# A Recurrent Neural Network for Real-Time Matrix Inversion*

Jun Wang

*Department of Industrial Technology*
*University of North Dakota*
*Grand Forks, North Dakota 58202-8057*

ABSTRACT

A recurrent neural network for computing inverse matrices in real-time is proposed. The proposed recurrent neural network consists of $n$ independent subnetworks where $n$ is the order of the matrix. The proposed recurrent neural network is proven to be asymptotically stable and capable of computing large-scale nonsingular inverse matrices in real-time. An op-amp based analog neural network is discussed. The operating characteristics of the op-amp based analog neural network is also demonstrated via an illustrative example.

## 1. INTRODUCTION

Matrix inversion has been widely used in a variety of applications such as robotics, control, and signal processing. For many large-scale problems, the orders of the matrices are very large and large-scale inverse matrices often need to be computed in real-time for monitoring and controlling dynamic systems. Large matrix inversion using existing algorithms in real-time is usually not efficient due to the nature of the sequential processing. For such applications, parallel distributed processing is more desirable.

Recently, neural networks have been proposed for matrix inversion and related applications. For example, Jang *et al.* [2] and Luo and Zheng [5] use modified Hopfield networks for matrix inversion. Neural networks have also been proposed for solving simultaneous linear equations [4, 8]. These investigations have shed light on neural network research for matrix inversion. The proposed recurrent neural networks can operate concurrently in real time. It

*This study was partially supported by a summer research professorship from the Graduate School of the University of North Dakota.



 0096-3003/93/$6.00

is the nature of parallel and distributed processing that renders the neural network approach the computational advantages over the existing sequential algorithms in real-time applications.

In this paper, we develop a linear recurrent neural network for matrix inversion. The difference between the proposed recurrent neural network and the existing ones for matrix inversion [2, 5] lies in their network architecture and dynamics. The proposed recurrent neural network can be decomposed into $n$ independent subnetworks and can be easily implemented in an electronic circuit. The proposed recurrent neural network is also proven to be asymptotically stable in large and capable of generating inverse matrices within a few characteristic time constants.

## 2. NETWORK CONFIGURATION

We are interested in computing the inverse matrix $A^{-1}$ of a large-scale $n \times n$ nonsingular matrix $A$ in real time. The proposed recurrent neural network for matrix inversion consists of $n^2$ neurons arranged spatially in a matrix form. Let $V(t)$ be an $n \times n$ activation state matrix of the recurrent neural network corresponding to the inverse matrix $A^{-1}$. The activation state $v_{ij}(t)$ of neuron in the $i$th row and $j$th column of $V(t)$ represents the element of the same position in the inverse matrix $A^{-1}$. The dynamics of the proposed neural network can be described by the following matrix-valued differential equation:

$$\frac{dV(t)}{dt} = -\eta A^T A V(t) + \eta A^T, \tag{1}$$

where $\eta$ is a positive scalar parameter and $A^T$ denotes the transpose of $A$. The connection weights matrix of the recurrent neural network is $-\eta A^T A$ and the biasing threshold matrix is $A^T$.

The state equation (1) indicates that the proposed recurrent neural network for matrix inversion is essentially a multivariate *linear* dynamic system. It can be viewed as a special case of general continuous recurrent neural networks where the activation function is monotone increasing. The proposed recurrent neural network for matrix inversion is a variation of the recurrent neural network for solving simultaneous linear equations presented in [8] in which the activation states constitute a vector instead of a matrix.

A closer examination of (1) reveals that the proposed recurrent neural network can actually be decomposed into $n$ independent subnetworks where $n$ is the order of $A$. Each subnetwork represents a column vector of $V(t)$.

