An Inverse Hollis–Paulos Artificial Neural Network

Louiza Sellami and Robert W. Newcomb, *Fellow, IEEE*

***Abstract*— The Hollis–Paulos artificial neural network (HPANN) is convenient in terms of its possibility for realization of variable weight artificial neural networks (ANN's) in very large scale integration (VLSI) by MOS transistor circuits, though it is nondynamical and not driven by external inputs. Here we introduce dynamics and inputs into the HPANN and show that over the range of operation covered by the Hollis–Paulos theory the system has an inverse. In particular, we derive that inverse, in semistate form, and give simulation results on its operation, showing how well the input to the original HPANN can be recovered from the output of the HPANN when fed into the inverse system. A comparison is made with the previous inverse of the Hopfield ANN. Possible applications of these inverse systems are to decoding of transmitted ANN signals and to inverse filtering for the extraction of input signals from processed signals.**

***Index Terms*—Dynamical neural networks, Hollis–Paulos neural networks, inverse neural networks, semistate theory.**

I. Introduction

The problem of inverse filtering (or deconvolution) has been studied for some time, at least since 1973, and several techniques (linear, nonlinear, iterative, noniterative) exist in the literature [1], [2]. These techniques have been widely used in applications involving seismic [3], radar [4], sonar [4], spectroscopic [5], instrumentation [6], and image configurations [2] where it is required to remove undesirable components from received signals. The advent of the neural-network paradigm generated a series of publications exploring the application of neural networks to solving deconvolution problems. Examples include the Widrow adaptive linear neuron (ADALINE) used to equalize frequency response and cancel echoes over data communication lines [7], deconvolution of finite impulse response (FIR) transversal filter via a Hopfield network [8], image and volume deblurring via cellular neural networks (CNN) [9], and image restoration via a multilayer perceptron using backpropagation and multilevel sigmoids [10]. Since a neural network modifies input data during processing, it could become important to "deconvolve" the neural network itself, i.e., find the inverse network to recover the input.

The Hollis–Paulos artificial neural network (HPANN) is convenient in terms of its possibility for realization of variable weight artificial neural networks (ANN's) in very large scale integration (VLSI) by MOS transistor circuits [11]. It is based on the Hopfield neuron model, though it is nondynamical and not driven by external inputs. The implementation uses MOSFET analog multipliers to construct weighted sums and permits asynchronous analog operation of Hopfield-type networks with fully programmable digital weights. In the basic Hopfield model the output is a nonlinear function of the weighted sum of the outputs over the entire set of neuron outputs. Hollis–Paulos's circuit implementation differs from Hopfield's network in one important aspect. Nonlinear limiting occurs at the input of each current-steering pair implementing analog neurons, whereas in the Hopfield model the limiting occurs at the output of the neurons. Because the continuous time Hopfield type network is general in its structure and contains most other common networks, such as that of Hollis–Paulos, as special cases upon simple modification, it is a good starting point for any theoretical development [12], [13]. Moreover, it is of great use to those interested in implementing electronic circuits based on neuro-biological architectures.

Starting in Section II with the previous inverse Hopfield network [13], a solution to the problem of approximating the input signal of a system via the construction of an inverse system using Hollis–Paulos artificial neural networks (HPANN) is developed. Toward this, we insert dynamics and inputs into the HPANN and create inverse neural networks for both the nondynamical and dynamical HPANN's. In each case, following [13], the general approach is to set up canonical semistate equations for the forward network, then solve for the inverse network to recover the input. We consider an HPANN with n interconnected neurons, n^2 weight operators, and n activation nonlinear functions. We assume that the outputs are available for observation in a given time interval and, consequently, we derive an inverse neural network to recover the inputs from the observed outputs. To support our theoretical development, we simulate an eight-neuron forward–inverse HPANN with dynamics via Simulink and demonstrate its capabilities and robustness.

II. Basic Inverse Network Theory

The basic principle of the inverse Hopfield network is to invert the semistate equations describing the forward network to solve for the input given the output [13]. The forward network equations are given in their canonical form as follows:

$$C\frac{dx}{dt} = W(v) - Rx + I_b + f(u), \qquad x(0) = x_o \tag{1}$$

$$W(v) = W \cdot v \tag{2}$$

$$v = g(x) \tag{3}$$

$$y = v \tag{4}$$

Manuscript received February 2, 1997; revised November 27, 1997 and May 18, 1998.

L. Sellami is with the Electrical Engineering Department, U.S. Naval Academy, Annapolis, MD 21402 USA. She is also with the Electrical Engineering Department, University of Maryland, College Park, MD 20742 USA.

