# A Discontinuous Recurrent Neural Network with Predefined Time Convergence for Solution of Linear Programming

Juan Diego Sánchez-Torres*, Edgar N. Sanchez* and Alexander G. Loukianov*
*Automatic Control Laboratory, CINVESTAV-IPN Guadalajara, Av. del Bosque 1145, CP 45019, México.
email: dsanchez@gdl.cinvestav.mx, sanchez@gdl.cinvestav.mx, louk@gdl.cinvestav.mx.

***Abstract*—The aim of this paper is to introduce a new recurrent neural network to solve linear programming. The main characteristic of the proposed scheme is its design based on the predefined-time stability. The predefined-time stability is a stronger form of finite-time stability which allows the *a priori* definition of a convergence time that does not depend on the network initial state. The network structure is based on the Karush-Kuhn-Tucker (KKT) conditions and the KKT multipliers are proposed as sliding mode control inputs. This selection yields to an one-layer recurrent neural network in which the only parameter to be tuned is the desired convergence time. With this features, the network can be easily scaled from a small to a higher dimension problem. The simulation of a simple example shows the feasibility of the current approach.**

## I. Introduction

Optimization methods have been widely applied in science and engineering. The optimization goal is to determine the decision variables values, which maximize or minimize an objective function, sometimes, subject to constraints. Some of this problems are large-scale real-time linear programming procedures. For such applications, sequential algorithms as the classical simplex or the interior point methods are often proposed. However, those traditional approaches may not be efficient since the computing time required for a solution is greatly dependent on the problem dimension and structure.

The use of dynamical systems which can solve real-time optimization was introduced in [1] and arises as a promising alternative. Extensions and new approaches were presented for linear programming [2]–[4] and for nonlinear programming [5]. For most of the cases, these systems are presented as the solution to a controller design problem [6], in the form of circuits [7], [8] or under the computational paradigm of the artificial neural networks (ANN) where are of the form of recurrent neural networks (RNN) [9]–[11]. Due to its inherent massive parallelism, those systems are able to solve optimization problems in running time at the orders of magnitude much faster than those of the most popular optimization algorithms executed on general-purpose digital computers [12], with unusual flexibility because the system constantly seeks new solutions as the parameters of the problem are varied [1]. Usually, the network structure is proposed based on the Karush-Kuhn-Tucker (KKT) optimality conditions [13], [14], by using the KKT multipliers as activation functions.

A major contribution to this class of solutions is the use of systems with motion on a sliding manifold, as proposed in [15], that is an integral manifold with finite reaching time [16], presented by some non-smooth systems, providing finite time convergence to the problem solution. Usually, those sliding modes appears when discontinuous activation functions are used for RNN design exhibiting features such as finite time convergence and insensitivity to external bounded disturbances [17]. For this case, the sliding modes and finite time convergence to sets defined by the optimization constraints are desirable characteristics of the neural network [18]. Taking advantage of these features presented by the discontinuous systems, the RNN design has been stated as sliding mode control problem [19] and several recurrent neural networks have been proposed using different discontinuous activation functions as hard-limiting [20]–[22], Heaviside [23] and dead-zone [24], [25]. Further results on networks with these dynamical properties were presented in [26]–[29], where the analysis is based on the theory of differential inclusions and differential equations with discontinuous right-hand [30]–[32]. In addition, a class of RNN with fixed time convergence have been recently proposed [33], providing convergence in a finite time that does not depends on the network initial condition [34], [35].

Although the mentioned works exhibit high performance, it is necessary to tune the network parameters such that the optimizer trajectories converge to the optimization solution. For most of the cases, the number of network parameters increases linearly with the optimization problem dimension, since for every decision variable there is an individual selection of each activation function. In addition, the fixed time property is not presented in most of the mentioned references. This last desirable property allows the design of finite time convergent systems with this time independent to initial conditions. The reference [33] provides a fixed time design, however there is not presented a straight method to select that time.