Let's define $v_i(t)$ as the $i$th column vector of $V(t)$ for $i = 1, 2, \ldots, n$. The dynamics of the $i$th subnetwork can be expressed as follows.

$$\frac{dv_i(t)}{dt} = -\eta A^T A v_i(t) + \eta A_i, \tag{2}$$

where $A_i$ is defined as the $i$th column vector of $A^T$ (i.e., the $i$th row vector in $A$). (2) indicates that each subnetwork is essentially the same as the recurrent neural network presented in [8]. Let $W$ be the connection weight matrix. The connection weight matrix $-\eta A^T A$ is identical for each subnetwork (i.e., $W = -\eta A^T A$) and the biasing threshold vector for the $i$th subnetwork is $A_i$. In other words, the connection weight $w_{ij}$ between neurons $v_{ik}(t)$ and $v_{jk}(t)$ in the $k$th column (subnet) is $-\eta \sum_{l=1}^{n} a_{li} a_{lj}$ (i.e., $w_{ij} = -\eta \sum_{l=1}^{n} a_{li} a_{lj}$) and the biasing threshold of activation state $v_{ik}(t)$ is $a_{ki}$ where $a_{ij}$ are the element in the $i$th row and the $j$th column of $A$. Figure 1 depicts the architecture of the proposed recurrent neural network for matrix inversion.

Because the connection weight matrices for every subnetwork are identical, the proposed recurrent neural network can also be realized by a single subnetwork with time-sharing threshold vectors. In each time slot, the

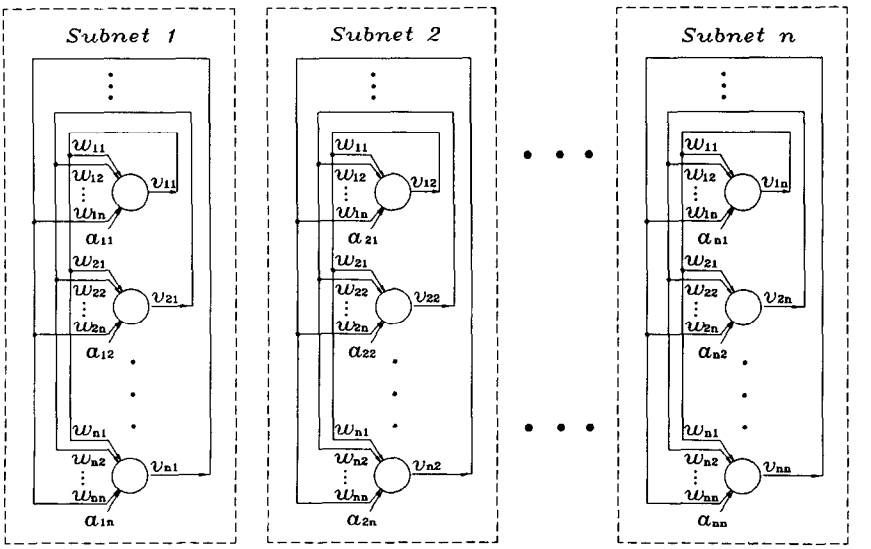



FIG. 1. Architecutre of the proposed recurrent neural network.

subnetwork biased by the corresponding threshold vector generates one column vector of the inverse matrix. Therefore, the spatial complexity of the neural network can be reduced by a factor of $n$.

## 3. ASYMPTOTIC PROPERTIES

The following two propositions substantiate the asymptotic stability of the activation states and the capability of the recurrent neural network for matrix inversion.

PROPOSITION 1. *If $A$ is nonsingular, then the equilibrium state of the recurrent neural network is asymptotically stable in the large.*

PROOF. Let's first prove the positive definiteness of $A^TA$. If $A$ is nonsingular, then $Ax = 0$ iff $x = 0$ for an $n$-vector $x$. For $\forall x \neq 0$, $x^TA^TAx = (Ax)^TAx = \|Ax\|^2 > 0$ where $\|\cdot\|$ is the Euclidean norm. Thus, $A^TA$ is a positive definite matrix. Since $-\eta A^TA$ is symmetric, all the eigenvalues of $-\eta A^TA$ are real. Furthermore, since $\eta > 0$ and $A^TA$ is positive definite, all the eigenvalues of $-\eta A^TA$ are negative. According to linear systems theory [1, 3], the equilibrium state of the recurrent neural network is in large asymptotically stable. ■

PROPOSITION 2. *If $A$ is nonsingular, then the equilibrium state matrix of the recurrent neural network represents the inverse matrix.*