R. W. Newcomb is with the Electrical Engineering Department, University of Maryland, College Park, MD 20742 USA.

Publisher Item Identifier S 1045-9227(98)05909-8.

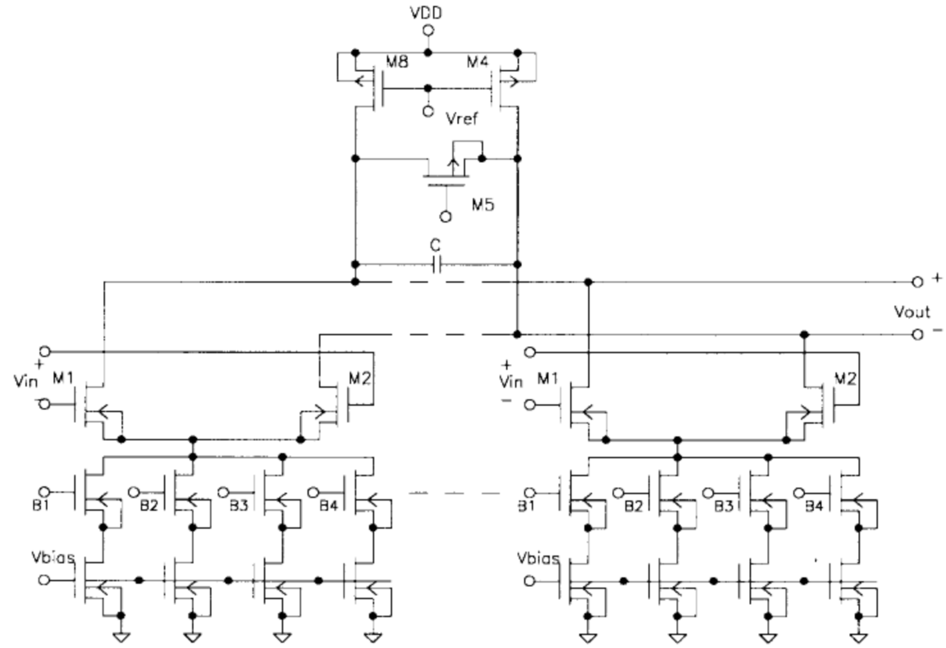


Fig. 1. Circuit schematic of the dynamical Hollis–Paulos current summing neuron with differential current-to-voltage converter. The dynamics are inserted by placing a capacitor between the drain and source of transistor M_5.

where u is the input vector, y the output vector, $[x^T, v^T]^T$ with T the transpose, the semistate vector, I_b a constant bias vector, $g(\cdot)$ the strictly monotonic nonlinear activation function vector, W the weight matrix, and C a nonsingular matrix introduced here to handle physical capacitors in a hardware implementation. In the Hopfield case, $f(\cdot)$ is the identity operator, but we introduce it here for later use in our modification to the Hollis–Paulos case.

Given $C, W, R, I_b, f(\cdot), g(\cdot), x_o$, and y over an interval of time, the inverse problem is to determine u over the same interval. The solution to this problem lies in the inverse system concept, which we develop below. First we rewrite (1) and (3) by solving for u and x

$$u = f^{-1}\left(C\frac{dx}{dt} + Rx - W(v) - I_b\right), \qquad x(0) = x_o \tag{5}$$

$$x = g^{-1}(v). \tag{6}$$

These, on defining $u_{\text{inv}} = y$ and $y_{\text{inv}} = u$, and incorporating (2) and (4) yield the inverse system description

$$y_{\text{inv}} = f^{-1}\left(C\frac{dx_{\text{inv}}}{dt} + Rx_{\text{inv}} - W(v_{\text{inv}}) - I_b\right)$$
$$x_{\text{inv}}(0) = x_o \tag{7}$$

$$W(v_{\text{inv}}) = W \cdot v_{\text{inv}} \tag{8}$$

$$x_{\text{inv}} = g^{-1}(v_{\text{inv}}) \tag{9}$$

$$u_{\text{inv}} = v_{\text{inv}} = y. \tag{10}$$

III. The Inverse Hollis-Paulos Network Without Dynamics

The basic Hollis–Paulos network uses no input, other than weight setting and initial conditions, and has no dynamics. Their design uses current steering MOS pairs with active loads to produce a neuron output voltage V_o given by (11)

$$V_o = R_L \sum_j (I_{D1} - I_{D2})_j \tag{11}$$

where

$$R_L = \frac{1}{K\lambda(V_{DD} - V_{\text{ref}} - V_{\text{th}})^2} \tag{12}$$

$$I_{D1} - I_{D2} = KV_i\sqrt{\frac{2I_o}{K} - V_i^2} \tag{13}$$

$$V_i = |V_{G1} - V_{G2}| \leq \sqrt{\frac{I_o}{K}} \tag{14}$$

Here the MOS transistor realization has $K = (1/2)\mu C_{\text{ox}} W/L, R_L$ as the output resistance, with λ being the channel modulation coefficient and V_{th} the threshold voltage, $I_o = I_{D1} + I_{D2}$ the differential pair tail current, and V_i the differential input voltage of the current steering pair limited to the range covered by (14).

