Analysis and Design of a Recurrent Neural Network for Linear Programming

Jun Wang

***Abstract*—Linear programming is an important tool for system optimization and modelling. This paper presents a recurrent neural network with a time-varying threshold vector for solving linear programming problems. The proposed recurrent neural network is proven to be asymptotically stable in the large and capable of generating optimal solutions to linear programming problems. An op-amp based analog circuit design for realizing the recurrent neural network is described. The asymptotic properties of the proposed recurrent neural network for linear programming are analyzed. A detailed example is also presented to demonstrate the performance and operating characteristics of the recurrent neural network.**

I. Introduction

A linear programming problem seeks to optimize a linear objective function subject to a set of linear functional constraints and nonnegativity constraints. Many real-time systems, such as machine vision in robotic operations, and large-scale systems, such as personnel and equipment maneuvers in military operations, require solving large-scale linear programming problems in real time. In such applications, existing sequential algorithms such as the classical simplex method are usually not efficient due to the limitation of sequential processing.

In recent years, neural networks have been proposed for solving various optimization problems. A neural network consists of a number of massively connected simple neurons that operate concurrently in a parallel distributed fashion. While the mainstream of the neural network approach to optimization has focused on combinatorial optimization problems, a number of neural network paradigms have been proposed for solving linear programming problems. For example, Tank and Hopfield [1] demonstrate the potential of the Hopfield network for solving linear programming problems via an example. Kennedy and Chua [2], [3] analyze the Tank–Hopfield linear programming circuit and the Chua–Lin nonlinear programming circuit from a circuit-theoretic viewpoint and reconcile a modified canonical nonlinear programming circuit that can also be applied for linear programming. Rodríguez-Vázquez et al. [4] develop switched-capacitor neural networks for linear and nonlinear programming. Maa and Shanblatt [5] analyze the properties of the recurrent neural networks for linear and quadratic programming. The results of these investigations have demonstrated great potential and shed much light on the neural network research. However, none of the paradigms can be theoretically guaranteed to generate optimal solutions to linear programs.

Recently, Wang [6] proposed a class of recurrent neural networks for optimization. This particular class of recurrent neural networks has been proven to be able to generate optimal solutions to linear programs [7]. In this class of recurrent neural networks, the interconnection weights are essentially time-varying. Although a block diagram of the proposed neural network is presented [7] in which two dummy neurons are defined to circumvent the time-varying connection weights, the complexity of the neural network configuration still makes circuit realization challenging.

The primary objective of this paper is to present an alternative recurrent neural network for solving linear programs. Possessing the same optimal properties of its predecessor, the new neural network is much simpler in configuration and hence much easier to implement in electronic circuits. We will start with a description of the proposed recurrent neural network and a designed analog circuit for realizing the recurrent neural network, then analyze its asymptotic properties and discuss the simulation results.

Manuscript received March 6, 1992; revised May 13, 1993. An earlier version of this paper was presented at the 1992 IEEE International Symposium on Circuits and Systems. This paper was recommended by Associate Editor G. E. Ford.

The author is with the Department of Industrial Technology, University of North Dakota, Grand Forks, ND 58202-8057.

IEEE Log Number 9210957.

II. Problem Statement

Consider a canonical form of linear programming problems described as

$$\text{minimize} \quad c^T v, \tag{1}$$

$$\text{subject to} \quad Av \le b, \tag{2}$$

$$v \ge 0, \tag{3}$$

where $v \in \boldsymbol{R}^n$ is a column vector of decision variables, $c \in \boldsymbol{R}^n$ and $b \in \boldsymbol{R}^m$ are column vectors of cost coefficients and right-hand side parameters, respectively, $A \in \boldsymbol{R}^{m \times n}$ is a constraint coefficient matrix, $m \le n$, and the superscript T denotes the transpose operator.

Equation (1) defines a linear objective function that is to be minimized. Equations (2) and (3) define a polyhedral feasible region in the n-dimensional Euclidean space. Let $\hat{V}$ denote the feasible region defined by (2) and (3) and let $v^* = [v_1^*, v_2^*, \ldots, v_n^*]^T$ denote an optimal solution; i.e., $v^* = \arg\min_{v \in \hat{V}} c^T v$. For operational reasons, we assume that the feasible region $\hat{V}$ is nonempty and the minimal objective function value $c^T v^*$ is bounded. Hence there always exists an upper bound $v_{\max}$ ($v_{\max} > 0$) such that $0 \le v_i^* \le v_{\max}$ for $i = 1, 2, \ldots, n$. Furthermore, since the inequality of (2) can be easily converted to equality by adding m slack variable(s), without loss of generality, we thereafter consider the equivalent linear program described as

