boldsymbol{\beta}^*$). If $x_i^* \in \{l_i, h_i\}$, the equilibrium points share the ith component z_i that is imposed by (15b) or (15c). For $l_i < x_i < h_i$, the activation function $\mathrm{P}_1(.)$ is invertible, i.e., $z_i = x_i^*$; hence, the equilibrium points share also the components z_i in the linear range. If $\alpha_i^* = 0$, the equilibrium points share the ith component w_i that is imposed by (13). For $\alpha_i > 0$, the activation function $\mathrm{P}_2(.)$ is invertible, i.e., $w_i = \alpha_i^*$; hence, the equilibrium points share also the components w_i in the linear range. Hence, the equilibrium point of system (11) is unique.

Now, we prove that every trajectory converges to the equilibrium point ($\mathbf{z}^*$, $\mathbf{w}^*$, $\boldsymbol{\beta}^*$).

Let

$$x_i' = x_i - x_i^* = \mathrm{P}_1(z_i) - \mathrm{P}_1(z_i^*) \tag{17a}$$
$$\alpha_i' = \alpha_i - \alpha_i^* = \mathrm{P}_2(w_i) - \mathrm{P}_2(w_i^*) \tag{17b}$$
$$\beta_i' = \beta_i - \beta_i^* \tag{17c}$$
$$z_i' = z_i - z_i^* \tag{17d}$$
$$w_i' = w_i - w_i^*. \tag{17e}$$

From the definition of $\mathrm{P}_1(.)$ and $\mathrm{P}_2(.)$, it follows:

$$0 \le \frac{x_i - x_i^*}{z_i - z_i^*} \le 1, \qquad i = 1, \ldots, n \tag{18a}$$
$$0 \le \frac{\alpha_i - \alpha_i^*}{w_i - w_i^*} \le 1, \qquad i = 1, \ldots, r. \tag{18b}$$

Substitution of (17) in (11) gives (we assume $\tau = 1$ for simplicity)

$$\dot{z}_i' = -\sum_{j=1}^{n} q_{ij} x_j' - \sum_{j=1}^{r} \alpha_j' a_{ji} - \sum_{j=1}^{m} \beta_j' b_{ji} + \lambda_i \left(x_i' - z_i'\right) \tag{19a}$$
$$\dot{w}_i' = \sum_{j=1}^{n} a_{ij} x_j' + \mu_i \left(\alpha_i' - w_i'\right) \tag{19b}$$
$$\dot{\beta}_i' = \sum_{j=1}^{n} b_{ij} x_j'. \tag{19c}$$

Consider the following candidate Liapunov function:

$$V(\mathbf{z}, \mathbf{w}, \boldsymbol{\beta}) = \sum_{i=1}^{n} \int_0^{z_i'} x_i'(\xi) d\xi + \sum_{i=1}^{r} \int_0^{w_i'} \alpha_i'(\xi) d\xi + \frac{1}{2} \sum_{i=1}^{m} \beta_i'^2. \tag{20}$$

This kind of function has already been used in the stability analysis of some neural network models for optimization [12]. V is nonnegative as a consequence of (18). Taking the time derivative and using (19), we have

$$\begin{aligned}
\dot{V}(\mathbf{z}, \mathbf{w}, \boldsymbol{\beta}) &= \sum_{i=1}^{n} \frac{\partial V}{\partial z_i'} \dot{z}_i' + \sum_{i=1}^{r} \frac{\partial V}{\partial w_i'} \dot{w}_i' + \sum_{i=1}^{m} \frac{\partial V}{\partial \beta_i'} \dot{\beta}_i' \\
&= \sum_{i=1}^{n} x_i' \left[-\sum_{j=1}^{n} q_{ij} x_j' - \sum_{j=1}^{r} \alpha_j' a_{ji} \right. \\
&\qquad \left. - \sum_{j=1}^{m} \beta_j' b_{ji} + \lambda_i \left(x_i' - z_i'\right) \right] \\
&\quad + \sum_{i=1}^{r} \alpha_i' \left[\sum_{j=1}^{n} a_{ij} x_j' + \mu_i \left(\alpha_i' - w_i'\right) \right] + \sum_{i=1}^{m} \beta_i' \sum_{j=1}^{n} b_{ij} x_j' \\
&= -\sum_{i=1}^{n} \sum_{j=1}^{n} x_i' x_j' q_{ij} + \sum_{i=1}^{n} \lambda_i x_i' \left(x_i' - z_i'\right) \\
&\quad + \sum_{i=1}^{r} \mu_i \alpha_i' \left(\alpha_i' - w_i'\right).
\end{aligned} \tag{21}$$