Next, choosing the reference voltage to be $V_{\text{ref}} = V_{SS} = -V_{DD}$ to guarantee the transistors' operation in the saturation region, we set up the forward system by inserting an input vector u and rewriting the equations in a nondynamical semistate form, which defines i_d as follows:

$$y = -Ri_d \tag{15}$$

$$i_d = W(y) + I_b + f(u) \tag{16}$$

where i_d denotes a differential current vector, y the output vector, $R = [R_{jj}]$ a constant diagonal (nonsingular) gain matrix with R_{jj} representing the output resistance R_L of the jth load MOSFET device given in (12), I_b is a constant bias vector, W a nonlinear weight $n \times 1$ operator with the jth

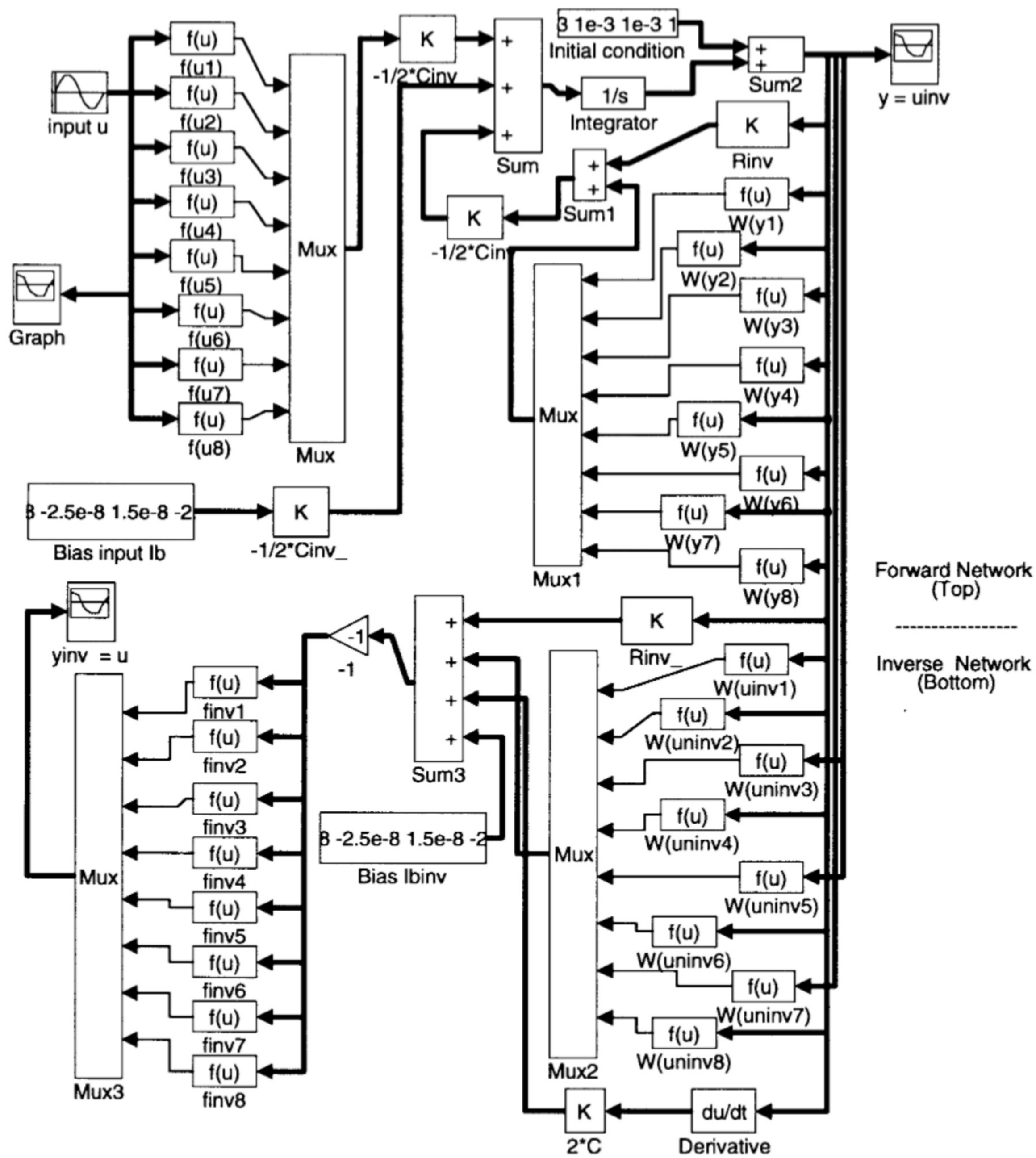


Fig. 2. Simulink block diagram of the (a) forward and (b) inverse Hollis–Paulos neural network with dynamics. The top Simulink diagram implements the dynamical forward network described by (27) where $R, W(\cdot)$ and $f(\cdot)$ are given by (12), (17), and (18). The input u is an array of eight signals which are separately fed through the activation function blocks $f(u_1), f(u_2), \cdots, f(u_8)$, then recombined by the Mux block. The bottom Simulink diagram implements the dynamical inverse network described by (28)–(31) where $f^{-1}(\cdot)$ is given by (21). The input $u_{\rm inv}$ to the inverse network is the output y of the forward network.