$$\text{minimize} \quad c^T v, \tag{4}$$

$$\text{subject to} \quad Av = b, \tag{5}$$

$$0 \le v_i \le v_{\max}, \qquad i = 1, 2, \ldots, n. \tag{6}$$

III. Network Configuration

The proposed recurrent neural network for linear programming consists of n massively connected artificial neurons. Each neuron represents one decision variable. The state equations of the recurrent neural network are presented as

$$\dot{u}(t) = -\alpha A^T A v(t) + \alpha A^T b - \beta \exp(-\eta t) c, \tag{7}$$

$$v_i(t) = \frac{v_{\max}}{1 + \exp[-\xi u_i(t)]}, \qquad i = 1, 2, \ldots, n, \tag{8}$$

where $u(t)$ is an n-dimensional column vector of instantaneous net inputs to neurons, $v(t)$ is an n-dimensional column vector of activation states corresponding to the decision variable vector,

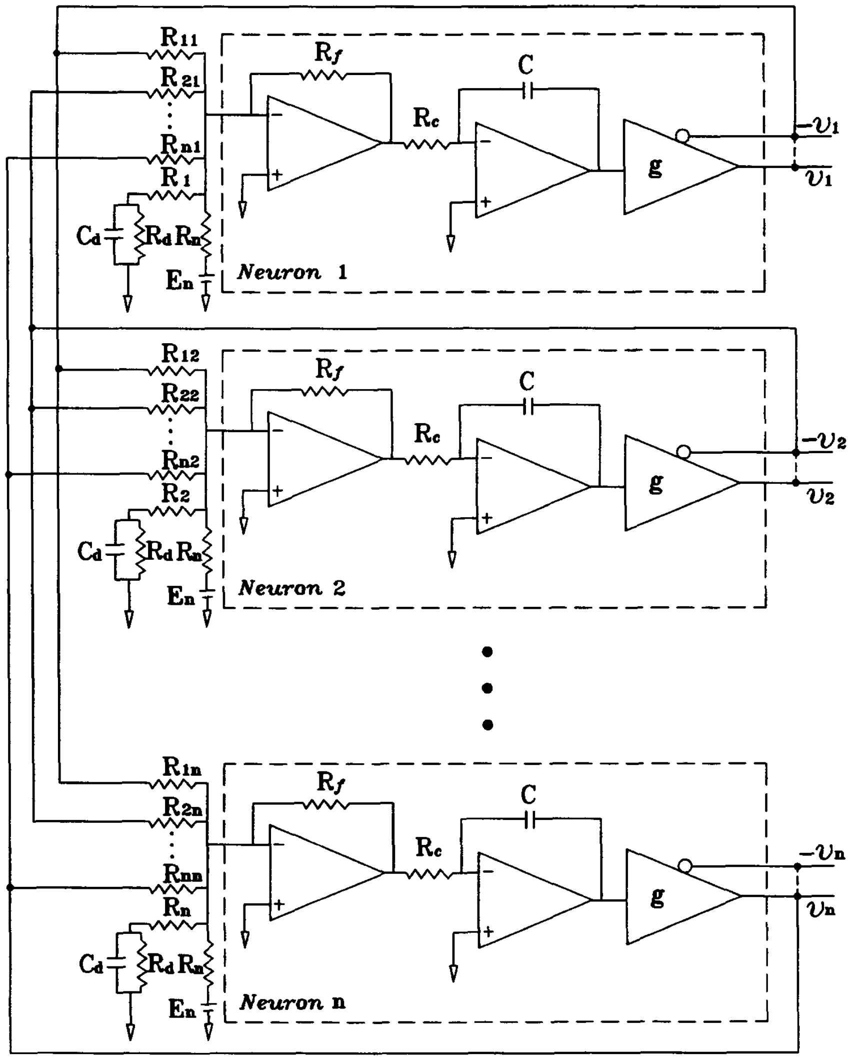

Fig. 1. Circuit schematic of an op-amp-based analog neural network.

$u(0)$ and $v(0)$ are specified, and α, β, η, and ξ are positive scalar parameters.

