A Feedback Neural Network for Solving Nonlinear Programming Problems with Hybrid Constraints

Hamid Reza Vahabi
Department of Mathematics, Eslamshahr Branch,
Islamic Azad University, Tehran, Iran

Hasan Ghasabi-Oskoei
Department of Mathematics, Eslamshahr Branch,
Islamic Azad University, Tehran, Iran

ABSTRACT

This paper proposes a high-performance feedback neural network model for solving nonlinear convex programming problems with hybrid constraints in real time by means of the projection method. In contrary to the existing neural networks, this general model can operate not only on bound constraints, but also on hybrid constraints comprised of inequality and equality constraints. It is shown that the proposed neural network is stable in the sense of Lyapunov and can be globally convergent to an exact optimal solution of the original problem under some weaker conditions. Moreover, it has a simpler structure and a lower complexity. The advanced performance of the proposed neural network is demonstrated by simulation of several numerical examples.

General Terms

Neural Networks, Nonlinear Programming Problems.

Keywords

Nonlinear programming, Feedback neural network, Global convergence and stability.

1. INTRODUCTION

In the recent years many artificial neural networks developed to solve the optimization problems, because it has many important applications in wide variety of scientific and engineering fields including network economics, transportation science, game theory, military scheduling, automatic control, signal processing, regression analysis, structure design, mechanical design, electrical networks planning, and so on [1]. In many scientific and engineering applications, it is desire to have real-time on-line solutions of optimization problems [2]. Traditional optimization algorithms [1, 3, 4, 5, 6 and 7] are not suitable for real-time on-line implementation on digital computers. One promising approach to handle these difficulties is to employ an artificial neural network based on circuit implementation. The most important advantages of the neural networks are massively parallel processing and fast convergence. Many continuous-time neural networks for constrained optimization problems have been developed using penalty parameters [8, 9 and 10]. To avoid the penalty parameters, some significant works have done in recent years [11, 12, 13, 14, 15 and 16].

In this paper, we propose a class of neural networks with one hidden layer structure in order to solve nonlinear convex programming problems with hybrid constraints in real time. This model of neural networks has no adjustable parameters and therefore has lower complexity. The projection operator on a closed convex set is employed to describe the network to solve nonlinear programming problem. We define a suitable Liapunov function and prove the global convergence of the network.

The paper organized as follow. In the next section we introduce nonlinear programming problem with bound constraints and its equivalent formulation is described. In section 3, a feedback neural network model with circuit implementation is proposed. Section 4 discusses the stability of the proposed network and analyzes its global convergence. Extension of a proposed neural network for solving nonlinear programming problems with hybrid constraints is given in Section 5. In Section 6, numerical examples are simulated to show the reasonableness of our theory and demonstrate the high performance of proposed neural networks approach. A comparative analysis is presented in Section 7. Some conclusions are summarized in the last section.

2. NONLINEAR PROBLEM WITH BOUND CONSTRAINTS AND FORMULATION

Consider the following nonlinear convex programming problem with bound constraints:

$$\begin{aligned} &\min \quad f(x) \\ &s.t. \ \ Ax \le b, \\ &\qquad l \le x \le h \end{aligned} \tag{1}$$

where $f:\mathbb{R}^n \to \mathbb{R}$ is nonlinear, convex and twice continuously differentiable, $b \in \mathbb{R}^m$, A is $m \times n$ matrix and $x, l, h \in \mathbb{R}^n$.

Lemma 1. Suppose X is a closed convex set, for any $z \in \mathbb{R}^k$, there is a unique point $P_{\mathrm{X}}(z) \in \mathrm{X}$ such that

$$\|z - P_{\mathrm{X}}(z)\| = \inf_{u \in \mathrm{X}} \|z - u\|$$

$P_{\mathrm{X}}(z)$ is called the projection of z on X and P_{X} is called the projection operator on X. Moreover,

$$\forall u, z \in \mathbb{R}^k : \quad \|P_{\mathrm{X}}(z) - P_{\mathrm{X}}(u)\| \le \|z - u\|$$

For $u \in \mathbb{R}^k, \tilde{u} \in \mathrm{X}$, $(u - \tilde{u})^T(\tilde{u} - z) \ge 0$ for all $z \in \mathrm{X}$ if and only if $\tilde{u} = P_{\mathrm{X}}(u)$.

Proof. See [17 and 18].