PROOF. At the equilibrium state $\bar{V}$, $d\bar{V}/dt = 0$. Thus, $A^TA\bar{V} = A^T$. Since $A^TA$ is positive definite, $(A^TA)^{-1}$ exists. Therefore, $\bar{V} = (A^TA)^{-1}A^T = A^{-1}(A^T)^{-1}A^T = A^{-1}I = A^{-1}$, where $I$ is the identity matrix. ■

According to the well-established linear systems theory [1, 3], the convergence rate of the proposed recurrent neural network is dominated by the smallest eigenvalue of $\eta A^TA$, $\lambda_{\min} = \min\{\lambda_i;\ i = 1, 2, \ldots, n\}$ where $\lambda_i$ is the $i$th eigenvalue of $\eta A^TA$. The recurrent neural network can reach the steady state in approximately $5/\lambda_{\min}$ time units. Since the $n$ subnetworks share identical connection weight matrix $-\eta A^TA$, the convergence rate of every subnetwork is the same. Furthermore, since the value of the positive scaling parameter $\eta$ is inversely proportional to the transient time required by the neural network, the convergence rate of the solution process can be controlled by selecting a sufficiently large parameter $\eta$.

## 4. ANALOG REALIZATION

The major advantage of neural network approach to matrix inversion lies in its potential of hardware realization. The state dynamics equation (1) indicates that the proposed recurrent neural network can be easily implemented in an analog circuit. The proposed electronic neural network consists of $n^2$ neurons (processing elements). $n$ neurons constitute an independent subnetwork. Each neuron can be implemented by three operational amplifiers: a summing amplifier, an integrating amplifier, and an inverting amplifier. A schematic diagram of an analog circuit for a subnetwork is shown in Figure 2. The symmetric connection weight $w_{ij}$ between neurons $v_{ik}$ and $v_{jk}$ $(i, j, k = 1, 2, \ldots, n)$ can be implemented by a feedback resistor $R_f$ and a connection resistor with ohmic value $R_{ij}$ such that $R_f/R_{ij} = |w_{ij}| = |\eta\Sigma_{l=1}^{n} a_{li}a_{lj}|$; i.e., $R_{ij} = R_f/|w_{ij}| = R_f/|\eta\Sigma_{l=1}^{n} a_{li}a_{lj}|$. If $w_{ij} > 0$ (i.e.,

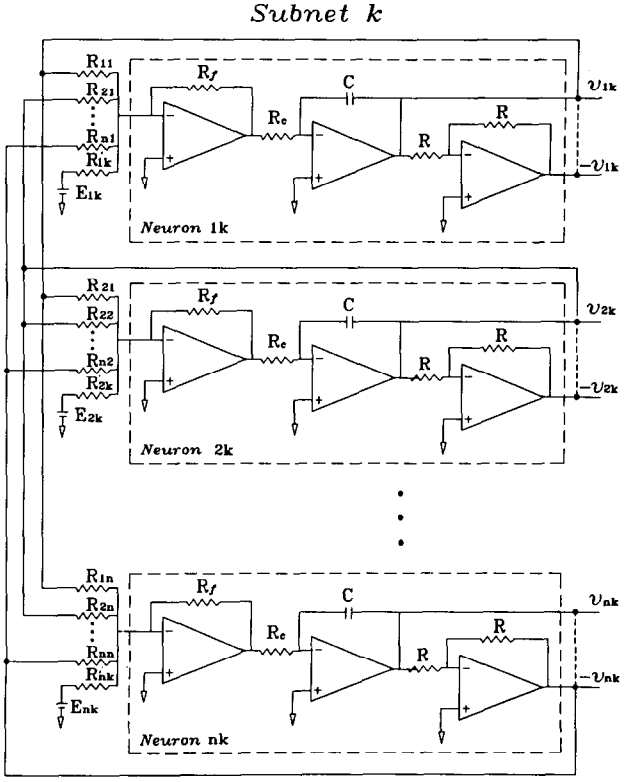



FIG. 2. Schematic diagram of an op-amp based analog circuit for a subnetwork.