From the positive definiteness of $\mathbf{Q}$ and from (18), it follows $\dot{V} \le 0$. V is radially unbounded with respect to $\boldsymbol{\beta}'$, hence $\boldsymbol{\beta}(t)$ must be bounded. Moreover, V is radially unbounded with respect to $\alpha_i' > 0$, hence $\boldsymbol{\alpha}(t)$ must be bounded. Taking into account the boundedness of $\boldsymbol{\beta}$, $\boldsymbol{\alpha}$, and $\mathbf{x}$, it is

$$\left| \sum_{j=1}^{n} q_{ij} x_j + p_i + \sum_{j=1}^{r} \alpha_j a_{ji} + \sum_{j=1}^{m} \beta_j b_{ji} \right| \le M < \infty \tag{22}$$

for every t. From (11a) and the positivity of λ_i, we have $\dot{z}_i < 0$ for $z_i > h_i + \mathrm{M}/\lambda_i$, $\dot{z}_i > 0$ for $z_i < l_i - \mathrm{M}/\lambda_i$. Hence, $\mathbf{z}(t)$ is bounded, and in the same way, we can prove that $\mathbf{w}(t)$ is bounded.

Because the trajectory is bounded and V is a global Liapunov function, from LaSalle's invariance principle [13], it follows that each trajectory will converge to the largest invariant subset of the set Ω, where $\dot{V} = 0$. From (21), we see that $\dot{V} = 0$ if and only if, $\forall i$, $x_i' = 0$, and $\alpha_i' = 0$ or $\alpha_i' = w_i'$. Assuming $x_i' = 0\ \forall i$, we have $\dot{\beta}_i' = 0$ from (19c) and $\dot{w}_i' = \mu_i(\alpha_i' - w_i')$ from (19b). When $\alpha_i' = w_i'$, it is $\dot{w}_i' = \dot{\alpha}_i' = 0$. Hence, $x_i(\mathrm{t}), \alpha_i(\mathrm{t}), \beta_i(t) \to$ constant as $t \to \infty$, and (11a) and (11b) can be rewritten as follows:

$$\dot{z}_i = -\lambda_i z_i + \phi_i(\mathrm{t})$$
$$\dot{w}_i = -\mu_i w_i + \gamma_i(\mathrm{t})$$

where functions $\phi_i(\mathrm{t})$ and $\gamma_i(\mathrm{t})$ have a finite limit as $t \to \infty$.

Because λ_i, $\mu_i > 0$, using well-known properties of differential equations, we have

$$\lim_{t \to \infty} z_i(t) = \phi_i(\infty)/\lambda_i$$
$$\lim_{t \to \infty} w_i(t) = \gamma_i(\infty)/\mu_i.$$

All the limits being finite, they must correspond to the unique equilibrium point of system (11). Hence, we conclude that every trajectory converges to ($\mathbf{z}^*$, $\mathbf{w}^*$, $\boldsymbol{\beta}^*$). □

Remark: The solution is guaranteed to be unique if the Hessian $\mathbf{Q}$ is positive definite. If $\mathbf{Q}$ is positive semidefinite (p.s.d.), multiple solutions exist and the theorem is not applicable. However, computer simulations showed a convergent behavior even in this case: the network converges to one of the, possibly infinite, solutions depending on the initial conditions (see Example 2 in Section VI). Computer simulations showed also that the property of global convergence is structurally stable. It was verified in presence of random errors on the connection matrices with weight relative error up to 5% (the results are not included due to lack of space).