Equation (7) defines the neural network dynamics, in which the first two terms on the right-hand side enforce the penalization of violation of constraint (5) and the third term enforces the minimization of the objective function. Equation (8) defines the sigmoid activation functions and enforces constraint (6). Since the infimum and supremum of the activation state are 0 and $v_{\max}$, respectively, the state space of the recurrent neural network is defined as a hypercube $V = [0, v_{\max}]^n$. Obviously, $\hat{V} \subseteq V$. Let W and θ be the synaptic connection weight matrix and biasing threshold vector, respectively. The connection weight matrix is defined as $-\alpha A^T A$ and the biasing threshold vector of the neurons is defined as $\alpha A^T b - \beta \exp(-\eta t) c$; i.e., $W = -\alpha A^T A$ and $\theta(t) = \alpha A^T b - \beta \exp(-\eta t) c$. In other words, the connection weight w_{ij} from neuron i to neuron j is defined as $-\alpha \sum_{k=1}^{m} a_{ki} a_{kj}$ and the biasing threshold θ_i of neuron i is defined as $\alpha \sum_{k=1}^{m} a_{ki} b_k - \beta c_i \exp(-\eta t)$, where a_{ij} are the elements in the ith row and the jth column of A, and b_i and c_i are the ith elements of b and c, respectively. Evidently, the connection weights are symmetric; i.e., $\forall i, j, w_{ij} = w_{ji}$. Furthermore, the eigenvalues of W are always real and nonpositive, and there are at least $n - m$ zero eigenvalues. There is a decaying term as well as a constant term in the biasing threshold of neuron i if $c_i \neq 0$. Because of the decaying term in the threshold, the proposed recurrent neural network is a time-varying dynamic system. The role of parameter α is to scale the connection weights and the constant term of the biasing threshold. The roles of parameters β and η are to scale the decaying term of the biasing threshold. The role of parameter ξ is to scale the sensitivity of the activation function.

IV. Circuit Realization

The essence of the neural network approach to linear programming lies in its potential of electronic implementation where the optimization procedure is truly parallel and distributed. This section discusses an analog circuit that realizes the proposed recurrent neural network.

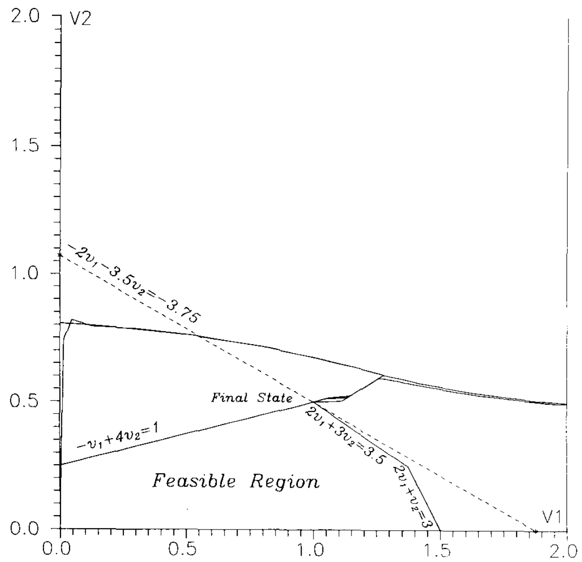

Fig. 2. Feasible region and state trajectories of the simulated neural network in the example.

Equations (7) and (8) show that the proposed recurrent neural network is easy to realize in an analog circuit. The analog recurrent neural network consists of n neurons (processing elements). Let g denote the sigmoid activation function; i.e., $g[u_i(t)] = v_{\max}/[1 + \exp(-\xi u_i(t))]$ for $i = 1, 2, \ldots, n$. A circuit schematic of an op-amp-based analog circuit is shown in Fig. 1, where both v_i and $-v_i$ terminals are given for neuron i ($i = 1, 2, \ldots, n$). The symmetric connection weight w_{ij} from neuron i to neuron j can be implemented by a feedback resistor R_f and a connection resistor with ohmic value R_{ij} such that $|w_{ij}| = R_f/R_{ij}$; i.e., $R_{ij} = R_f/|w_{ij}| = R_f/|\Sigma_{k=1}^m a_{ki}a_{kj}|$. If $w_{ij} > 0$, then the v_i terminal of the analog neuron is to be used; if $w_{ij} < 0$, then the $-v_i$ terminal is to be used, instead of realizing a negative resistor; and if $w_{ij} = 0$, then no link is to be established. The constant term $\alpha\Sigma_{k=1}^m a_{ki}b_k$ in the biasing threshold θ_i of neuron i can be realized by a voltage source E_i and a series resistor R_i such that $R_f E_i/R_i = \theta_i$ or by a current source I_i such that $R_f I_i = \theta_i$. The decaying term $\beta c_i \exp(-\eta t)$ in the threshold θ_i can be realized by a discharging loop. Initially, C_d in neuron i is charged $-\beta c_i$ V for $i = 1, 2, \ldots, n$. The parallel resistor R_d is to provide a discharging path for realizing the decaying term of the threshold. Let $R_d \ll R_f$. Then the decaying time constant is approximately $R_d C_d$; i.e., $\eta \approx 1/(R_d C_d)$.

V. Asymptotic Properties

This section presents the analytical results on the asymptotic stability of the recurrent neural network, and the feasibility and optimality of the steady-state solutions.

Theorem 1: The proposed recurrent neural network is asymptotically stable in the large; i.e., $\forall v(0) \in V$, $\exists \bar{v} \in V$ such that $\lim_{t\to\infty} v(t) = \bar{v}$, where $\bar{v}$ is a steady state of $v(t)$.