We first give a sufficient and necessary condition for the solution of problem (1). It is the theoretical foundation for us to design the network for optimization problems.

Suppose that $\mathrm{X}_0 = \{y = (y_1, \ldots, y_m)^T \in \mathbb{R}^m\}$,

$P_{\mathrm{X}_0}(y) = (P_{\mathrm{X}_0}(y_1), \ldots, P_{\mathrm{X}_0}(y_m))^T$

and for $i = 1, \ldots, m$; $P_{\mathrm{X}_0}(y_i) = \max\{0, y_i\}$.

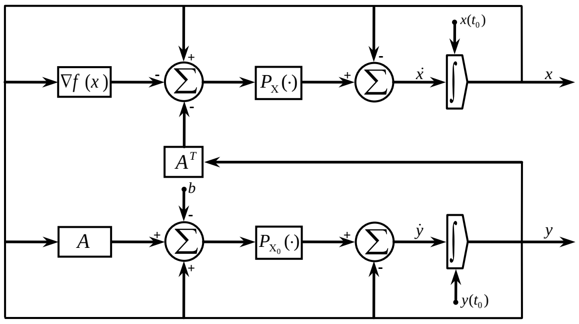

Fig. 1: Block diagram of the feedback neural network (3).

$$X = \{x = (x_1, \ldots, x_n)^T \in \mathbb{R}^n \,|\, l \le x \le h\},$$

$P_X(x) = (P_X(x_1), \ldots, P_X(x_n))^T$ and for $j = 1, \ldots, n$

$$P_X(x_j) = \begin{cases} l_j & x_j < l_j \\ x_j & l_j \le x_j \le h_j \\ h_j & x_j > h_j \end{cases}$$

According to the Karush-Kuhn-Tucker condition [19] and using the projection theorem [20], we can obtain easily the following lemma.

Lemma 2. x^* is an optimal solution of (1) if and only if there exists $y^* \in \mathbb{R}^m$, such that $(x^*, y^*)^T$ satisfies the following equations

$$\begin{cases} x - P_X(x - \nabla f(x) - A^T y) = 0 \\ y - P_{X_0}(y + Ax - b) = 0 \end{cases} \qquad (2)$$

Proof. See [20].

3. NEURAL NETWORK MODEL AND CIRCUIT IMPLEMENTATION

Based on the equivalent formulation in Lemma 2, we propose a feedback neural network for solving (1), with its time dependent dynamical system being given by

$$\begin{cases} \dfrac{dx}{dt} = P_X(x - \nabla f(x) - A^T y) - x \\ \dfrac{dy}{dt} = P_{X_0}(y + Ax - b) - y \end{cases} \qquad (3)$$

$$x(t_0) = x^0,\ y(t_0) = y^0$$

where $x \in \mathbb{R}^n, y \in \mathbb{R}^m$. Also $P_X(\cdot)$ and $P_{X_0}(\cdot)$ are projection operators on X and X_0. We shall prove that network (3) is globally convergent to the solutions set of problem (1).

The dynamical equations described by (3), can be easily realized by a feedback neural network with a single-layer structure as shown in Fig. 1. Where the vectors $x(t_0)$ and $y(t_0)$ are the initial external inputs and x and y are the network outputs.

4. STABILITY AND CONVERGENCE OF THE NETWORK

In this section, we shall study the dynamics of network (3). We first define a suitable Liapunov function and then prove the global convergence of network (3) in Theorem 2.

Theorem 1. For any $x(t_0) \in X$ and $y(t_0) \in X_0$, the neural network (3) has a unique solution $z(t) = (x(t), y(t))^T$.

Proof. Let

$$M(z) = \begin{Bmatrix} P_X(x - \nabla f(x) - A^T y) - x \\ P_{X_0}(y + Ax - b) - y \end{Bmatrix}$$

By lemma 1, note that $P_X(x - \nabla f(x) - A^T y)$ and $P_{X_0}(y + Ax - b)$ are Lipschitz continuous. Then it is easy to see that $M(z)$ is also Lipschitz continuous. From the existence and uniqueness theorem of ordinary differential equations [21], there exists a unique solution $z(t)$ with $z(t_0) = (x(t_0), y(t_0))^T$ for the neural network (3).