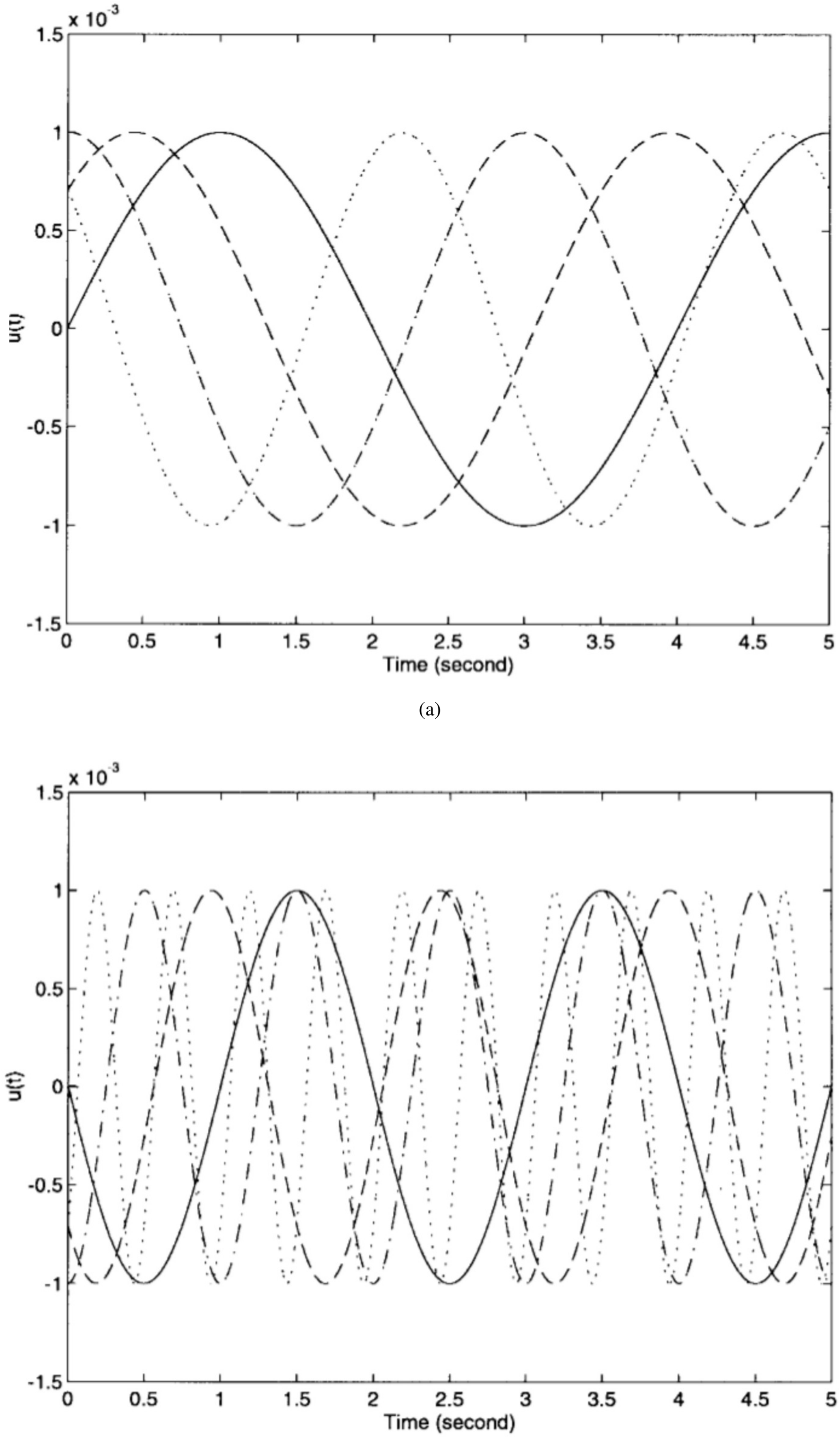
suboperator being

$$W_j(y) = K\sum_k y_k\sqrt{\frac{2I_{ojk}}{K} - y_k^2} \qquad \text{for} \quad y_k^2 \le \frac{I_o}{K}. \qquad (17)$$

Here I_{ojk} is the tail current of the kth differential pair forming the jth weight. By adding the input through the same form, that is using differential pairs, as Hollis and Paulos do for the weight output currents, $f(\cdot)$ becomes a square root operator, with the jth entry being

$$f_j(u_j) = Ku_j\sqrt{\frac{2I_0}{K} - u_j^2} \qquad \text{for} \quad u_j^2 \le \frac{I_o}{K}. \qquad (18)$$