Proof: Let

$$E[v(t), t] = \frac{1}{2}[Av(t) - b]^T[Av(t) - b] + \frac{\beta}{\alpha}c^T[\tilde{v} + v(t)]\exp(-\eta t), \quad (9)$$

where $\tilde{v} = [\tilde{v}_1, \tilde{v}_2, \ldots, \tilde{v}_n]^T$, $\tilde{v}_i = \operatorname{sgn}(c_i)(v_{\max} + \epsilon)$, $\operatorname{sgn}(\cdot)$ is the signum function, and ϵ is an arbitrary positive number. Obviously, $\forall t \geq 0$, $\forall v \in V$, $E[v(t), t] \geq 0$ and $E[v(t), t] \to \infty$ as $\|v\| \to \infty$. Since $-\dot{u}(t)/\alpha = A^TAv(t) - A^Tb + \beta\exp(-\eta t)c/\alpha$ from (7),

$$\frac{dE[v(t), t]}{dt} = \left[A^TAv(t) - A^Tb + \frac{\beta}{\alpha}\exp(-\eta t)c\right]^T\dot{v}(t) - \frac{\eta\beta}{\alpha}c^T[\tilde{v} + v(t)]\exp(-\eta t)$$

$$= -\frac{1}{\alpha}\dot{u}^T(t)\dot{v}(t) - \frac{\eta\beta}{\alpha}c^T[\tilde{v} + v(t)]\exp(-\eta t).$$

Since $\alpha > 0$ and $v(t)$ is monotone increasing with respect to $u(t)$, $\dot{u}^T(t)\dot{v}(t) \geq 0$; i.e., $-\dot{u}^T(t)\dot{v}(t)/\alpha \leq 0$. Furthermore, since $\eta > 0$, $\beta > 0$, and $c^T[\tilde{v} + v(t)] > 0$, $-\eta\beta c^T[\tilde{v} + v(t)]\exp(-\eta t) < 0$ for $0 \leq t < \infty$; i.e., $dE[v(t), t]/dt < 0$. Therefore, $E[v(t), t]$ is a strict Lyapunov (energy) function. ■

Theorem 2: The steady state of the proposed neural network represents a feasible solution to the linear programming problem defined by (4), (5), and (6); i.e., $\bar{v} \in \hat{V}$.

Proof: Let a penalty function be defined as

$$P[v(t)] = \frac{1}{2}[Av(t) - b]^T[Av(t) - b] = \frac{1}{2}\|Av(t) - b\|^2, \quad (10)$$

where $\|\cdot\|$ is the Euclidean norm. Obviously, $\forall \bar{v}$, $P[\bar{v}] = 0$ iff $A\bar{v} = b$. The proof of Theorem 1 shows that the energy function $E[v(t), t]$ is nonnegative and monotone decreasing with respect to time t. Moreover, because both $\|Av - b\|$ and c^Tv are convex, $E[v(t), t]$ is also convex in the state space. Therefore, $\lim_{t\to\infty} E[v(t), t] = 0$. Since $\lim_{t\to\infty}\exp(-\eta t) = 0$, $\lim_{t\to\infty} P[v(t)] = 0$ or $A\bar{v} = b$. ■

According to Theorems 1 and 2, the asymptotic stability of the proposed recurrent neural network implies the feasibility of the steady-state solution of the recurrent neural network.

Since the activation function $g[v_i(t)] = v_{\max}/[1 + \exp(u_i(t))]$, the first-order derivative of $g[u_i(t)]$ with respect to $u_i(t)$, $dg[u_i(t)]/du_i = \xi v_i(t)[v_{\max} - v_i(t)]/v_{\max}$ for $i = 1, 2, \ldots, n$, and the maximum value of the derivative is $\xi v_{\max}/4$. Because $v_i(t) = g[u_i(t)]$, $\dot{v}_i(t) = \{dg[u_i(t)]/du_i\}\dot{u}_i \leq \xi v_{\max}\dot{u}_i(t)/4$ and $\dot{v}(t) \leq \xi v_{\max}[Wv(t) + \theta(t)]/4$. Thus the recurrent neural network converges slower than a linearized neural network $\dot{v}'(t) = \xi v_{\max}[Wv'(t) + \theta(t)]/4$. According to linear systems theory (e.g., see [8]), the state trajectory of the linearized neural network can be described as $v'(t) = k_0 + \Sigma_{j=1}^n k_j\exp(-\xi v_{\max}\alpha\lambda_j t/4) + k_{n+1}\exp(-\eta t)$, where k_0 and k_{n+1} are constant vectors, k_j ($j = 1, 2, \ldots, n$) is a vector of constants or polynomials of t, and λ_i is the ith eigenvalue of A^TA (i.e., $\xi v_{\max}\alpha\lambda_j/4$ is the jth nonzero eigenvalue of $\xi v_{\max}W/4$). The convergence rate of $v'(t)$ in the linearized neural network is dominated by $\min\{\xi v_{\max}\alpha\lambda_{\min}/4, \eta\}$, where $\lambda_{\min} = \min\{\lambda_i | \lambda_i \neq 0;\ i = 1, 2, \ldots, n\}$ (i.e., $\lambda_{\min}$ is the smallest singular value of A). Since $-\alpha A^TAv(t) + \alpha A^Tb = -\alpha\nabla\|Av(t) - b\|_2^2$ and $-\beta\exp(-\eta t)c = -\beta\exp(-\eta t)\nabla[c^Tv(t)]$, where ∇ denotes the gradient operator, the first two terms in the right-hand side of (7) enforce constraint satisfaction and the third term enforces objective minimization. Parameters α and $\beta\exp(-\eta t)$ serve as weighting