Theorem 2. Let

$$E_0(z) = (z - P_X(z - F(z)))^T F(z) - \frac{1}{2}\|P_X(z - F(z)) - z\|^2$$

and

$$F(z) = \begin{pmatrix} \nabla f(x) + A^T y \\ b - Ax \end{pmatrix},$$

then $E(z(t))$ that is defined as follow is Liapunov function of system (3).

$$E(z(t)) = E_0(z) + \frac{1}{2}\|z(t) - z^*\|^2$$

Where z^* is an equilibrium point of (3). Moreover, the network (3) is globally convergent to the solutions set of problem (1).

Proof. We have

$$\frac{dE_0(z)}{du} = F(z) - \nabla F(z)(P_X(z - F(z)) - z) + P_X(z - F(z)) - z$$

Let $g(z) = P_X(z - F(z)) - z$, then

$$\frac{dE(z)}{dt} = \frac{dE(z)}{dz}^T \frac{dz}{dt}$$

$$= \left(F(z) - \nabla F(z) g(z) + g(z) + z - z^*\right)^T g(z)$$

$$= \left(F(z) + z - z^*\right)^T g(z) + \|g(z)\|^2 - g(z)^T \nabla F(z) g(z)$$

In the inequality of Lemma 1, let $u = z - F(z)$ and let $x = z^*$, we get

$$(g(z) + z - z^*)^T (-g(z) - F(z)) \ge 0$$

Then

$$(F(z) + z - z^*)^T g(z) \le -F(z)^T (z - z^*) - \|g(z)\|^2$$

It follows that

$$\frac{dE(z)}{dt} \le -F(z)^T (z - z^*) - \|g(z)\|^2$$
$$+\|g(z)\|^2 - g(z)^T \nabla F(z) g(z)$$
$$= -F(z)^T (z - z^*) - g(z)^T \nabla F(z) g(z) \le 0$$

On the other side, in the inequality of Lemma 1, let $u = z - F(z)$ and let $x = z$, we obtain

$$F(z)^T g(z) \le -\|g(z)\|^2$$

It follows that $E_0(z) \ge \frac{1}{2}\|g(z)\|^2$ and thus

$$E(z) \ge \frac{1}{2}\left(\|g(z)\|^2 + \|z - z^*\|^2\right) \ge \frac{1}{2}\|z - z^*\|^2 .$$

Since $E(z)$ is positive definite and radially unbounded, for any initial point $z(t_0)$ there exists a convergent subsequence $\{z(t_k)\}$ such that $\lim_{k\to\infty} z(t_k) = \hat{z}$, where $\frac{dE(\hat{z})}{dt} = 0$. We next prove that $\frac{dE(\hat{z})}{dt} = 0$ if and only if $\hat{z}$ is a KKT point. Note that $\nabla F(z)$ is positive semi-define and $F(z)^T (z - z^*) \ge 0$. Then

$$\frac{dE(z)}{dt} \le -F(z)^T (z - z^*) - g(z)^T \nabla F(z) g(z) \le 0 .$$

Moreover, $\frac{dE(\hat{z})}{dt} = 0$ if and only if

$$\begin{cases} F(\hat{z})^T (\hat{z} - z^*) = 0 \\ \left(P_X(\hat{z} - F(\hat{z})) - \hat{z}\right)^T \nabla F(\hat{z}) \left(P_X(\hat{z} - F(\hat{z})) - \hat{z}\right) = 0 \end{cases}$$

It can be seen that

$$\left(P_X(\hat{z} - F(\hat{z})) - \hat{z}\right)^T \nabla F(\hat{z}) \left(P_X(\hat{z} - F(\hat{z})) - \hat{z}\right)$$
$$= \left(P_X(\hat{x} - \nabla f(\hat{x}) - A^T \hat{y}) - \hat{x}\right)^T \nabla^2 f(\hat{x}) \left(P_X(\hat{x} - \nabla f(\hat{x}) - A^T \hat{y}) - \hat{x}\right)$$

where $\hat{z} = (\hat{x}, \hat{y})^T$, and $\nabla^2 f(\hat{x})$ is Hessian matrix.