Note that when the signals get too large, that is for $u_j^2 > I_o/K$, (17) and (18) need to be modified to saturate at I_o, as the differential pairs saturate at the tail currents, making the inverse network nonunique. To guarantee uniqueness, we restrict $f(\cdot)$ to be monotone by considering only the values of



(a)

(b)

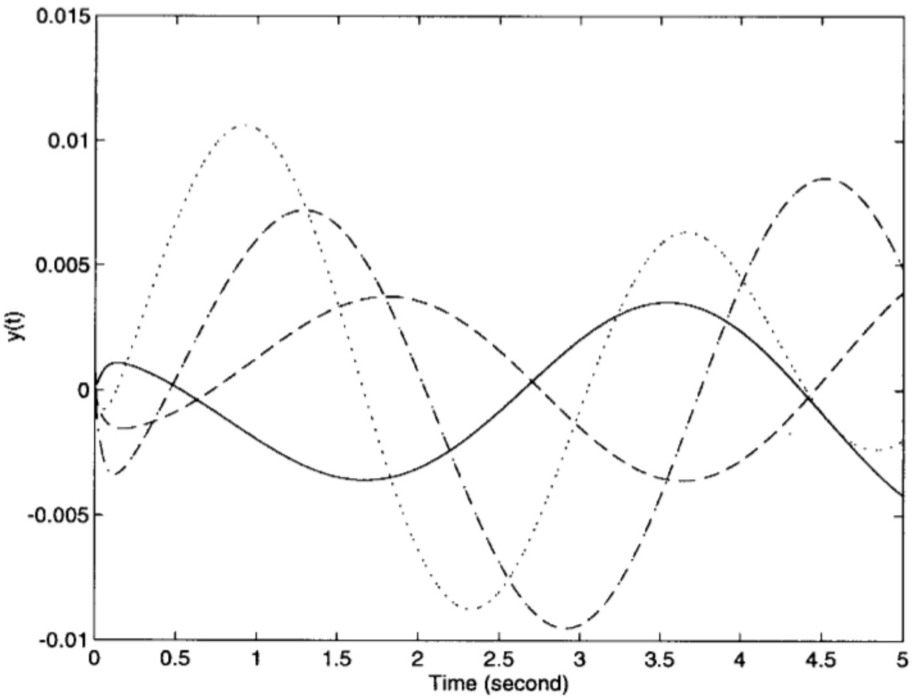
Fig. 3. The input signals to the forward network. (a) The first four input signals. (b) The last four input signals.

u satisfying $u^2 < I_o/K$. Furthermore, in (18) each K could be different by having different W/L for each of the MOS transistors, and so could I_o.