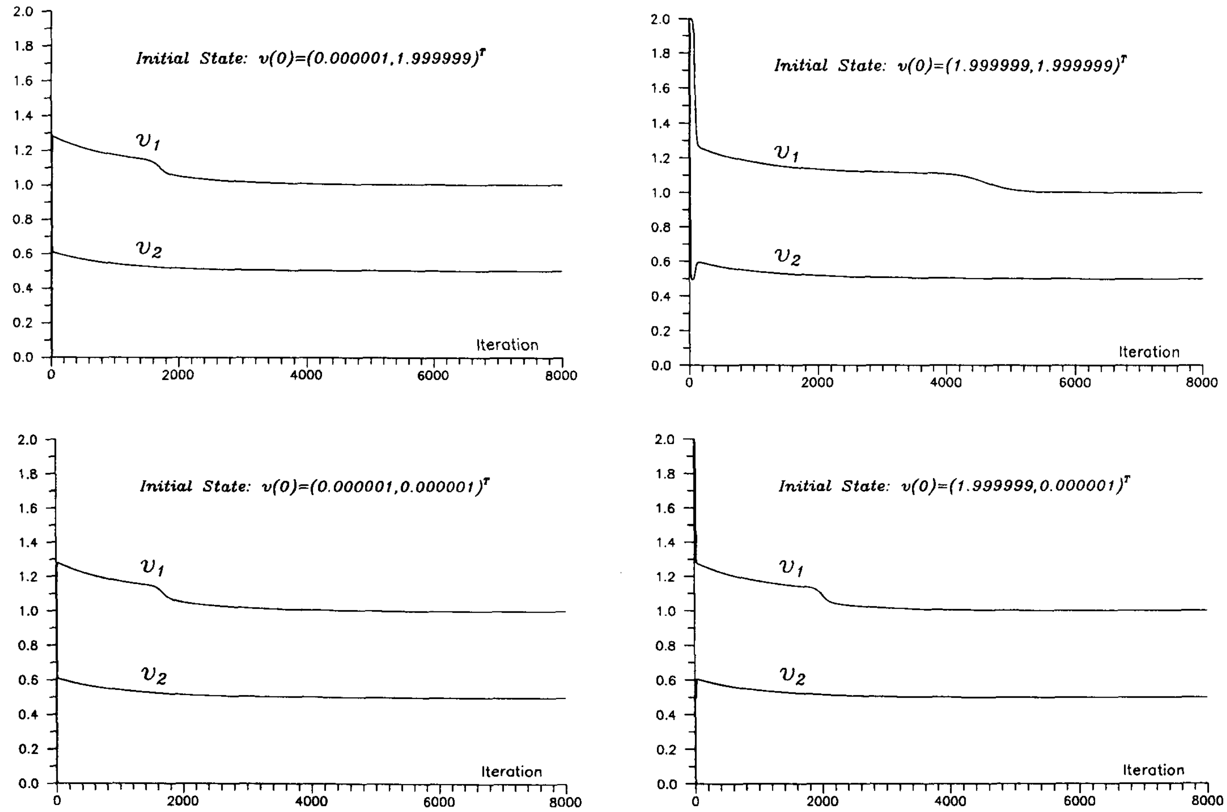

Fig. 3. Transient behavior of the states in the simulated neural network in the example.