V. Circuit Implementation

The network in Fig. 2 is similar to the scheme proposed in [9] with two main differences: 1) the saturation nonlinearities are moved from the input to the output of the integrators; 2) there is a different self-feedback loop around the integrators, corresponding to the terms $\lambda_i(x_i - z_i)$ and $\mu_i(\alpha_i - w_i)$. These modifications allow a simpler circuit realization, because both saturation and self-feedback can be implicitly obtained by the operational amplifier used to implement the integrator.

For sake of clarity, let us consider ith (11a) corresponding to the scheme in Fig. 3(a); it can be realized by the inverting integrator shown in Fig. 3(b). The projection operator $\mathrm{P}_1(.)$ is implicitly obtained by the saturation of the op-amp, assuming bipolar voltage supply. The saturation values V_H and V_L can be different, respectively, from h_i and l_i. In this case, the voltages v_i^x are scaled versions of the QP variables

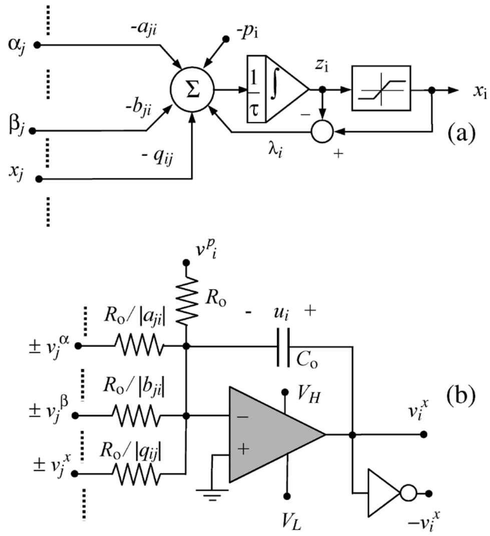

Fig. 3. (a) Scheme corresponding to the ith state equation (11a); (b) its circuit implementation.

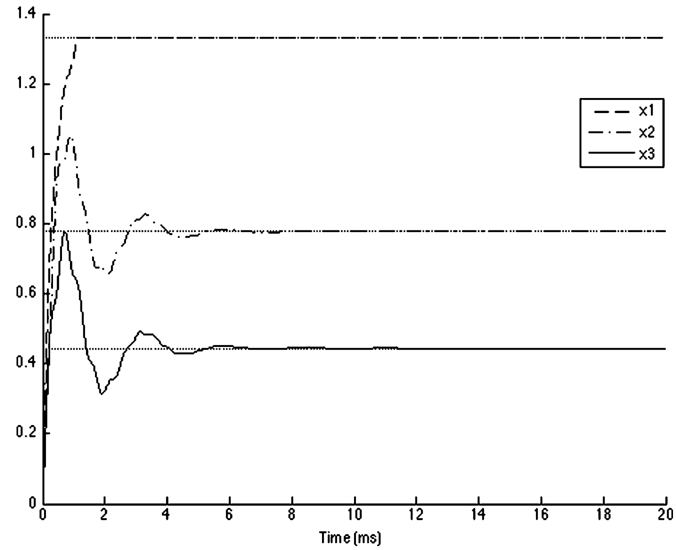

Fig. 4. Transient behavior of the network from zero initial conditions for Example 1.

x_i. The connections are realized with v_j^x $(-v_j^x)$ if $q_{ij} > 0$ $(q_{ij} < 0)$ through a resistance $R_o/|q_{ij}|$, where R_o is a normalization resistance. An inverting amplifier provides the voltage $-v_i^x$. The remaining connections are obtained in the same way. Voltages v_j^α and v_j^β correspond, respectively, to the variables α_j and β_j. Voltage v_i^p corresponds to the constant p_i.