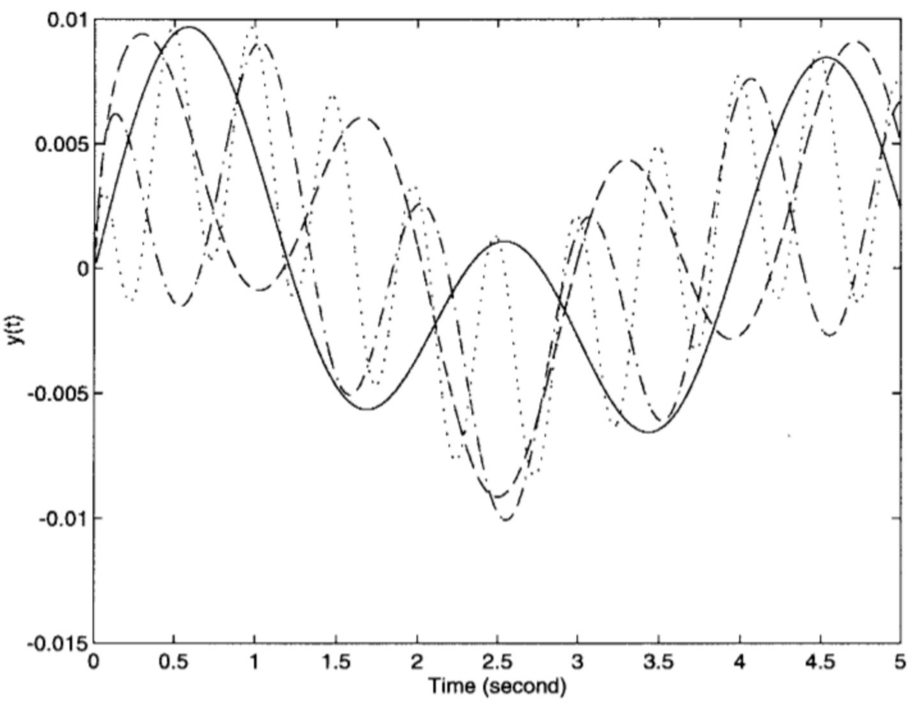
Now, the system described by (15) and (16) is readily inverted to find the u (in an explicit analytical form) that gives a received y

$$u = f^{-1}(i_d - W(y) - I_b) \tag{19}$$

$$= f^{-1}(-R^{-1}y - W(y) - I_b) \tag{20}$$



(a)



(b)

Fig. 4. The output signals of the forward network. (a) The first four output signals. (b) The last four output signals.

where $f^{-1}(\cdot)$ is given by (21), upon solving the quadratic equation of (18), $x = f(u)$, for u with components

$$f_j^{-1}(x_j) = \mathrm{sign}(x_j) \times \sqrt{\frac{I_o}{K}\left(1 - \sqrt{1 - \frac{x_j^2}{I_o^2}}\right)}. \qquad (21)$$

Thus, with $y_{\mathrm{inv}} = u, u_{\mathrm{inv}} = y$, and $g(\cdot)$ being the identity operator, the semisate description of the inverse network is

$$y_{\mathrm{inv}} = f^{-1}(-R^{-1}x_{\mathrm{inv}} - W(v_{\mathrm{inv}}) - I_b) \qquad (22)$$

$$W_j(v_{\mathrm{inv}}) = K \sum_k v_{\mathrm{inv}_k} \sqrt{\frac{2I_{ojk}}{K} - v_{\mathrm{inv}_k}^2}$$