Since $\nabla^2 f(x)$ is positive definite

$$\left(P_X(\hat{x} - \nabla f(\hat{x}) - A^T \hat{y}) - \hat{x}\right) = 0$$

thus

$$\forall x \ge 0 : \quad (x - \hat{x})^T (\nabla f(\hat{x}) + A^T \hat{y}) \ge 0$$

using

$$\begin{cases} (x^* - \hat{x})^T (\nabla f(\hat{x}) + A^T \hat{y}) \ge 0 \\ (\hat{x} - x^*)^T (\nabla f(x^*) + A^T y^*) \ge 0 \end{cases}$$

we get

$$(\hat{x} - x^*)^T \left(\nabla f(x^*) + A^T y^* - (\nabla f(\hat{x}) + A^T \hat{y})\right) \ge 0$$

thus

$$(\hat{x} - x^*)^T \left(A^T y^* - A^T \hat{y}\right)$$
$$\ge (\hat{x} - x^*)^T \left(\nabla f(x^*) + A^T y^* - \nabla f(\hat{x}) - A^T y^*)\right).$$

On the other side, $P_X(x)$ is monotone.

That is, for any x and y

$$\left(P_X(x) - P_X(y)\right)^T (x - y) \ge 0$$

then

$$(\hat{x} - x^*)^T (\hat{x} - (\nabla f(\hat{x}) + A^T \hat{y})$$
$$- (x^* - (\nabla f(x^*) + A^T y^*))) \ge 0$$

It follows that

$$\|\hat{x} - x^*\|^2 \ge (\hat{x} - x^*)^T (A^T \hat{y} - A^T y^*)$$
$$\ge 2(\hat{x} - x^*)^T \left(\nabla f(\hat{x}) + A^T y^* - (\nabla f(x^*) + A^T y^*)\right)$$

let

$$\gamma = (\hat{x} - x^*)^T \left(\nabla f(\hat{x}) + A^T y^* - (\nabla f(x^*) + A^T y^*)\right)$$

If $\hat{x} \ne x^*$, then $\gamma > 0$, since $\nabla f(x) + A^T y^*$ is strictly monotone. It follows that when $\gamma = \|\hat{x} - x^*\|^2$ then

$$\|\hat{x} - x^*\|^2 \ge 2\|\hat{x} - x^*\|^2 > \|\hat{x} - x^*\|^2 .$$

This would constitute a contradiction. So $\hat{x} = x^*$ and $A\hat{x} - b \le 0$.

Now, $F(\hat{z})^T (\hat{z} - z^*) = 0$ implies that

$$(\hat{x} - x^*)^T (\nabla f(\hat{x}) + A^T \hat{y}) - (\hat{y} - y^*)^T (A\hat{x} - b) = 0 \quad .$$

Since $\hat{x} = x^*$, $(\hat{y} - y^*)^T (A\hat{x} - b) = 0$. Thus

$$(\hat{y})^T (A\hat{x} - b) = (y^*)^T (A\hat{x} - b) = (y^*)^T (Ax^* - b) = 0$$

Therefore, $\hat{z} = (\hat{x}, \hat{y})^T$ is a KKT point.

Finally, define again

$$\hat{E}(z(t)) = E_0(z(t)) + \frac{1}{2}\|z(t) - \hat{z}\|^2$$

Then, $\hat{E}(\hat{z}) = 0$ and thus $\lim_{k\to\infty} \hat{E}(z(t_k)) = \hat{E}(\hat{z}) = 0$.

So, for $\forall \varepsilon > 0$ there exists $\eta > 0$ such that for all $t_k \ge t_\eta$ we have $\hat{E}(z(t_k), \hat{z}) < \varepsilon$.

Similar to the previous analysis, we have $\frac{d}{dt}\hat{E}(z(t)) \le 0$.

So, for $t \ge t_\eta$

$$\|z(t) - \hat{z}\| \le 2\hat{E}(z(t)) \le \hat{E}(z(t_\eta)) \le 2\varepsilon .$$

It follows that $\lim_{t\to\infty}\|z(t) - \hat{z}\| = 0$ and $\lim_{t\to\infty} z(t) = \hat{z}$.

Therefore, the proposed neural network in (3) is globally convergent to a KKT point of (1).

5. NEURAL NETWORK FOR NONLINEAR PROBLEM WITH HYBRID CONSTRAINTS

Corollary 1. Suppose nonlinear programming problem is as follows

$$\begin{aligned} \min\quad & f(x) \\ s.t.\quad & A_1 x \le b_1 \\ & A_2 x = b_2 \\ & l \le x \le h \end{aligned} \tag{4}$$

If y and w are dual variables vectors of inequality and equality constraints, then the neural network (3) changes into the following form:

$$\begin{cases} \dfrac{dx}{dt} = P_X(x - \nabla f(x) - A_1^T y + A_2^T w) - x \\ \dfrac{dy}{dt} = P_{X_0}(y + A_1 x - b_1) - y \\ \dfrac{dw}{dt} = b_2 - A_2 x \end{cases} \tag{5}$$

$$x(t_0) = x^0,\ y(t_0) = y^0, w(t_0) = w^0$$

Corollary 2. We can have same analytical discussions about neural network (5) and handle them to reach optimum solution of problem (4).