factors to scale the effects of constraint satisfaction and objective minimization, respectively. On one hand, since the constraints must be satisfied absolutely, whereas the objective function is to be minimized subject to the constraints (over the feasible region), constraint satisfaction holds a higher priority than objective minimization. The priority requirement is realized by using the constant weighting parameter α and the decaying weighting parameter $\beta \exp(-\eta t)$. As time elapses, the decaying parameter $\beta \exp(-\eta t)$ decreases monotonically. Consequently, a relatively lighter weight is put on objective minimization than on constraint satisfaction as the activation state vector of the neural network approaches its equilibrium. On the other hand, since a feasible solution is not necessarily an optimal solution, it is essential for the decaying bias term not to vanish until $v(t)$ reaches a feasible solution in order to sustain the minimization process. Because the decaying rate of the biasing threshold is determined by η and the fastest convergence rate of $v(t)$ toward a feasible solution is determined by $\alpha \xi V_{\max} \lambda_{\min}/4$, it is usually necessary that $\eta \leq \alpha \xi V_{\max} \lambda_{\min}/4$ for the activation state to reach an optimal solution with an arbitrary initial state. In designing a specific neural network, a large value of η above the upper bound may result in a suboptimal solution. A simple way is to set $\eta \ll \alpha \xi V_{\max} \lambda_{\min}/4$; e.g., $\eta = 100(\alpha \xi V_{\max} \lambda_{\min}/4) = 25\alpha \xi V_{\max} \lambda_{\min}$. Consequently, the convergence rate of the recurrent neural network is dominated by the objective minimization term and the convergence time is approximately $5/\eta$ s.

VI. Simulation Results

Consider a linear programming example as

$$\begin{aligned} \min \quad & -2v_1 - 3.5v_2, \\ \text{s.t.} \quad & -v_1 + 4v_2 \leq 1, \\ & 2v_1 + 3v_2 \leq 3.5, \\ & 2v_1 + v_2 \leq 3, \\ & v_1 \geq 0, v_2 \geq 0. \end{aligned}$$

The optimal solution of this problem is $v^* = [1.0, 0.5]^T$, the minimal objective value is $c^T v^* = -3.75$, and the minimum nonzero eigenvalue of $A^T A$ is $\lambda_{\min} = 1$.

Based on $v_{\max} = 2$, $\alpha = \beta = 10^4$, $\xi = 10$, $\eta = 10^3$, and $\Delta t = 10^{-6}$, the simulation results show that the steady state of the recurrent neural network always accurately represents the optimal solution starting from four different corners of a 2×2 square in the state space; i.e., $\bar{v} = [1.000000, 0.500000]^T$ and $c^T \bar{v} = -3.750000$. Figures 2 and 3 illustrate, respectively, the convergent state trajectories on the $v_1 - v_2$ plane and the convergent states versus the number of time intervals (iterations). Figures 2 and 3 indicate that state v_1 initially remains unchanged for all the four starting points. It is interesting to note that the four trajectories merge in their final stages. This phenomenon indicates a deep valley of the energy function in the vicinities of the trajectories. Figure 4 illustrates the convergence patterns of the values of objective function, penalty function, and energy function versus the number of time intervals.