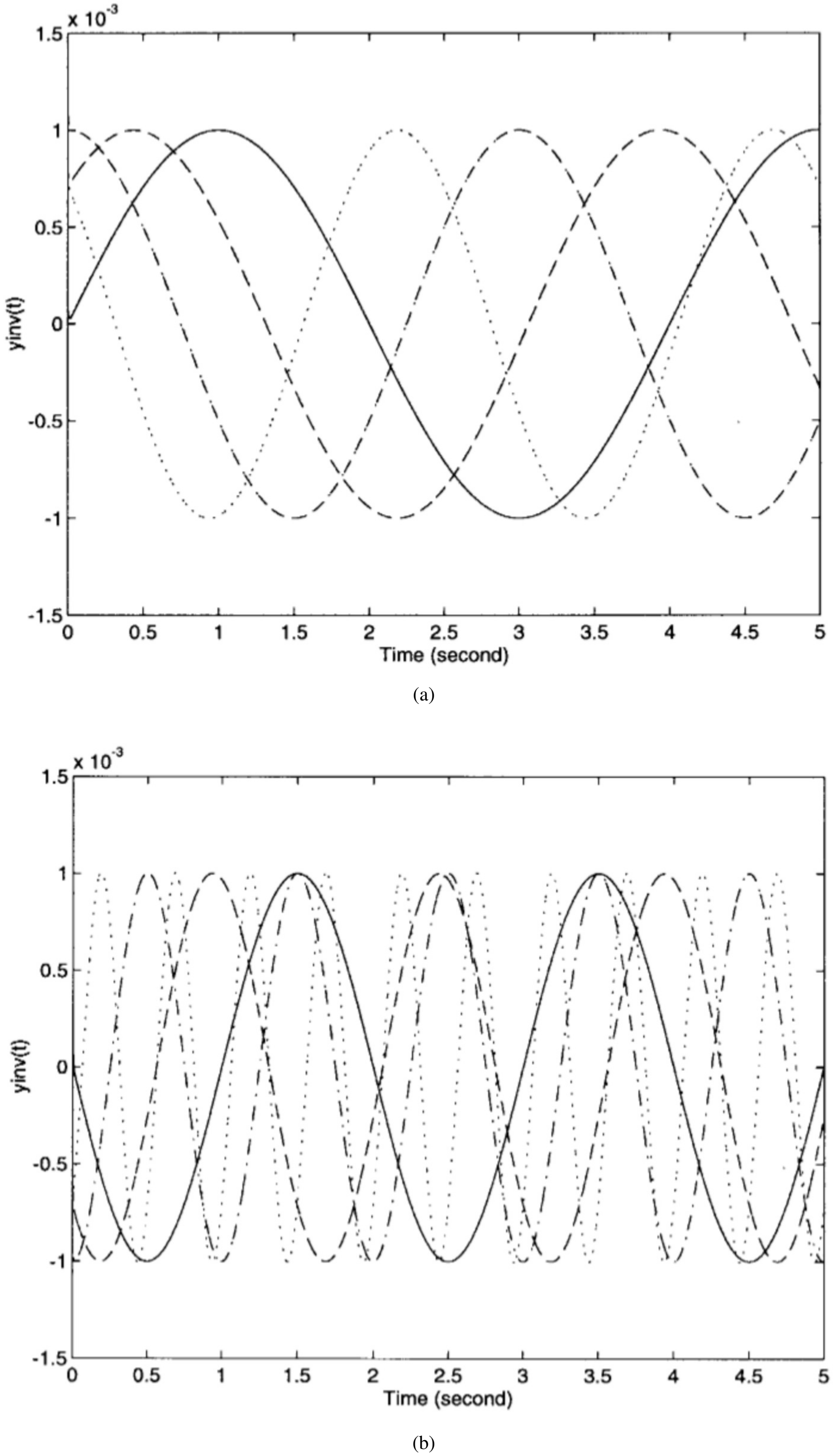


Fig. 5. The output signals of the inverse network. (a) The first four output signals. (b) The last four output signals.

for

$$v_{\mathrm{inv}_k}^2 \leq \frac{I_o}{K} \tag{23}$$

$$x_{\mathrm{inv}} = g^{-1}(v_{\mathrm{inv}}) = v_{\mathrm{inv}} \tag{24}$$

$$u_{\mathrm{inv}} = v_{\mathrm{inv}} = y. \tag{25}$$

IV. The Inverse Hollis–Paulos Network with Dynamics

In practical circuit implementations, the load transistors have parasitic capacitors which modify the behavior of the circuit; alternatively, one may purposely insert dynamics for smooth operation. To take this into account, we insert dynamics in the forward network by placing a capacitor across the output transistors, as shown by C in Fig. 1 (for one Hollis–Paulos neuron) and derive semistate equations for it. Writing the output voltage y in terms of the differential current i_d and the capacitor current i_c leads to

$$y = -R\left(i_d - 2C\frac{dy}{dt}\right) \tag{26}$$

which, on substituting the expression of i_d [given in (16)] yields

$$2C\frac{dy}{dt} = -W(y) - I_b - f(u) - R^{-1}y. \tag{27}$$

Here $W(\cdot)$ and $f(\cdot)$ have the same functional form as for the nondynamical case [see (17) and (18)]. The inverse system is obtained by solving for the input $u = y_{\text{inv}}$ that produced the observed forward output $y = u_{\text{inv}}$ from (27).

$$y_{\text{inv}} = f^{-1}\left(-W(v_{\text{inv}}) - I_b - R^{-1}x_{\text{inv}} - 2C\frac{dx_{\text{inv}}}{dt}\right) \tag{28}$$

$$W_j(v_{\text{inv}_k}) = K\sum_k v_{\text{inv}_k}\sqrt{\frac{2I_{ojk}}{K} - v_{\text{inv}_k}^2} \quad \text{for} \quad v_{\text{inv}_k}^2 \le \frac{I_o}{K} \tag{29}$$

$$x_{\text{inv}} = g^{-1}(v_{\text{inv}}) = v_{\text{inv}} \tag{30}$$

$$u_{\text{inv}} = v_{\text{inv}} = y \tag{31}$$

with $f^{-1}(\cdot)$ being given by (21). Here as at (18) we limit the values of u to those satisfying $u^2 < I_o/K$ to guarantee uniqueness of the inverse system.