In this paper is proposed a RNN for the solution of linear programming. Its design is considered as a sliding mode control problem, where the network structure is based on the Karush-Kuhn-Tucker (KKT) optimality conditions [13], [14] and the KKT multipliers are regarded as control inputs. At this point, to the best of the authors knowledge, a controller with vector structure and predefined time stability is firstly proposed. The predefined time stability refers to a particular case of the fixed time stability where the convergence time can be selected *a priori*. This structure allows the problem to be solved without the individual selection of each stabilizing input, instead a multivariable function, based on the unit control [36], [37], is used. On the other hand, the predefined time stability ensures the possibility to select a time independent to the initial conditions in which the system converges. This controller is used to the KKT multiplier design, enforcing a sliding mode in which the optimization problem is solved.

Thus, the proposed approach have very attractive features as: predefined time of convergence to the optimization problem solution and only a tuning parameter, namely the desired convergence time, regardless of the optimization problem dimension. Therefore, it offers the scalability characteristic, that allows the on-line solution of problems with low and higher dimension without major changes of the system.

In the following, Section II presents the mathematical preliminaries and some useful definitions. Section III describes the proposed system for the solution of linear programming, including the stability analysis and an academic example which illustrates the fixed time convergence feature of the system. A simulation example id presented in Section IV. Finally, in Section V the conclusions are presented.

## II. Mathematical Preliminaries

Consider the system

$$\dot{\xi} = f(t, \xi) \tag{1}$$

where $\xi \in \mathbb{R}^n$ and $f : \mathbb{R}_+ \times \mathbb{R}^n \to \mathbb{R}^n$. For this system, its initial conditions or initial state are $\xi(t_0)$ where $t_0 \in \mathbb{R}_+$. The time variable $t$ is defined on the interval $[t_0, \infty)$.

*Definition 2.1 (Integral manifolds):* Let $\sigma(\xi)$ a smooth function $\mathbb{R}^n \mapsto \mathbb{R}^k$ with $k \leq n$, and let the manifold

$$\mathcal{M} = \{\xi \in \mathbb{R}^n : \sigma(\xi) = 0\}$$

If for an initial condition $\xi(t_0) \in \mathcal{M}$, the trajectory $\xi(t, \xi_0) \in \mathcal{M}$ for all $t$, the manifold $\mathcal{M}$ is called an integral manifold.

*Definition 2.2 (Sliding mode manifold [16]):* If there is a non-empty set

$$\mathcal{N} \subset \mathbb{R}^n \setminus \mathcal{M}$$

such that for every initial condition $\xi(t_0) \in \mathcal{N}$ is a time $t_0 < t_s < \infty$ in which the system reaches the manifold $\mathcal{M}$, then the manifold $\mathcal{M}$ is called an sliding mode manifold.

*Remark 2.1:* A sliding mode on a certain sliding manifold can only appears if $f$ is a non-smooth (usually discontinuous) function. For this case, the solutions of (1) are understood in Filippov sense [30].

Now, consider the system (1) forced by an affine control input

$$\dot{\xi} = f(t, \xi) + B(\xi)u \tag{2}$$

where $u \in \mathbb{R}^k$ and $B(\xi)$ is a full rank matrix for every $\xi$. For this case it is assumed that $\sigma(\xi)$ has relative degree one, that is

$$\dot{\sigma} = a(t, \xi) + b(\xi)u$$

where $a(t, \xi) = \frac{\partial \sigma}{\partial \xi} \cdot f(t, \xi)$ and $b(\xi) = \frac{\partial \sigma}{\partial \xi} \cdot B(\xi)$. In addition, in contrast to (1), $f(t, \xi)$ is continuous, so it is $a(\xi)$, and the sliding mode on $\mathcal{M}$ is induced by means of a non-smooth $u$. This is exposed in the following definition:

*Definition 2.3 (Sliding mode control [16], [37]):* If for the system (2) $u$ is such that the system evolves on the manifold $\mathcal{M}$ after a time $t_f$, $u$ is a sliding mode control input.

Often, the sliding mode control input is discontinuous as follows:

$$u(\xi) = \begin{cases} u^+(\xi) & \text{if } \sigma(\xi) > 0 \\ u^-(\xi) & \text{if } \sigma(\xi) < 0 \end{cases}$$

where $u^+(\xi)$ and $u^-(\xi)$ are continuous functions such that $u^+(\xi) \neq u^-(\xi)$.