Applying Kirchhoff's current law at the inverting input of the op-amp and taking into account that this input is not a virtual ground at saturation, we obtain the following equation:

$$R_o C_o \dot{u}_i = -\sum_{j=1}^{n} q_{ij} v_j^x - v_i^p - \sum_{j=1}^{r} v_j^\alpha a_{ji} - \sum_{j=1}^{m} v_j^\beta b_{ji} + \lambda_i \left(v_i^x - u_i\right) \tag{23}$$

where

$$\lambda_i = 1 + \sum_{j=1}^{n} |q_{ij}| + \sum_{j=1}^{r} |a_{ji}| + \sum_{j=1}^{m} |b_{ji}|. \tag{24}$$

It is $v_i^x = u_i$ if $\mathrm{V}_L < u_i < V_H$; $v_i^x = 0$ if $u_i \le \mathrm{V}_L$; $v_i^x = V_H$ if $u_i \ge V_H$. Equation (23) is equivalent to (11a) with $\tau = R_o C_o$. It is worth noting that the last term in (23), due to the nonzero differential input voltage of the op-amp at saturation, provides the corrective term $\lambda_i(x_i - z_i)$ in (11a) without extra circuitry. Equation (11b) can be realized by a similar circuit with the difference that the lower saturation voltage of the op-amp is zero and positive saturation must be avoided; in this case, it is

$$\mu_i = 1 + \sum_{j=1}^{n} |a_{ij}|. \tag{25}$$

Equation (11c) can be realized by a simple linear integrator.

With respect to the network in [9], the proposed circuit requires the same number of connections and the same number of integrators, but avoids the $n + r$ piecewise-linear activation functions and the $n + r$ summers.

VI. Simulation Examples

Example 1: Consider the following QP problem [9]:

$$\begin{aligned}
\text{Minimize} \quad & J(\mathbf{x}) = x_1^2 + x_2^2 + 0.5x_3^2 + x_1(x_2 + x_3 - 4) \\
& \qquad - 3x_2 - 2x_3 \\
\text{subject to} \quad & x_1 - x_2 + x_3 \le 1 \\
& -x_1 + x_2 + x_3 \le -1/9 \\
& 2x_1 - 2x_2 + 2x_3 \le 2 \\
& 3x_1 - 9x_2 + 9x_3 = 1 \\
& x_1 + x_2 + 2x_3 = 3 \\
& 0 \le x_i \le 4/3, \qquad i = 1, 2, 3.
\end{aligned} \tag{26}$$

It can be shown that problem (26) has a unique optimal solution $\mathbf{x}^* = [4/3 \quad 7/9 \quad 4/9]^{\mathrm{T}}$.

Problem (26) can be recast in a matrix form as follows:

$$\begin{aligned}
&\mathbf{l} = [0\ 0\ 0]^{\mathrm{T}} && \mathbf{h} = [4/3\ 4/3\ 4/3]^{\mathrm{T}} \\
&\mathbf{Q} = \begin{bmatrix} 2 & 1 & 1 \\ 1 & 2 & 0 \\ 1 & 0 & 1 \end{bmatrix} && \mathbf{p} = \begin{bmatrix} -4 \\ -3 \\ -2 \end{bmatrix} \\
&\mathbf{A} = \begin{bmatrix} 1 & -1 & 1 \\ -1 & 1 & 1 \\ 2 & -2 & 2 \end{bmatrix} && \mathbf{d} = \begin{bmatrix} 1 \\ -1/9 \\ 2 \end{bmatrix} \\
&\mathbf{B} = \begin{bmatrix} 3 & -9 & 9 \\ 1 & 1 & 2 \end{bmatrix} && \mathbf{c} = \begin{bmatrix} 1 \\ 3 \end{bmatrix}.
\end{aligned} \tag{27}$$

The corresponding neural network has been simulated by numerical integration of (11), using Euler's method. Expressions (24) and (25) have been used for λ_i and μ_i, even if these values do not influence the steady-state solution, as proved in Section V. In Fig. 4, the transient behavior of the network is shown, starting from zero initial conditions (all the state variables are zero in $t = 0$). The time constant is $\tau = 1$ ms. Convergence to the correct solution is observed within 5 ms.