$\sum_{l=1}^{n} a_{li}a_{lj} < 0$), then the positive terminal $v_{ik}$ of the analog neuron is to be used; if $w_{ij} < 0$ (i.e., $\sum_{l=1}^{n} a_{li}a_{lj} > 0$), then the negative terminal $-v_{ik}$ is to be used; and if $w_{ij} = 0$ (i.e., $\sum_{l=1}^{n} a_{li}a_{lj} = 0$), then no link is to be established. The biasing threshold $a_{ji}$ for neuron $v_{ij}$ can be realized by a voltage source $E_{ij}$ with a series resistor $R'_{ij}$ such that $R_f E_{ij}/R'_{ij} = a_{ji}$ or by a current source $I_{ij}$ such that $R_f I_{ij} = a_{ji}$.

In real-time applications, for a different set of signal matrices $A$, the values of connection resistors and biasing voltage sources have to be changed accordingly. The connection resistors $R_{ij}$ can be implemented by a controllable device such as an optoelectronic device or MOS field-effect transistor. The biasing voltage sources can be realized by a voltage-controlled voltage source. In the case that the time-varying threshold vector is used, $n$ multiplexers have to be used.

Since the outputs of recurrent neural networks implemented in linear analog circuits are linear only within a range, output saturation is practically inevitable. Sundareshan and Sudharsanan [6, 7] analyze the stability properties of a class of recurrent neural networks with piece-wise linear activation function. In their work, they prove the equilibrium states of recurrent neural networks with piece-wise linear activation function always rest at the nonsaturation portion of the activation function. Sudharsanan and Sundareshan's result implies that the proposed recurrent neural network will still function in presence of output saturation. The proposed linear recurrent neural network, however, is easier to implement than the piece-wise linear neural networks because the range of nonsaturation needs not to be taken into consideration in designing a neural network.

## 5. COMPUTER SIMULATION

The following illustrative example demonstrates the operating characteristics of the op-amp based analog neural network simulated using *PSpice* version 5.0A.

EXAMPLE. Let

$$A = \begin{pmatrix} 0.43 & 0.55 & 0.32 & 0.82 & 0.29 & 0.05 \\ 0.24 & 0.61 & 0.69 & 0.23 & 0.52 & 0.28 \\ 0.83 & 0.09 & 0.14 & 0.25 & 0.62 & 0.32 \\ 0.44 & 0.58 & 0.26 & 0.19 & 0.28 & 0.52 \\ 0.27 & 0.73 & 0.56 & 0.24 & 0.17 & 0.09 \\ 0.15 & 0.92 & 0.45 & 0.62 & 0.71 & 0.41 \end{pmatrix}.$$

Then

$$A^{-1} = \begin{pmatrix} -0.0418 & -1.0001 & 1.1190 & 0.0808 & 1.4208 & -0.5997 \\ -1.3680 & -2.7717 & 0.5004 & -0.4841 & 2.9891 & 1.6270 \\ 1.1492 & 3.4549 & -0.9789 & 0.4339 & -1.5148 & -1.9536 \\ 2.1298 & 1.1111 & -0.9389 & 0.6120 & -2.1636 & -0.5868 \\ -1.3450 & -0.5792 & 1.3696 & -1.9267 & 0.5829 & 1.8063 \\ 0.9328 & 2.1163 & -1.4101 & 2.9915 & -3.3021 & -1.0887 \end{pmatrix}.$$

Let $\eta = 10^6$; the connection weight matrix is as follows:

$$-\eta A^T A$$

$$= \begin{pmatrix} -1.2204 & -1.0479 & -0.7525 & -0.8567 & -1.0397 & -0.6689 \\ -1.0479 & -2.3984 & -1.5831 & -1.4696 & -1.4722 & -0.9716 \\ -0.7525 & -1.5831 & -1.1818 & -0.9189 & -1.0259 & -0.6241 \\ -0.8567 & -1.4696 & -0.9189 & -1.2569 & -1.0466 & -0.5600 \\ -1.0397 & -1.4722 & -1.0259 & -1.0466 & -1.3503 & -0.8105 \\ -0.6689 & -0.9716 & -0.6241 & -0.5600 & -0.8105 & -0.6299 \end{pmatrix}$$