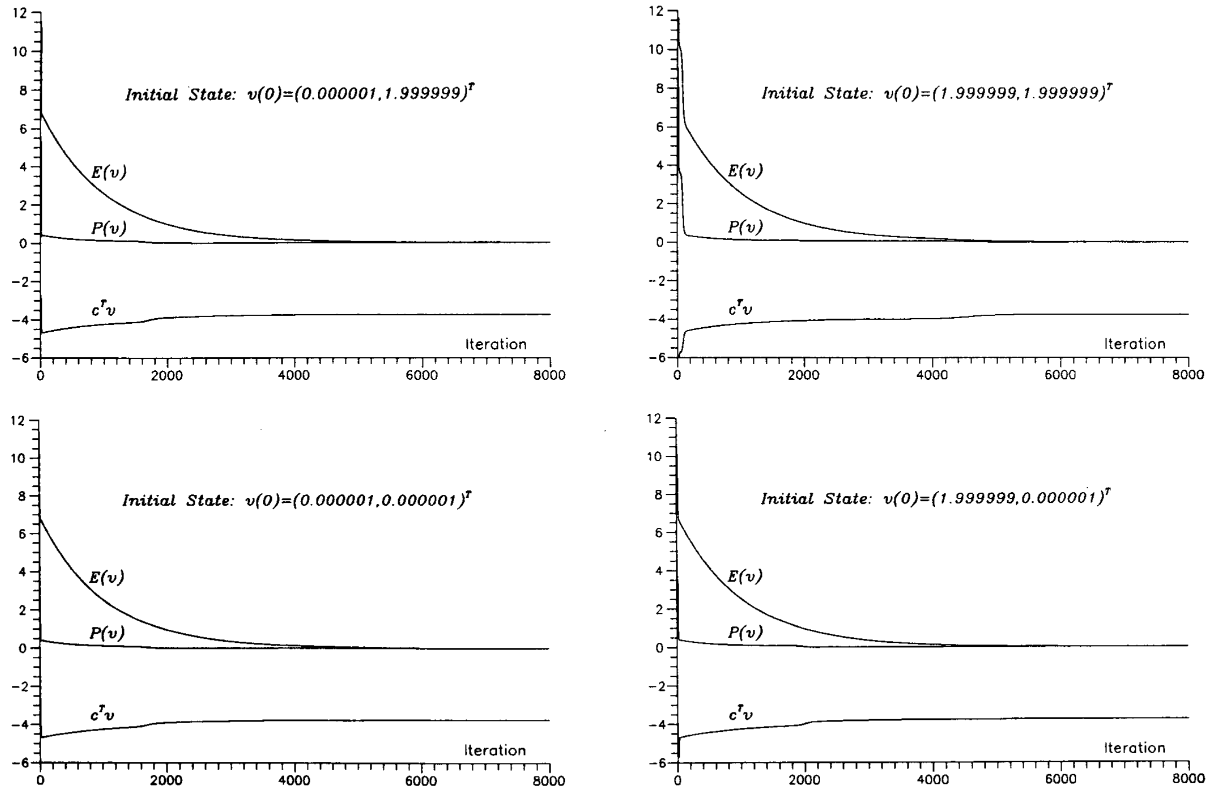

Fig. 4. Transient behavior of the values of the objective function, penalty function, and energy function.

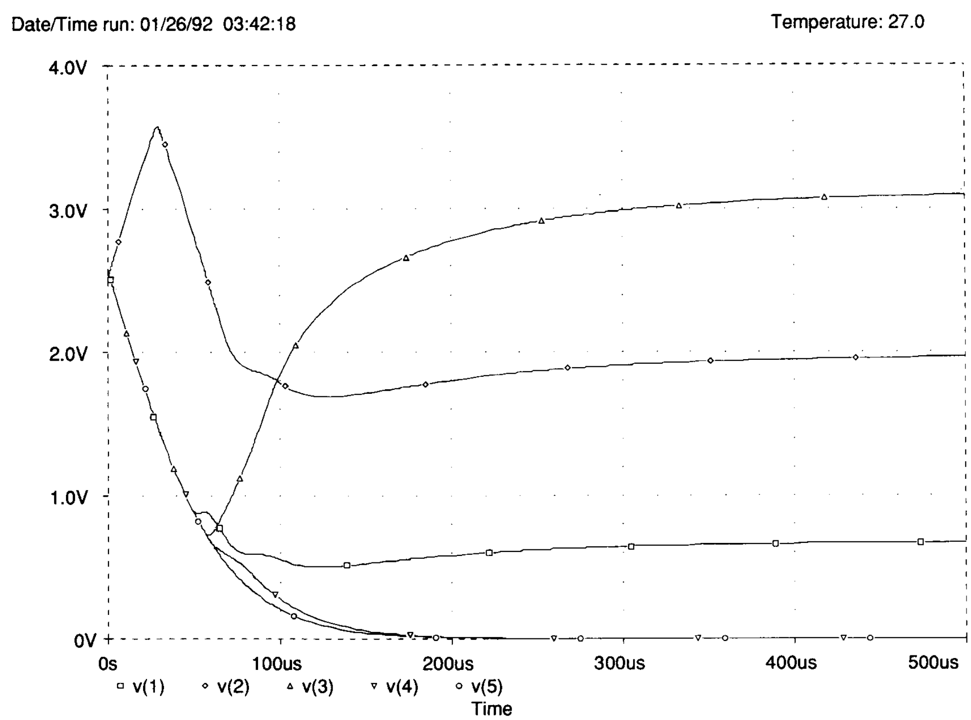

Fig. 5. Transient behavior of the states in the op-amp-based analog neural network.

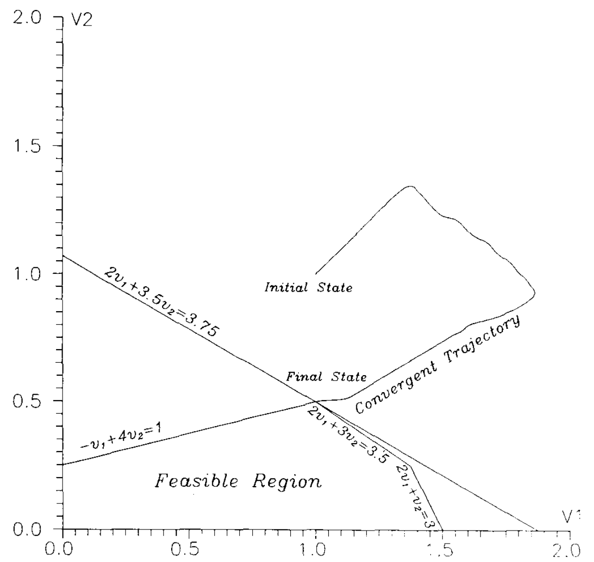

Fig. 6. Feasible region and state trajectory of the op-amp-based analog neural network.