Defining $f^+ = f(t, \xi) + B(\xi)u^+(\xi)$ and $f^- = f(t, \xi) + B(\xi)u^-(\xi)$, the Filippov solution of (2) on the manifold $\mathcal{M}$ is

$$\dot{\xi} = \mu f^+ + (1 - \mu) f^-$$

where $0 \leq \mu \leq 1$ is solution to $\dot{\sigma}(\xi) = 0$.

Similarly to the Filippov solutions on a sliding manifold $\mathcal{M}$, the concept of equivalent control is given in the following definition:

*Definition 2.4 (Equivalent control [37]):* Let the system (2) evolving on the sliding manifold $\mathcal{M}$. The equivalent control $u_{\text{eq}}$ is the continuous solution to $\dot{\sigma}(\xi) = 0$ which results from $a(t, \xi) + b(\xi)u_{\text{eq}} = 0$.

With the definition of the equivalent control, the motion of the system (2) on the sliding manifold $\mathcal{M}$ is given by

$$\dot{\xi} = f(t, \xi) + B(\xi)u_{\text{eq}}.$$

*Remark 2.2:* Note that for the affine control input case, the Filippov solution results to the same motion on the manifold $\mathcal{M}$ than that obtained with the equivalent control method [37].

The idea of the sliding mode control is highly related with the finite- time stability. This time however often depends on the system initial conditions. The case when convergence time presents a class of uniformity with respect to the initial conditions not depending on the initial conditions is known as fixed time stability [34]. The following definition presents a precise statement of the fixed time stability:

*Definition 2.5 (Globally fixed-time attraction [35]):* Let a non-empty set $M \subset \mathbb{R}^n$. It is said to be globally fixed-time attractive for the system (1) if any solution $\xi(t, \xi_0)$ of (1) reaches $M$ in some finite time moment $t = T(\xi_0)$ and the settling-time function $T(\xi_0) : \mathbb{R}^n \to \mathbb{R}_+ \cup \{0\}$ is bounded by some positive number $T_{\text{max}}$, i.e. $T(\xi_0) \leq T_{\text{max}}$ for $\xi_0 \in \mathbb{R}^n$.

Note that for some systems $T_{\text{max}}$ can be tuned by a particular selection of the system parameters, this notion refers to the predefined stability which is given in [38] and presented in the following definition:

*Definition 2.6 (Predefined-time attraction):* For the case of fixed-time attraction when the system parameters can be selected such that the time $T_{\text{max}}$ can be predefined as desired, it is said that $M$ is predefined-time attractive.

## III. A RNN for Linear Programming Problem

Before to present the predefined-time RNN, it will be exposed novel control structures to be used in the optimizing system design.

### A. Preliminary Results

With the definition of a predefined-time attractive set, the following lemma provides a Lyapunov characterization of a class of these sets on the state space:

*Lemma 3.1 (Lyapunov function):* If there exists a continuous radially unbounded function

$$V : \mathbb{R}^n \to \mathbb{R}_+ \cup \{0\}$$

such that $V(\xi) = 0$ for $\xi \in M$ and any solution $\xi(t)$ satisfies

$$\dot{V} \le -\alpha \exp(V(\xi(t))) \tag{3}$$

for $\alpha > 0$, then the set $M$ is globally predefined-time attractive for the system (1) and $T_{\max} = \frac{1}{\alpha} + t_0$.

*Proof* The solution of (3) is

$$V(t) = \ln\left(\frac{1}{\alpha(t-t_0)+\exp(-V_0)}\right)$$

where $V_0 = V(\xi_0)$ and $\xi_0 = \xi(t_0)$. Note that $V(t) = 0$ if $\frac{1}{\alpha(t-t_0)+\exp(-V_0)} = 1$, hence the settling-time function is

$$T(\xi_0) = \frac{1-\exp(-V_0)}{\alpha} + t_0.$$

From $0 < \exp(-V_0) < 1$, it follows that $T_{\max} = \frac{1}{\alpha} + t_0$. ■