6. SIMULATION EXAMPLES

We discuss the simulation results using numerical examples to demonstrate the global convergence property and effectiveness of the proposed neural networks. We have written a Matlab 2010 code for solving models (3) and (5) and executed the code on an Intel Corei5. Based on numerical simulations, the proposed models have a very fast convergence to exact optimal solutions of problems (1) and (4). This is one of the advantages of our networks in comparison with existing neural networks.

Example 1. Consider the following nonlinear programming problem:

$$\min \quad \frac{1}{4}x_1^4 + \frac{1}{2}x_1^2 + \frac{1}{4}x_2^4 + \frac{1}{2}x_2^2 - 0.9x_1x_2$$

$$s.t. \begin{cases} x_1 + x_2 \le 2 \\ -x_1 + x_2 \le 2 \\ x_1 - 3x_2 \le -2 \\ x_1, x_2 \ge 0 \end{cases} \tag{6}$$

This problem has an optimal solution $x^* = (0.427, 0.809)^T$. It can be seen that objective function is strictly convex and the feasible region is a convex set. Note that

$$\nabla f(x) = \begin{pmatrix} x_1^3 + x_1 - 0.9x_2 \\ x_2^3 + x_2 - 0.9x_1 \end{pmatrix}$$

and

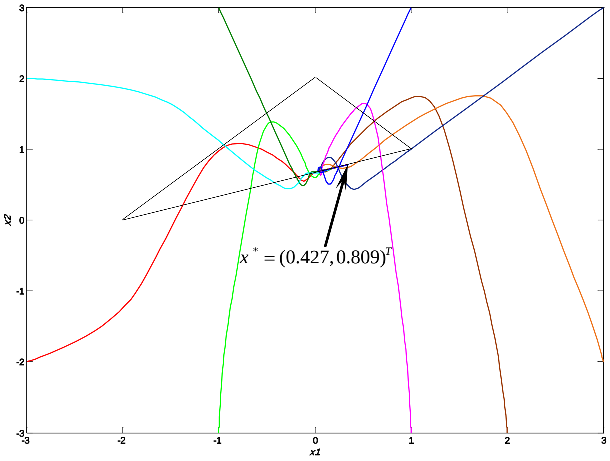

Fig.2: Transient behavior $(x_1, x_2)^T$ of proposed neural network (3) with various initial points in Example 1.

$$\nabla^2 f(x) = \begin{pmatrix} 3x_1^2 + 1 & -0.9 \\ -0.9 & 3x_2^2 + 1 \end{pmatrix}$$

is positive–definite on X. Theorem 2 guarantees that the neural network (3) is globally convergent to optimal solution of problem (6). We use the neural network (3) to solve this problem. Simulation results show the trajectories of (3) with any initial points are always convergent to x^*. The feasible region and the transient behavior $(x_1, x_2)^T$ based on (3) with various initial points are displayed in Fig. 2. Starting points for dual variables are random.

Example 2. Consider the following nonlinear programming problem:

$$\begin{aligned} \min\quad & \frac{3}{2}(x_1^2 + x_2^2) + 2(x_3^2 + x_4^2) - \ln(x_1 x_4) \\ & + 3x_1x_2 + 4x_3x_4 - 2x_1 - 3x_4 \end{aligned} \tag{7}$$

$$s.t. \begin{cases} x_1 + x_2 = 1 \\ x_3 + x_4 = 1 \\ 0.1 \le x_1 \le 10, \quad 0 \le x_2 \le 10, \\ \quad 0 \le x_3 \le 10,\ 0.1 \le x_4 \le 10 \end{cases}$$

The nonlinear programming problem (7) has the exact optimum solution $x^* = (1, 0, 0, 1)^T$ while its dual has the optimum solution $y^* = (0, 0)^T$. The model (5) is used to find optimal solutions x^* and y^*, simultaneously. The convergent path for the variables $(x, y)^T = (x_1, x_2, x_3, x_4, y_1 y_2)^T$ is shown in Fig. 3 with various random starting points. Simulation results show that the neural network (5) is globally convergent to x^* and y^*. The obtained solutions are absolutely exact and real time.