To verify the global convergence, the network was simulated starting from different initial conditions. The network always converged to the correct solution, regardless of whether the initial point is inside or

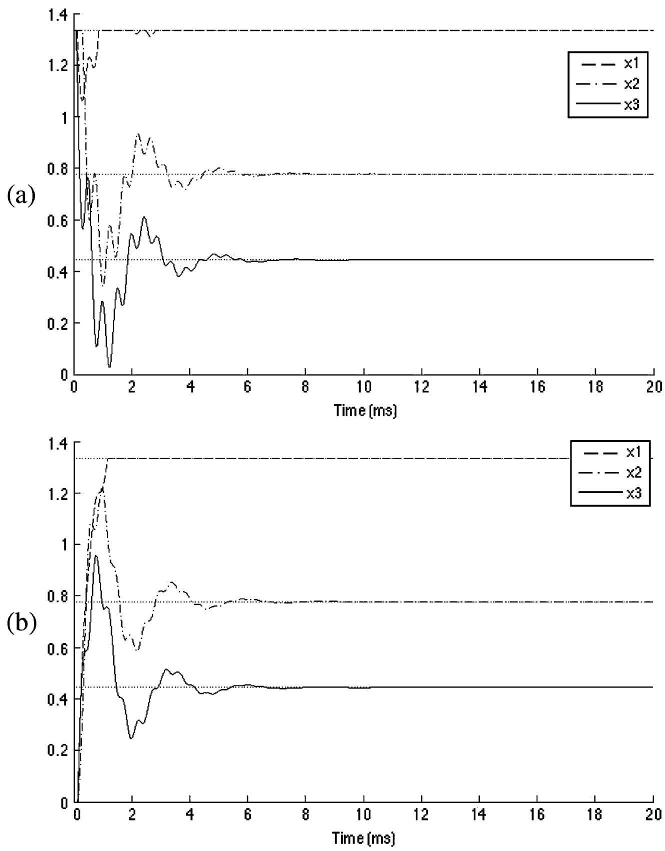

Fig. 5. Transient behavior for Example 1 with starting points (a) $\mathbf{z}(0) = [2\ 2\ 2]^{\mathrm{T}}$ and (b) $\mathbf{z}(0) = [-2\ -2\ -2]^{\mathrm{T}}$.

outside the feasible region. In particular, in Fig. 5, the transient behavior is shown for the starting points $\mathbf{z}(0) = [2\ 2\ 2]^{\mathrm{T}}$ and $\mathbf{z}(0) = [-2\ -2\ -2]^{\mathrm{T}}$. The remaining state variables were initially zero.

Finally, to investigate the effect of finite precision, a Monte Carlo simulation was performed by adding to the connection weights a random error with uniform distribution. Assuming a maximum relative error of 1%, we obtained the following mean values of the variables after 1000 runs (in parenthesis the standard deviation): $\mathrm{x}_1 = 1.3313\ (0.0055)$, $\mathrm{x}_2 = 0.7775\ (0.0049)$, $\mathrm{x}_3 = 0.4448\ (0.0040)$.

Example 2 [8]:

$$\begin{aligned} &\text{Minimize} \quad J(\mathbf{x}) = x_2^2 + x_3^2 \\ &\text{subject to} \quad x_1 \geq 0. \end{aligned}$$

This problem has infinitely many optimal solutions $\mathbf{x}^* = [x_1\ 0\ 0]^{\mathrm{T}}$ with $x_1 \geq 0$, corresponding to the optimal value of the objective function $J(\mathbf{x}^*) = 0$, so the theorem does not apply. Notwithstanding, the network converges to an optimal solution for arbitrary initial conditions. For example, from $\mathbf{z}(0) = [1\ 1\ 1]^{\mathrm{T}}$, the network converges to $[1\ 0\ 0]^{\mathrm{T}}$, and from $\mathbf{z}(0) = [-1\ -1\ -1]^{\mathrm{T}}$, the evolution converges to $[0\ 0\ 0]^{\mathrm{T}}$. The corresponding transient behavior is shown in Fig. 6.