Using PSpice, the steady-state vector of an analog neural network based on LT1028 op-amps was $\bar{v} = [1.001, 0.500]^T$, where $R_f = 1$ MΩ, $R_i = 10$ kΩ, $R_c = 10$ kΩ, $C = 100$ μF, $R_d = 1$ Ω, $C_d = 100$ μF, $v_{\max} = 2$ V, $\alpha = \beta = 100$, $\eta \approx 10\,000$, and $\xi = 2000$. Figures 5 and 6 illustrate, respectively, the transient behavior and convergent state trajectory of the op-amp-based analog neural network in solving the linear program.

VII. Conclusions

In this paper, a recurrent neural network with a time-varying threshold vector for solving linear programming problems has been proposed. An op-amp-based analog circuit has been designed, the asymptotic properties of the neural network have been analyzed, and an illustrative example has also been detailed. It has been shown that the proposed recurrent neural network is easy to implement in an analog circuit. It has also been substantiated that the proposed recurrent neural network is capable of generating optimal solutions to linear programming problems. The proposed recurrent neural network has been applied to solve special linear programming problems; e.g., [9]. Because of these desirable features, the proposed recurrent neural network has been perceived as a potentially powerful computational model for solving real-time and large-scale linear programming problems.

References

[1] 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. CAS-33, no. 5, pp. 533–541, 1986.

[2] M. P. Kennedy and L. O. Chua, "Unifying the Tank and Hopfield linear programming circuit and the canonical nonlinear programming circuit of Chua and Lin," *IEEE Trans. Circuits Syst.*, vol. CAS-34, no. 2, pp. 210–214, 1988.

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

[4] A. Rodríguez-Vázquez, R. Domínguez-Castro, A. Rueda, J. L. Huertas, and E. Sánchez-Sinencio, "Nonlinear switched-capacitor 'neural networks' for optimization problems," *IEEE Trans. Circuits Syst.*, vol. 37, no. 3, pp. 384–397, 1990.

[5] C. Y. Maa and M. A. Shanblatt, "Linear and quadratic programming neural network analysis," *IEEE Trans. Neural Networks*, vol. NN-3, pp. 580–594, 1992.

[6] J. Wang, "On the asymptotic properties of recurrent neural networks for optimization," *Int. J. Pattern Recognition Artificial Intelligence*, vol. 5, no. 4, pp. 581–601, 1991.

[7] J. Wang and V. Chankong, "Recurrent neural networks for linear programming: Analysis and design principles," *Comput. Operat. Res.*, vol. 19, nos. 3/4, pp. 297–311, 1992.

[8] T. Kailath, *Linear Systems*. Englewood Cliffs, NJ: Prentice-Hall, 1980.

[9] J. Wang, "Analog neural network for solving the assignment problem," *Electron. Lett.*, vol. 28, no. 11, pp. 1047–1050, 1992.

Minimum Stopband Attenuation of Cauer Filters without Elliptic Functions and Integrals

D. M. Rabrenović and M. D. Lutovac

Abstract—**This paper provides simple formulas for precise determination of the minimum stopband attenuation of elliptic filters as a function of the selectivity factor. In contrast to other commonly used procedures, the knowledge and use of elliptic functions and integrals is not a prerequisite for the calculation.**

I. Introduction

Elliptic filters offer the steepest rolloff characteristics, and consequently meet an assigned set of filter performance specifications with the lowest filter order. As a consequence of the extreme interest in these filters, many different design procedures were developed [1]–[6]. Unfortunately, all of them presume knowledge of the Jacobian elliptic functions and integrals, which are unfamiliar to most filter designers.

The determination of the minimum stopband loss as a function of the normalized stopband frequency, as presented in this paper, does not imply the use of the Jacobian elliptic functions and integrals. This is achieved by applying the Chebyshev rational functions in the well-known relations formerly used only for the Chebyshev polynomials.

II. New Relations

The transfer function $H_n(s)$ is defined as input-to-output voltage ratio,

$$H_n(s) = \frac{V_1(s)}{V_2(s)}. \tag{1}$$

The Chebyshev rational characteristic function $R_n(\omega)$ is then defined via

$$\epsilon^2 R_n^2(\omega) = H_n(j\omega)H_n(-j\omega) - 1 \tag{2}$$

with

$$-1 \le R_n(\omega) \le 1 \quad \text{for } -1 \le \omega \le 1.$$

$R_n(\omega)$ can also be expressed as a function of the attenuation

Manuscript received January 31, 1992; revised March 15, 1993. This paper was recommended by Associate Editor S. Karni.

D. M. Rabrenović is with the Faculty of Electrical Engineering, University of Belgrade, 11001 Belgrade, Yugoslavia.

M. D. Lutovac is with the Telecommunications and Electronics Institute, 11080 Belgrade, Yugoslavia.

IEEE Log Number 9210956.