In order to apply the previous result to control design, consider the dynamic system

$$\dot{\xi} = \Delta(\xi, t) + u \tag{4}$$

with $\xi, u \in \mathbb{R}^n$ and $\Delta : \mathbb{R}_+ \times \mathbb{R}^n \to \mathbb{R}^n$. The main objective is to drive the system (4) to the point $\xi = 0$ in a predefined fixed time in spite of the unknown non-vanishing disturbance $\Delta(\xi, t)$. A solution to this problem which does not requires an individual selection of each of the $n$ control variables based on the *unit control* is presented in the following theorem:

*Theorem 3.1 (Predefined-time multivariable control):* Let the function $\phi(\xi, t)$ to be bounded as $\|\Delta(\xi, t)\| \le \delta$, with $0 < \delta < \infty$ a known constant. Then, by selecting the control input

$$u = -\left(\frac{1}{T_c} + \delta\right)\frac{\xi}{\|\xi\|}\exp(\|\xi\|)$$

with $T_c$ being a scalars the system (4) is globally predefined-time stable with settling-time $T_c$.

*Proof*: Let the Lyapunov function $V = \|\xi\|$, its derivative is given by $\dot{V} = \frac{\xi^T}{\|\xi\|}\dot{\xi}$. Therefore

$$\begin{aligned}
\dot{V} &= \frac{\xi^T}{\|\xi\|}\left[\Delta(\xi,t) - \left(\frac{1}{T_c}+\delta\right)\frac{\xi}{\|\xi\|}\exp(\|\xi\|)\right] \\
&\le \frac{\xi^T}{\|\xi\|}\left[\delta - \left(\frac{1}{T_c}+\delta\right)\frac{\xi}{\|\xi\|}\exp(\|\xi\|)\right] \\
&\le -\frac{1}{T_c}\exp(\|\xi\|)
\end{aligned} \tag{5}$$

that, by replacing the Lyapunov function reduces to $\dot{V} \le -\frac{1}{T_c}\exp(V)$ . Finally, by direct application of *Lemma 3.1*, the proof is finished. ■

### B. Linear Programming Problem Statement

Let the following linear programming problem:

$$\begin{cases} \min_x & \mathbf{c}^T x \\ \text{s.t.} & \mathbf{A}x = \mathbf{b} \\ & l \le x \le h \end{cases} \tag{6}$$

where $x = \begin{bmatrix} x_1 & \dots & x_n \end{bmatrix}^T \in \mathbb{R}^n$ are the decision variables, $\mathbf{c} \in \mathbb{R}^n$ is a cost vector, $\mathbf{A}$ is an $m \times n$ matrix such that $\text{rank}(\mathbf{A}) = m$ and $m \le n$; $\mathbf{b}$ is a vector in $\mathbb{R}^m$ and, $l = \begin{bmatrix} l_1 & \dots & l_n \end{bmatrix}$, $h = \begin{bmatrix} h_1 & \dots & h_n \end{bmatrix} \in \mathbb{R}^n$.

Let $y = \begin{bmatrix} y_1 & \dots & y_m \end{bmatrix}^T \in \mathbb{R}^m$ and $z = \begin{bmatrix} z_1 & \dots & z_n \end{bmatrix}^T \in \mathbb{R}^n$. Hence, the Lagrangian of (6) is

$$L(x, y, z) = \mathbf{c}^T x + z^T x + y^T(\mathbf{A}x - \mathbf{b}). \tag{7}$$

The KKT conditions establish that $x^*$ is a solution for (6) if and only if $x^*$, $y$ and $z$ in (6)-(7) are such that

$$\nabla_x L(x^*, y, z) = \mathbf{c} + z + \mathbf{A}^T y = 0 \tag{8}$$

$$\mathbf{A}x^* - \mathbf{b} = 0 \tag{9}$$

$$z_i x_i^* = 0 \text{ if } l_i < x_i^* < h_i,\ \forall i = 1, \dots, n. \tag{10}$$

### C. RNN Design with Predefined-Time Convergence

Following the KKT approach, a recurrent neural network which solves the problem (6) in finite time is proposed. For this purpose, let $\Omega_e = \{x \in \mathbb{R}^n : \mathbf{A}x - \mathbf{b} = 0\}$ and $\Omega_d = \{x \in \mathbb{R}^n : l \le x \le h\}$ . According to (6), $x^* \in \Omega$ where $\Omega = \Omega_d \cap \Omega_e$.