Example 3: In this example, we verified the behavior of the circuit implementation described in Section V through a simulation with *PSpice*. To this end, we considered the following problem:

$$\begin{aligned} &\text{Minimize} \quad J(\mathbf{x}) = \frac{1}{2}\left(x_1^2 + x_2^2\right) \\ &\text{subject to} \quad x_1 + x_2 - 4 \geq 0 \\ &\qquad\qquad\quad 0 \leq x_1 \leq 1, \qquad 0 \leq x_2 \leq 4. \end{aligned}$$

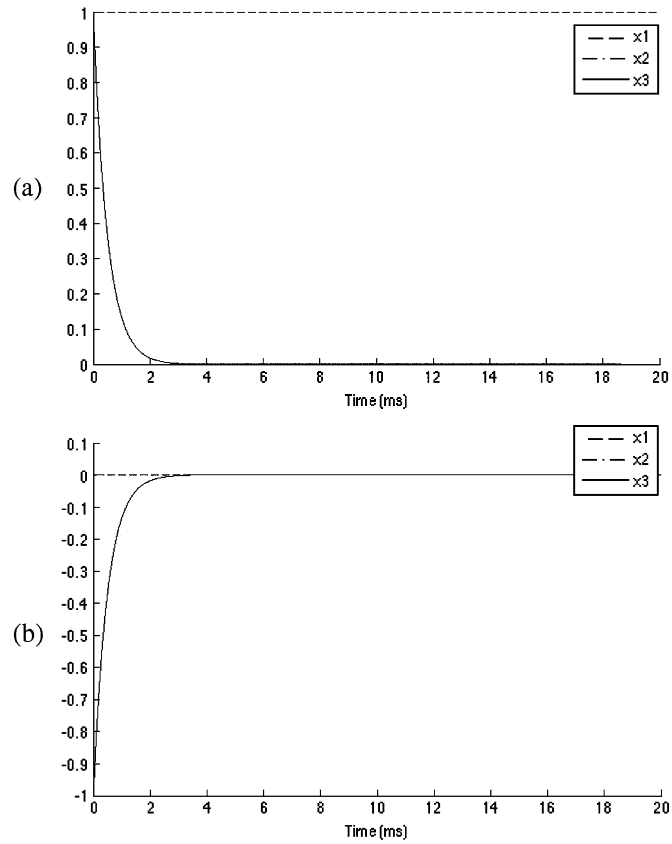

Fig. 6. Transient behavior for Example 2.

In matrix notation

$$\begin{aligned} \mathbf{l} &= [0\ 0]^{\mathrm{T}} & \mathbf{h} &= [1\ 4]^{\mathrm{T}} \\ \mathbf{Q} &= \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} & \mathbf{p} &= \begin{bmatrix} 0 \\ 0 \end{bmatrix} \\ \mathbf{A} &= [-1\ -1] & \mathbf{d} &= [-4] \\ \mathbf{B} &= 0 & \mathbf{c} &= 0. \end{aligned} \tag{28}$$

The optimal solution is $x_1 = 1$ and $x_2 = 3$. Even if it is trivial, this example shows the effects of physical devices and finite accuracy of connection resistors. Moreover, it exhibits the two different steady-state behaviors since x_1 saturates while x_2 does not. The network was realized using 1% tolerance resistors and TL082 op-amps; the normalization resistance and capacitance were $R_{\mathrm{o}} = 1$ kΩ and $C_{\mathrm{o}} = 1$ μF. The circuit requires four op-amps: two integrators for the variables, one integrator for the inequality constraint, and one inverting amplifier to realize the α_1 term in (11a). Saturation at zero is obtained by using a voltage supply $\mathrm{V}_L = -0.678$ V. Op-amp integrators for the optimization variables have positive saturation levels 3 and 12 V, respectively (voltage supplies $\mathrm{V}_H = 3.678$ V and $\mathrm{V}_H = 12.678$ V). As a consequence, the output voltages v_1^x and v_2^x correspond, respectively, to $3x_1$ and $3x_2$. Hence, the constant input -4, corresponding to d_1, must be scaled accordingly (-12 V). The capacitors were initially uncharged. An example of transient behavior is depicted in Fig. 7. The final values of the variables are $v_1^x \cong 3$ V and $v_2^x \cong 9$ V, corresponding to the exact solution.