$$\times 10^6.$$

Without loss of generality, let the initial voltages be zero; i.e., $v_{ij}(0) = 0$ for $i, j = 1, 2, \ldots, 6$. The steady-state matrix of the analog neural network based on ideal op-amps is shown as follows, where $R_f = 100\ \mathrm{M}\Omega$, $R = R_c = 10\ \mathrm{K}\Omega$, $R'_{ij} = 100\ \mathrm{K}\Omega$, and $C = 100\ \mu\mathrm{F}$:

$$\bar{V} = \begin{pmatrix} -0.042 & -1.000 & 1.119 & 0.080 & 1.421 & -0.600 \\ -1.368 & -2.772 & 0.500 & -0.484 & 2.989 & 1.627 \\ 1.149 & 3.455 & -0.979 & 0.434 & -1.515 & -1.954 \\ 2.129 & 1.111 & -0.939 & 0.612 & -2.164 & -0.587 \\ -1.345 & -0.579 & 1.370 & -1.927 & 0.583 & 1.806 \\ 0.933 & 2.116 & -1.410 & 2.992 & -3.302 & -1.089 \end{pmatrix}.$$

Figure 3 illustrates the transient behavior of the simulated analog neural network for this example. In this example, $\eta = 10^6$ and the eigenvalues of the weight matrix $\eta A^T A$ are $\{1288, 1880, 123, 3645, 7308, 66223\} \times 10^2$. The minimum of these eigenvalues is $1.23 \times 10^4$ and $5/1.23 \times 10^{-4} = 4.07 \times 10^{-4}$ seconds or 407 $\mu$s. As shown in Figure 3, the analog neural network can indeed reach its steady states in approximately 407 microseconds.

The accuracy and convergence rate of the analog neural network based real op-amps degrade to a degree that depends on the precision of the devices and selection of the parameters. For example, the steady-state matrix

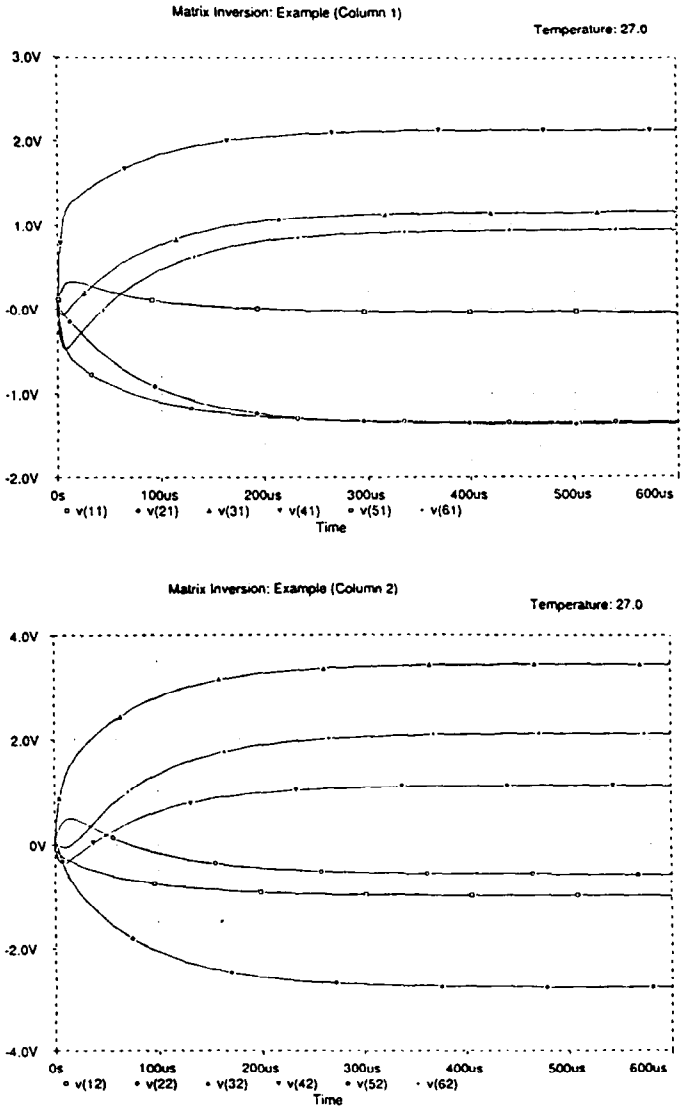



FIG. 3. Transient behavior of the simulated analog neural network.