From (8), let

$$\dot{x} = -\mathbf{c} + \mathbf{A}^T y + z, \tag{11}$$

then, $y$ and $z$ must be designed such that $\Omega$ is an attractive set, fulfilling conditions (8)-(10). For this case, in addition to condition (10), $z$ is considered such that

$$\begin{cases} z_i \ge 0 & \text{if } x_i \ge h_i \\ z_i \le 0 & \text{if } x_i \le l_i \end{cases}, \tag{12}$$

and the variable $\sigma \in \mathbb{R}^m$ is defined as

$$\sigma = \mathbf{A}x - \mathbf{b}. \tag{13}$$

In order to obtain predefined-time stability to the solution $x^*$, the terms $y$ and $z$ are proposed in (11) as

$$y = (\mathbf{A}\mathbf{A}^T)^{-1}\left[\mathbf{A}\mathbf{c} - \mathbf{A}z + \frac{1}{T_s}\phi(\sigma)\right] \tag{14}$$

and

$$z = \left(\|\mathbf{c}\| + \frac{1}{T_s}\right)\varphi(x, l, h) \tag{15}$$

respectively, where $T_s > 0$.

For this case, the multivariable activation functions are $\varphi(x, l, h) = \begin{bmatrix} \varphi_1(x, l_1, h_1) & \dots & \varphi_n(x, l_n, h_n) \end{bmatrix}^T$, with $\varphi_i(x, l_i, h_i)$ of the form

$$\varphi_i(x, l_i, h_i) = \begin{cases} -\frac{x_i - l_i}{\|x - l\|}\exp(\|x - l\|) & \text{if } x_i \le l_i \\ 0 & \text{if } l_i < x_i < h_i \\ -\frac{x_i - h_i}{\|x - h\|}\exp(\|x - h\|) & \text{if } x_i \ge h_i \end{cases} \tag{16}$$

and

$$\phi(\sigma) = -\frac{\sigma}{\|\sigma\|}\exp(\|\sigma\|). \tag{17}$$

Therefore, with the structure given by (11) and the KKT multiplier as in (14) and (15), with activation functions (16) and (17), the following Theorem presents a RNN which solves (6) in predefined-time.

*Theorem 3.2 (Predefined-time RNN for linear programming):* For the RNN

$$\dot{x} = -\mathbf{c}\Lambda + \left(\|\mathbf{c}\| + \frac{1}{T_s}\right)\Lambda\varphi(x,l,h) + \frac{1}{T_s}\mathbf{A}^{+}\phi(\sigma) \tag{18}$$

where $\Lambda = I - \mathbf{A}^T(\mathbf{A}\mathbf{A}^T)^{-1}\mathbf{A}$, $\mathbf{A}^{+} = \mathbf{A}^T(\mathbf{A}\mathbf{A}^T)^{-1}$ and $T_s > 0$, the point $x^*$ is globally predefined-time stable with settling-time $T_s$.

*Proof*: The dynamics of (13) is given by

$$\dot{\sigma} = \mathbf{A}\left(-\mathbf{c} + \mathbf{A}^T y + z\right). \tag{19}$$

Therefore, with the selection of $y$ as in (14), the system (19) reduces to

$$\dot{\sigma} = -\frac{1}{T_s}\frac{\sigma}{\|\sigma\|}\exp(\|\sigma\|).$$

Thus, from *Theorem 3.1*, a sliding mode is induced on the manifold $\sigma = 0$. Therefore, the set $\Omega_e$ is predefined-time attractive with settling-time $T_s$.