VII. Conclusion

In this brief, a novel neural model for the real-time solution of convex quadratic optimization problems has been presented. It is obtained by exploiting a partial dual Lagrangian approach to avoid slack variables.

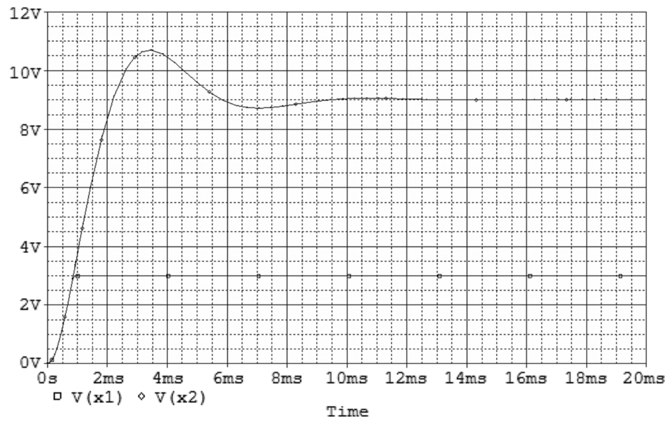

Fig. 7. PSpice simulation concerning Example 3.

Moreover, some operations (limitation and self-feedback) can be embedded in the op-amp integrators. The resulting circuit implementation requires less components with respect to existing neural networks for the same class of problems. The global convergence to a unique equilibrium point has been proven using a Liapunov function approach. The simulation results confirm the robust behavior of the proposed network and the accuracy of steady-state solutions, also in physical realizations with real-life devices.

REFERENCES

[1] I. B. Pyne, "Linear programming on an electronic analogue computer," *Trans. Amer. Inst. Electr. Eng.*, vol. 75, pp. 139–143, 1956.

[2] J. B. Dennis, *Mathematical Programming and Electrical Networks*. London, U.K.: Chapman & Hall, 1959.

[3] L. O. Chua and G.-N. Lin, "Nonlinear programming without computation," *IEEE Trans. Circuits Syst.*, vol. CAS-31, no. 2, pp. 182–188, Feb. 1984.

[4] G. Wilson, "Quadratic programming analogs," *IEEE Trans. Circuits Syst.*, vol. CAS-33, no. 9, pp. 907–911, Sep. 1986.

[5] M. P. Kennedy and L. O. Chua, "Neural networks for nonlinear programming," *IEEE Trans. Circuits Syst.*, vol. CAS-35, no. 5, pp. 554–562, May 1988.

[6] S. Zhang and A. G. Constantinides, "Lagrange programming neural networks," *IEEE Trans. Circuits Syst. II, Analog Digit. Signal Process.*, vol. 39, no. 7, pp. 441–452, Jul. 1992.

[7] Y. Xia, "A new neural network for solving linear and quadratic programming problems," *IEEE Trans. Neural Netw.*, vol. 7, no. 6, pp. 1544–1547, Nov. 1996.

[8] Y. Leung, K.-Z. Chen, Y.-C. Jiao, X.-B. Gao, and K. S. Leung, "A new gradient-based neural network for solving linear and quadratic programming problems," *IEEE Trans. Neural Netw.*, vol. 12, no. 5, pp. 1074–1083, Sep. 2001.

[9] Y. Xia and G. Feng, "An improved neural network for convex quadratic optimization with application to real-time beamforming," *Neurocomputing*, vol. 64, pp. 359–374, 2005.

[10] R. Perfetti and E. Ricci, "Analog neural network for support vector machine learning," *IEEE Trans. Neural Netw.*, vol. 17, no. 4, pp. 1085–1091, Jul. 2006.