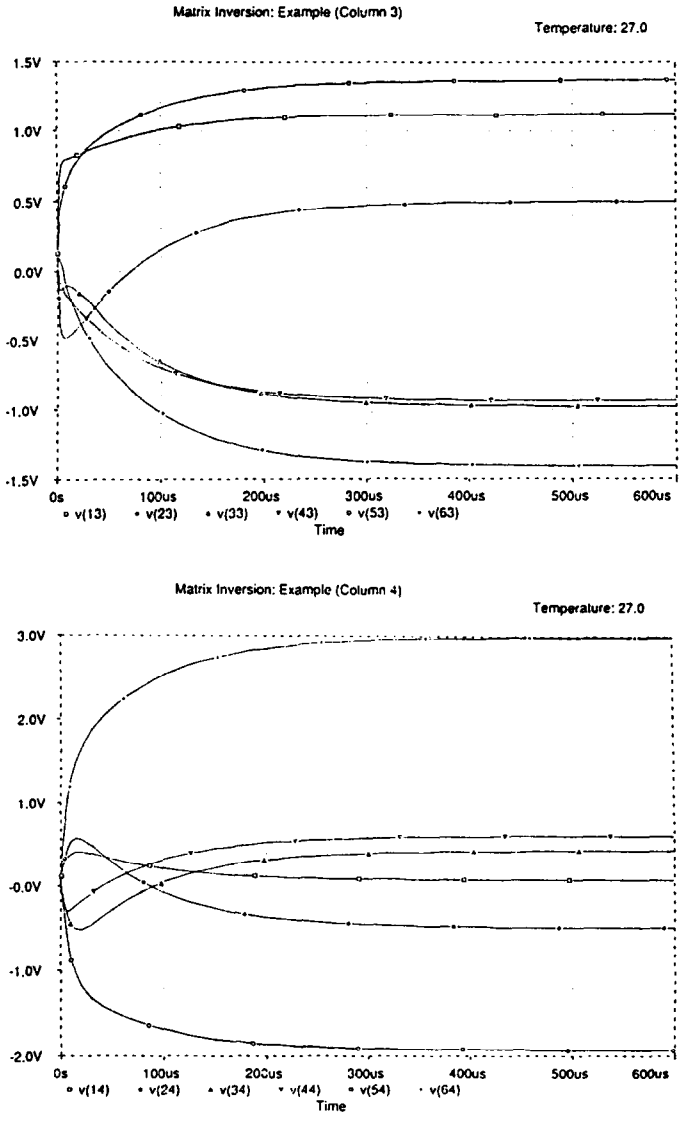
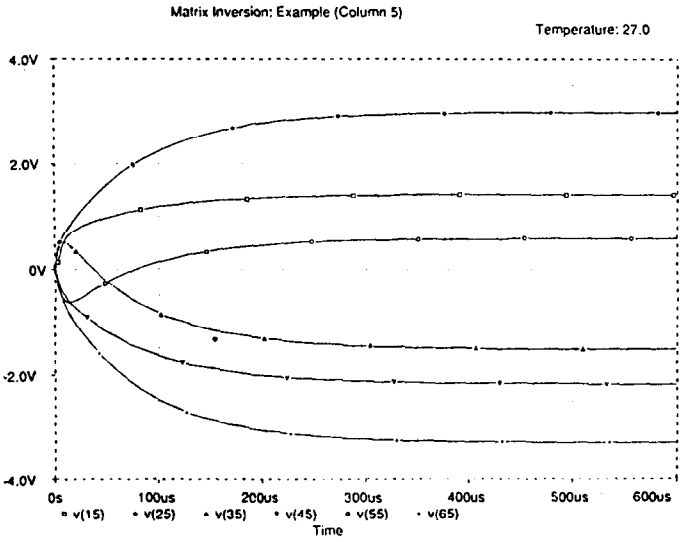



FIG. 3. (Continued).