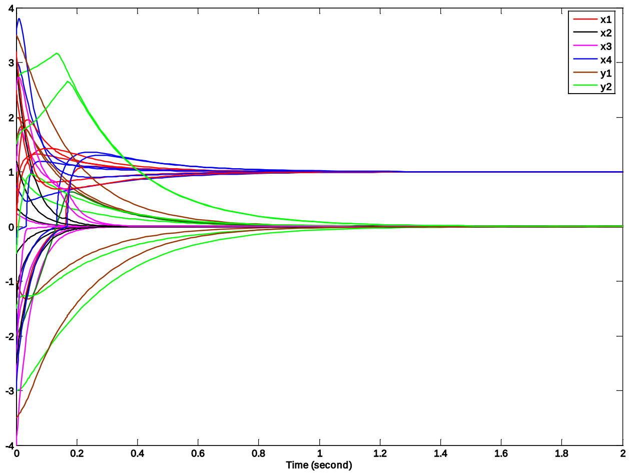

Fig. 3. The trajectories of proposed neural network (5) with various random starting points in Example 2.

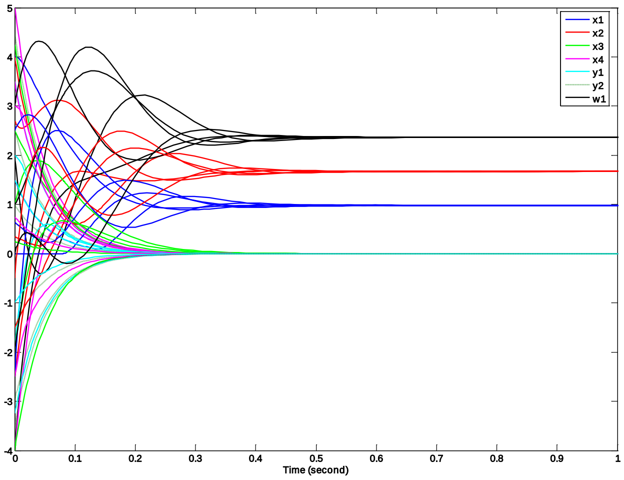

Fig. 4. The trajectories of proposed neural network (5) with various arbitrary initial points in Example 3.

Example 3. Consider the following nonlinear programming problem:

$$\min \quad 0.4x_1 + x_1^2 + x_2^2 - x_1x_2 + 0.5x_3^2 + 0.5x_4^2 + \frac{1}{30}x_1^3$$

$$s.t. \begin{cases} -x_1 + x_2 - x_3 \le 2, \\ 3x_1 + x_2 - x_3 - x_4 \le 18, \\ \frac{1}{3}x_1 + x_2 - x_4 = 2, \\ x_1, x_2, x_3, x_4 \ge 0 \end{cases} \tag{8}$$

This problem has an optimal solution $x^* = (0.982, 1.672, 0, 0)^T$. Fig. 4 show that all state variable trajectories of neural network (5) are globally convergent to the optimal solution $z^* = (x^*, y^*, w^*)^T = (0.982, 1.672, 0, 0, 0, 0, 2.363)^T$ in problem (8) and its dual, simultaneously. These trajectories indicate fast convergence to exact optimal solutions.

7. COMPARATIVE ANALYSIS

To see how well the present neural network model is, we compare it with one existing network model for solving nonlinear convex programming problems. Consider (4) is

$$\begin{aligned} &\min \quad f(x) \\ &s.t. \quad Ax = b \\ &\qquad l \le x \le h \end{aligned} \tag{9}$$

Then the neural network (5) is as follows

$$\begin{cases} \frac{dx}{dt} = P_X(x - \nabla f(x) + A^T y) - x \\ \frac{dy}{dt} = b - Ax \end{cases} \tag{10}$$

Neural network model for solving (9) was developed in [15]. Its dynamical equation is described by

$$\begin{cases} \frac{dx}{dt} = P_X(x - \nabla f(x) + A^T y) - x \\ \frac{dy}{dt} = -AP_X(x - \nabla f(x) + A^T y) + b \end{cases} \tag{11}$$

It has two layers and because of an additional nonlinear term, it is more complex in structure than the proposed neural network model (10). Therefore our network implementation is more economical, which is very important for implementing of the large-scale neural network.