[11] A. Bouzerdoum and T. R. Pattison, "Neural network for quadratic optimization with bound constraints," *IEEE Trans. Neural Netw.*, vol. 4, no. 2, pp. 293–304, Mar. 1993.

[12] M. Bazaraa, D. Sherali, and C. Shetty, *Nonlinear Programming: Theory and Algorithms*, ser. Wiley-Interscience Series in Discrete Mathematics and Optimization. New York: Wiley, 1992.

[13] M. Forti and A. Tesi, "New conditions for global stability of neural networks with applications to linear and quadratic programming problems," *IEEE Trans. Circuits Syst. I, Fundam. Theory Appl.*, vol. 42, no. 7, pp. 354–366, Jul. 1995.

[14] J. La Salle, *The Stability of Dynamical Systems*. Philadelphia, PA: SIAM, 1976.

[15] J. K. Hale, *Ordinary Differential Equations*. New York: Wiley, 1969.

Global μ-Synchronization of Linearly Coupled Unbounded Time-Varying Delayed Neural Networks With Unbounded Delayed Coupling

Tianping Chen, Wei Wu, and Wenjuan Zhou

***Abstract*—In this brief, we study the global synchronization of linearly coupled neural networks with delayed couplings, where the intrinsic systems are recurrently connected neural networks with unbounded time-varying delays, and the couplings include instant couplings and unbounded delayed couplings. The concept of μ-synchronization is introduced. Some sufficient conditions are derived for the global μ-synchronization for the underlined coupled systems.**

***Index Terms*—Linearly coupled recurrently connected neural networks, unbounded time-varying delay, global μ-synchronization, moore-penrose inverse.**

I. INTRODUCTION

Complex networks have been widely investigated in science, engineering, and nature for decades due to its applications in chemical reactions, biological systems study, secure communications, etc. Typical examples of complex networks include the Internet, World Wide Web (WWW), food webs, cellular and metabolic networks, etc. (e.g., see [1]–[8]). Stability, bifurcation, and chaos synchronization are also studied by many researchers.

Linearly coupled neural networks provide a large class of models that can be used to describe coupled systems with continuous time and state values, as well as discrete spatial states in many research fields. The dynamical behavior of a coupled network is governed by the following two mechanisms: the intrinsic nonlinear dynamics of the neural network at each node and the diffusion due to the spatial coupling among nodes. They have been investigated as theoretical models of spatio–temporal phenomena sof complex networks (for example, see [9]).

Because chaos synchronization in an array of linearly coupled dynamical systems was investigated by Pecora in [1], many results on local and global synchronization in various coupled systems have been obtained (for details, see comprehensive paper [8]). In [10], Lu and Chen gave criteria for local and global synchronization of linearly coupled dynamical systems. In [11], Wu and Chen discussed synchronization of coupled neural networks with time-varying coupling configuration. However, it is inevitable that time-delays occur due to the finite speeds of transmission and spreading as well as traffic congestions. Therefore, the study of delayed-coupling systems is quite important. There are also several papers investigating the coupled systems with a delay [12]–[18]. However, the delays are always constant and bounded.

Recurrently connected neural network with delays can be written as

$$\frac{dx_k(t)}{dt} = -d_k x_k(t) + \sum_{l=1}^{n} w_{kl} g_l(x_l(t)) + \sum_{l=1}^{n} w^1_{kl} f_l(x_l(t-\tau_{kl}(t))) + I_k(t), \qquad k=1,\ldots,n \tag{1}$$

Manuscript received November 14, 2007; accepted February 7, 2008. First published September 26, 2008; current version published October 8, 2008. This work was supported by the National Science Foundation of China under Grants 60574044 and 60774074.

The authors are with the Key Laboratory of Nonlinear Mathematics Science, School of Mathematical Sciences, Fudan University, Shanghai 200433, China (e-mail: tchen@fudan.edu.cn).

Digital Object Identifier 10.1109/TNN.2008.2001773