On the manifold $\sigma = 0$, the equivalent value of $\phi$ is the solution of $\dot{\sigma} = 0$. Resulting to $\phi_{\text{eq}} = 0$ or $y_{\text{eq}} = (\mathbf{A}\mathbf{A}^T)^{-1}[\mathbf{A}\mathbf{c} - \mathbf{A}z]$. Therefore, the dynamics of (11) on that manifold is

$$\dot{x} = -\mathbf{c}\Lambda + \Lambda z. \tag{20}$$

With the selection of $z$ as in (15), the system (20) results to

$$\dot{x} = -\mathbf{c}\Lambda + \Lambda\left(\|\mathbf{c}\| + \frac{1}{T_s}\right)\varphi(x,l,h).$$

Consider the Lyapunov function $V = \|x\|$. Its derivative is given by $\dot{V} = \frac{x^T}{\|x\|}\dot{x}$. Therefore

$$\begin{aligned}\dot{V} &= \frac{x^T}{\|x\|}\left[-\mathbf{c}\Lambda + \Lambda\left(\|\mathbf{c}\| + \frac{1}{T_s}\right)\varphi(x,l,h)\right]\\ &\leq \frac{x^T}{\|x\|}\left[\frac{1}{T_s}\varphi(x,l,h)\right].\end{aligned} \tag{21}$$

Replacing the Lyapunov function

$$\dot{V}\begin{cases}\leq -\frac{1}{T_s}\exp(V) \text{ if } x < l \text{ or } x > h\\ = 0 \text{ if } l \leq x \leq h\end{cases} \tag{22}$$

From *Theorem 3.1*, the set $\Omega_d$ is predefined-time attractive with settling-time $T_s$.

In the set $\Omega_d$ the equivalent value of $\varphi$, $\varphi_{\text{eq}}$, is the solution to $\dot{x} = 0$. With the application of *Theorem 3.1*, the conditions (9) and (10) are satisfied, providing predefined-time convergence to the set $\Omega$. Now, by using the equivalent control method, the solution of $\dot{x} = 0$ and $\dot{\sigma} = 0$ in (11) for $t > T_s$ has the form

$$\mathbf{c} + \mathbf{A}^T y_{\text{eq}} + z_{\text{eq}} = 0.$$

Therefore, the condition (8) is fulfilled, implying the point $x^* \in \Omega$ is globally fixed-time stable. ■

*Remark 3.1:* Note that, in contrast to the most of the RNN presented in the literature, this scheme only needs the tuning of one variable, namely $T_s$ in spite of the problem dimensions.

## IV. Application Example

Let the following linear programming problem [24]:

$$\begin{cases}\min_x & 4x_1 + x_2 + 2x_3\\ \text{s.t} & x_1 - 2x_2 + x_3 = 2\\ & -x_1 + 2x_2 + x_3 = 1\\ & -5 \leq x_1, x_2, x_3 \leq 5\end{cases}. \tag{23}$$

The proposed neural network (18), with the parameter $T_s = 0.1$, gives the results shown in Fig. 1.

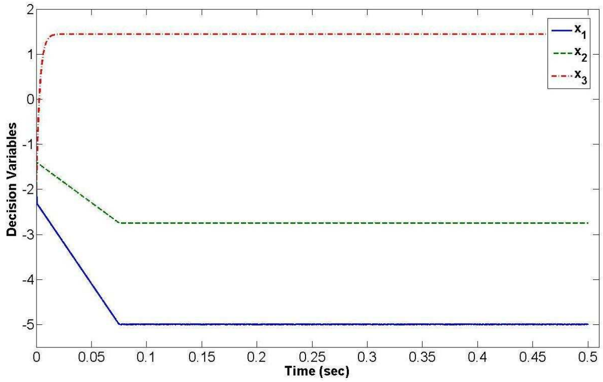

Fig. 1. Transient behavior of the $x$ variables.

Here, it can be observed that the network converges to the optimal solution $x^* = [-5, -2.75, 1.5]$.

## V. Conclusions

In this paper a novel optimization algorithm is proposed. It can solve linear programming problems in a predefined time. The convergence an optimality proofs were presented. In addition, in order to illustrate the method, a simulation example was given. As future work, the optimization algorithm proposed will be extend to solve quadratic programming problems, larger dimension problems and to other structures which provide predefined time convergence.

## Acknowledgment

This work was supported by the National Council of Science and Technology (CONACYT), Mexico, under Grant 129591.

## References

[1] I. B. Pyne, “Linear programming on an electronic analogue computer,” *American Institute of Electrical Engineers, Part I: Communication and Electronics, Transactions of the*, vol. 75, no. 2, pp. 139–143, 1956.