8. CONCLUDING REMARKS

We have proposed a feedback neural network for solving nonlinear convex programming problems with hybrid constraints in real time using the projection technique. We have also given a complete proof of the stability and global convergence of the proposed network by definition of a suitable Liapunov function. Compared with the existing neural network for solving such problems, the proposed neural network has a simple single layer structure, without a penalty parameter and amenable to parallel implementation. The simulation results displayed the reasonableness of our theory and fast convergence of the proposed neural networks to exact optimal solutions of nonlinear optimization problems.

9. ACKNOWLEDGMENTS

The authors gratefully acknowledge the financial and other support of this research, provided by Islamic Azad University, Eslamshahr Branch, Tehran, Iran. Also the authors would like to thank M. Saderi-Oskoei for her valuable comments and discussions.

10. REFERENCES

[1] Bazaraa, M.S., Sherali, H.D., and Shetty, C.M. 1993. Nonlinear Programming Theory and Algorithms, 2nd ed. New York: Wiley.

[2] Kalouptisidis, N. 1997. Signal Processing Systems, Theory and Design. New York: Wiley.

[3] Avriel, M. 1976. Nonlinear Programming: Analysis and Methods. Englewood Cliffs, NJ: Prentice-Hall.

[4] Fletcher, R. 1981. Practical Methods of Optimization. New York: Wiley.

[5] Harker, P.T. and Pang, J.S. 1990. Finite-dimensional variational inequality and nonlinear complementarity problems: A survey of theory, algorithms, and applications, Mathematical Programming, 48, 161–220.

[6] He, B.S. and Liao, L.Z. 2002. Improvements of some projection methods for monotone nonlinear variational inequalities, Journal of Optimization Theory and Applications 112(1), 111–128.

[7] He, B.S. and Zhou, J. 2000. A modified alternating direction method for convex minimization problems, Applied Mathematics Letters 13(2), 123–130.

[8] Kennedy, M.P. and Chua, L.O. 1988. Neural networks for nonlinear programming, IEEE Transactions on Circuits and Systems 35(5), 554–562.

[9] Lillo, W.E., Loh, M.H., Hui, S. and Zak, S.H. 1993. On solving constrained optimization problems with neural networks: A penalty method approach, IEEE Transactions on Neural Networks 4(6), 931–940.

[10] Rodríguez-Vázquez, A., Domínguez-Castro, R., Rueda, A., Huertas, J.L. and Sánchez-Sinencio, E. 1990. Nonlinear switched-capacitor 'neural networks' for optimization problems, IEEE Transactions on Circuits and Systems 37(3), 384–397.

[11] Ghasabi-Oskoei, H. 2005. Numerical solutions for constrained quadratic problems using high-performance neural networks, Applied Mathematics and Computation 169(1), 451–471.

[12] Ghasabi-Oskoei, H. and Mahdavi-Amiri, N. 2006. An efficient simplified neural network for solving linear and quadratic programming problems, Applied Mathematics and Computation 175(1), 452–464.

[13] Ghasabi-Oskoei H. 2007. Novel artificial neural network with simulation aspects for solving linear and quadratic programming problems, Computers and Mathematics with Applications 53, 1439–1454.

[14] Leung, Y., Chen, K. and Gao, X. 2003. A high-performance feedback neural network for solving convex nonlinear programming problems, IEEE Transactions on Neural Networks 14(6), 1469–1477.

[15] Tao, Q., Cao, J.D., Xue, M.S. and Qiao, H. 2001. A high performance neural network for solving nonlinear programming problems with hybrid constraints, Phys. Lett. A, 288(2), 88–94.

[16] Xia, Y.S. 1996. A new neural network for solving linear and quadratic programming problems, IEEE Transactions on Neural Networks 7(6), 1544–1547.

[17] Kinderlerer, D. and Stampcchia, G. 1980. An Introduction to Variational Inequalities and Their Applications, Academic Press, New York.

[18] Zhang, X., Li, X. and Chen, Z. 1982. The Theory of Ordinary Differential Equations in Optimal Control Theory, Advanced Educational Press, Beijing, in Chinese.

[19] Luenberger, D.G. 1989. Introduction to Linear and Nonlinear Programming, Addison-Wesley Reading, MA, Chapter 12.

[20] Bertsekas D.P. and Tsitsiklis, J. N. 1989. Parallel and Distributed Computation: Numerical Methods. Englewood Cliffs, NJ: Prentice-Hall.

[21] Robinson, J. 2004. An Introduction to Ordinary Differential Equations, Cambridge University Press.