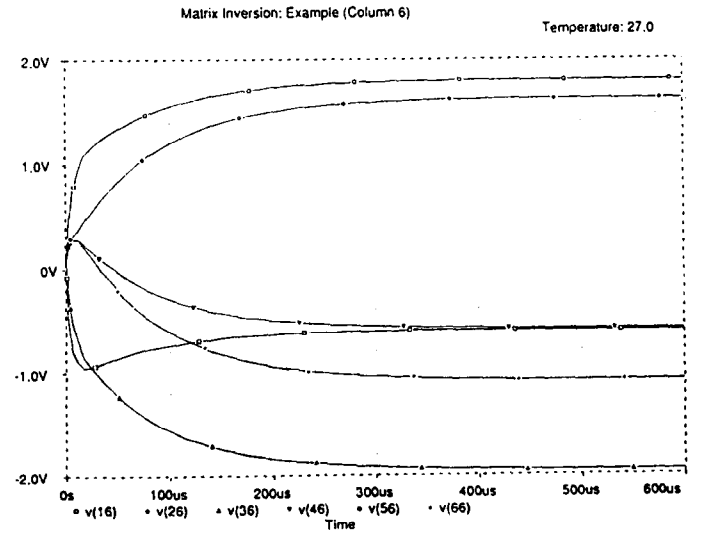




FIG. 3. (Continued).

of an analog based on LT1028 op-amps and the same parameter as the one based on ideal op-amps is shown as follows:

$$\bar{V} = \begin{pmatrix} -0.040 & -0.998 & 1.121 & 0.083 & 1.423 & -0.597 \\ -1.361 & -2.765 & 0.508 & -0.477 & 2.996 & 1.634 \\ 1.144 & 3.449 & -0.984 & 0.428 & -1.521 & -1.959 \\ 2.125 & 1.107 & -0.943 & 0.608 & -2.168 & -0.591 \\ -1.341 & -0.575 & 1.374 & -1.922 & 0.587 & 1.811 \\ 0.926 & 2.109 & -1.417 & 2.984 & -3.309 & -1.096 \end{pmatrix}.$$

## 6. CONCLUSIONS

The proposed recurrent neural network has been proven to be able to generate inverse matrices. Since the inversion process is inherently parallel and distributed, the convergence rate of the electronic neural network is independent of the order of the matrices. This feature renders the electronic neural network an advantage over traditional sequential procedures for large-scale matrix inversion. Since the convergence rate can be controlled by properly selecting the scaling contrast $\eta$, the proposed electronic neural network is also advantageous in real-time applications.

Further investigations based on the proposed neural network may be aimed at extension to computing pseudo-inverse matrices, applications of the proposed recurrent neural network to specific problems of interest, experimentation of the neural network using off-the-shelf components, and implementation of the neural network in analog VLSI circuits.

## REFERENCES

1 C. T. Chen, *Linear Systems Theory and Design*, Holt, Rinehart and Winston, New York, 1984.

2 J. Jang, S. Lee, and S. Shin, An optimization network for matrix inversion, in *Neural Information Processing Systems*, (D. Z. Anderson, Ed.), American Institute of Physics, New York, 1988, pp. 397–401.

3 T. Kailath, *Linear Systems*, Prentice Hall, Englewood Cliffs, New Jersey, 1980.

4 C. Koch, J. Luo, and C. Mead, Computing motion using resistive networks, in *Neural Information Processing Systems*, (D. Z. Anderson, Ed.), American Institute of Physics, New York, 1988, pp. 422–431.

5 F. L. Luo and B. Zheng, Neural network approach to computing matrix inversion, *Appl. Math. Comput.* 47:109–120 (1992).

6 S. I. Sudharsanan and M. K. Sundareshan, Equilibrium characterization of dynamical neural networks and a systematic synthesis procedure for associative memories, *IEEE Transactions on Neural Networks* 2:509–521 (1991).
7 S. I. Sudharsanan and M. K. Sundareshan, Exponential stability and a systematic synthesis of a neural network for quadratic minimization, *Neural Networks* 4:599–613 (1991).
8 J. Wang, Electronic realization of a recurrent neural network for solving simultaneous linear equations, *Electronics Letters* 28:493–495 (1992).