[2] L. Chua and G.-N. Lin, “Nonlinear programming without computation,” *IEEE Transactions on Circuits and Systems*, vol. 31, no. 2, pp. 182–188, 1984.

[3] D. Tank and J. Hopfield, “Simple ’neural’ optimization networks: An A/D converter, signal decision circuit, and a linear programming circuit,” *IEEE Transactions on Circuits and Systems*, vol. 33, no. 5, pp. 533–541, 1986.

[4] R. W. Brockett, “Dynamical systems that sort lists, diagonalize matrices and solve linear programming problems,” in *Proc. 27th IEEE Conf. Decision and Control*, 1988, pp. 799–803.

[5] M. P. Kennedy and L. O. Chua, “Neural networks for nonlinear programming,” *IEEE Transactions on Circuits and Systems*, vol. 35, no. 5, pp. 554–562, 1988.

[6] F. A. Pazos and A. Bhaya, “Control liapunov function design of neural networks that solve convex optimization and variational inequality problems,” *Neurocomputing*, vol. 72, no. 1618, pp. 3863 – 3872, 2009.

[7] G. Wilson, “Quadratic programming analogs,” *IEEE Transactions on Circuits and Systems*, vol. 33, no. 9, pp. 907–911, 1986.

[8] A. Rodriguez-Vazquez, R. Dominguez-Castro, A. Rueda, J. L. Huertas, and E. Sanchez-Sinencio, “Nonlinear switched capacitor ‘neural’ networks for optimization problems,” *IEEE Transactions on Circuits and Systems*, vol. 37, no. 3, pp. 384–398, 1990.

[9] J. J. Hopfield, “Neural networks and physical systems with emergent collective computational abilities,” *Proceedings of the National Academy of Sciences*, vol. 79, no. 8, pp. 2554–2558, 1982.

[10] J. Wang, “Analysis and design of a recurrent neural network for linear programming,” *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, vol. 40, no. 9, pp. 613–618, 1993.

[11] Y. Xia and J. Wang, “A general methodology for designing globally convergent optimization neural networks,” *IEEE Transactions on Neural Networks*, vol. 9, no. 6, pp. 1331–1343, 1998.

[12] A. Cichocki and R. Unbehauen, *Neural networks for optimization and signal processing*. J. Wiley, 1993.

[13] W. Karush, “Minima of functions of several variables with inequalities as side constraints,” Master’s thesis, Dept. of Mathematics, Univ. of Chicago, Chicago, Illinois., 1939.

[14] H. W. Kuhn and A. W. Tucker, “Nonlinear programming,” in *Proc. Second Berkeley Symp. on Math. Statist. and Prob. (Univ. of Calif. Press)*, 1951.

[15] S. K. Korovin and V. I. Utkin, “Using sliding modes in static optimization and nonlinear programming,” *Automatica*, vol. 10, no. 5, pp. 525 – 532, 1974.

[16] S. V. Drakunov and V. Utkin, “Sliding mode control in dynamic systems,” *International Journal of Control*, vol. 55, pp. 1029–1037, 1992.

[17] V. Utkin, J. Guldner, and J. Shi, *Sliding Mode Control in Electro-Mechanical Systems, Second Edition (Automation and Control Engineering)*, 2nd ed. CRC Press, 5 2009.

[18] E. K. P. Chong, S. Hui, and S. H. Zak, “An analysis of a class of neural networks for solving linear programming problems,” *IEEE Transactions on Automatic Control*, vol. 44, no. 11, pp. 1995–2006, 1999.

[19] M. Glazos, S. Hui, and S. Zak, “Sliding modes in solving convex programming problems,” *SIAM Journal on Control and Optimization*, vol. 36, no. 2, pp. 680–697, 1998.

[20] Q. Liu and J. Wang, “A one-layer recurrent neural network with a discontinuous hard-limiting activation function for quadratic programming,” *IEEE Transactions on Neural Networks*, vol. 19, no. 4, pp. 558–570, 2008.

[21] ——, “Finite-time convergent recurrent neural network with a hard-limiting activation function for constrained optimization with piecewise-linear objective functions,” *IEEE Transactions on Neural Networks*, vol. 22, no. 4, pp. 601–613, 2011.

[22] Z. Guo, Q. Liu, and J. Wang, “A one-layer recurrent neural network for pseudoconvex optimization subject to linear equality constraints,” *IEEE Transactions on Neural Networks*, vol. 22, no. 12, pp. 1892–1900, 2011.

[23] Q. Liu and J. Wang, “A one-layer dual recurrent neural network with a heaviside step activation function for linear programming with its linear assignment application,” in *Artificial Neural Networks and Machine Learning  ICANN 2011*, ser. Lecture Notes in Computer Science, T. Honkela, W. Duch, M. Girolami, and S. Kaski, Eds. Springer Berlin Heidelberg, 2011, vol. 6792, pp. 253–260.

[24] ——, “A one-layer recurrent neural network with a discontinuous activation function for linear programming,” *Neural Computation*, vol. 20, no. 5, pp. 1366–1383, Nov. 2007.

[25] Q. Liu, Z. Guo, and J. Wang, “A one-layer recurrent neural network for constrained pseudoconvex optimization and its application for dynamic portfolio optimization,” *Neural Networks*, vol. 26, pp. 99 – 109, 2012.

[26] L. V. Ferreira, E. Kaszkurewicz, and A. Bhaya, “Convergence analysis of neural networks that solve linear programming problems,” in *Neural Networks, 2002. IJCNN ’02. Proceedings of the 2002 International Joint Conference on*, vol. 3, 2002, pp. 2476–2481.

[27] M. Forti, P. Nistri, and M. Quincampoix, “Generalized neural network for nonsmooth nonlinear programming problems,” *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 51, no. 9, pp. 1741–1754, 2004.

[28] L. Cheng, Z.-G. Hou, N. Homma, M. Tan, and M. M. Gupta, “Solving convex optimization problems using recurrent neural networks in finite time,” in *Proc. Int. Joint Conf. Neural Networks IJCNN 2009*, 2009, pp. 538–543.

[29] Q. Liu, J. Cao, and G. Chen, “A novel recurrent neural network with finite-time convergence for linear programming,” *Neural Computation*, vol. 22, no. 11, pp. 2962–2978, Aug. 2010.

[30] A. F. Filippov, *Differential equations with discontinuous righthand sides*, Mathematics and . its Applications (Soviet Series), Eds. Kluwer Academic Publishers Group, Dordrecht, 1988.

[31] F. Clarke, *Optimization and Nonsmooth Analysis*, ser. Classics in applied mathematics. Society for Industrial and Applied Mathematics (SIAM), 1990.

[32] J. Aubin and A. Cellina, *Differential Inclusions: Set-Valued Maps and Viability Theory*, ser. Grundlehren Der Mathematischen Wissenschaften. Springer London, 2012.

[33] J. D. Sanchez-Torres, E. N. Sanchez, and A. G. Loukianov, “Recurrent neural networks with fixed time convergence for linear and quadratic programming,” in *Neural Networks (IJCNN), The 2013 International Joint Conference on*, 2013, pp. 1–5.

[34] E. Cruz-Zavala, J. Moreno, and L. Fridman, “Uniform second-order sliding mode observer for mechanical systems,” in *Variable Structure Systems (VSS), 2010 11th International Workshop on*, june 2010, pp. 14 –19.

[35] A. Polyakov, “Nonlinear feedback design for fixed-time stabilization of linear control systems,” *IEEE Transactions on Automatic Control*, vol. 57, no. 8, pp. 2106–2110, 2012.

[36] C. M. Dorling and A. S. I. Zinober, “Two approaches to hyperplane design in multivariable variable structure control systems,” *International Journal of Control*, vol. 44, no. 1, pp. 65–82, 1986.

[37] V. Utkin, *Sliding Modes in Control and Optimization*. Springer Verlag, 1992.

[38] L. Fraguela, M. Angulo, J. Moreno, and L. Fridman, “Design of a prescribed convergence time uniform robust exact observer in the presence of measurement noise,” in *Decision and Control (CDC), 2012 IEEE 51st Annual Conference on*, Dec 2012, pp. 6615–6620.