V. Simulations and Results

To demonstrate the performance of the proposed theory and further illustrate its capabilities, simulation runs on a dynamical eight-neuron Hollis–Paulos network were done with Simulink, a package for use with Matlab for modeling, simulating, and analyzing dynamical systems. The Simulink block diagram of Fig. 2 shows both (a) the forward and (b) the inverse networks where the output y of the forward network is fed as the input to the inverse network. The top Simulink block diagram simulates the forward system described by (27) where $W(\cdot), f(\cdot)$, and R are as given by (17), (18), and (14). The capacitance matrix $C = 1\cdot 10^{-6}I_8$, where I_8 is the 8×8 identity matrix, and the bias input $I_b = [1.5\cdot 10^{-8}, -2.5\cdot 10^{-8}, 1.5\cdot 10^{-8}, -2.5\cdot 10^{-8}, 1.5\cdot 10^{-8}, -2.5\cdot 10^{-8}]^T$. The parameters used for evaluating $W(\cdot), f(\cdot)$, and R^{-1} $(= 3.92\cdot 10^{-5}I_8)$ are $\mu C_{\text{ox}} = 5.08\cdot 10^{-5}$, $W = 10\ \mu, L = 10\ \mu, V_{\text{th}} = 0.8582V, \lambda = 1.842\cdot 10^{-2}$, $I_o = 1\cdot 10^{-3}$, $V_{DD} = 5\ V_{\text{ref}} = -5$ V. The bottom Simulink block simulates the inverse network described by (28)–(31) where $f^{-1}(\cdot)$ and $W(\cdot)$ are given by (17) and (21). The input u to the forward network is an array of eight sine waves of amplitude 1 mV and periods 4, 3.5, 3, 2.5, 2, 1.5, 1, and 0.5 s, and phases $\pi/4, \pi/2, 3\pi/4, \pi, 5\pi/4, 3\pi/2$, and $7\pi/4$, respectively. Each input signal is separately fed through the nonlinear activation function $f(\cdot)$ then recombined by the MUX block to form an array of signals. The network is simulated with the initial condition $y_o = [1\cdot 10^{-3}, 1\cdot 10^{-3}, 1\cdot 10^{-3}, 1\cdot 10^{-3}, 1\cdot 10^{-3}, 1\cdot 10^{-3}, 1\cdot 10^{-3}, 1\cdot 10^{-3}]^T$ for a time interval of 5 s, using the Runge–Kutta 5 method with a step size between 0.000 01 and 0.001 and a tolerance of 10^{-10} as a stopping criterion to achieve a desired performance.

The plots of the forward input signals are given in Fig. 3, the resulting forward output signals are in Fig. 4, and the inverse output signals are in Fig. 5. It can be seen from Figs. 3 and 5 that the output of the inverse network is able to reproduce very closely the input of the forward network, except for an error of $<10^{-4}$ at the initial time for one of the signals. To further investigate the capabilities and robustness of the inverse network, we ran simulations with various values of tolerance, step size, bias input, and capacitance, as well as with a variety of input signals (constant, step function, square wave, and triangular wave) of different amplitudes, frequencies, and starting time. In particular, we simulated the network with values of C as low as 10^{-12} and frequencies as high as 1000 GHz. The results show that even when the activation function $f(\cdot)$ comes within 0.01% of saturating, the output of the inverse network still tracks the input of the forward system with an excellent accuracy, generally well under 0.01%. Note that these are simulation results which are used to validate the theory. If the system were realized with MOS transistors, operation at such high frequencies as 1000 GHz would not be expected due to the physical limitations of the devices.

VI. Conclusions

In this paper we have developed the semistate theory for an inverse Hollis–Paulos neural network for the nondynamical and dynamical cases into which external inputs are inserted. The mathematical formalism presented is based on setting up the canonical semistate equations for the forward network, then inverting them to derive the inverse network. We have shown, via computer simulations, that the suggested theoretical scheme provides an excellent technique for tracking the input signal for the dynamical HPANN. We note that this is for the case of a nonsaturated activation function, the case treated by Hollis–Paulos. However, in the case where there is saturation, there is no unique inverse network so that one needs to use signals that keep the system out of saturation for these, and the Hollis–Paulos, results to be useful. Further, we note that similar results have been achieved in the case of the Hopfield-type networks with sigmoidal type activation functions [13].

The Hollis–Paulos neural network is a circuit that uses analog multipliers to implement the weights. Consequently, it is appropriate for VLSI circuit design of neural-type microsystems which possess signal processing capabilities of physiological neural systems. Therefore, the semistate theory developed here could be useful for future VLSI implementations of the inverse network.

References

[1] G. H. Wing, *A Primer on Integral Equations of the First Kind: The Problem of Deconvolution and Unfolding*. Philadelphia, PA: Soc. Ind. Appl. Math., 1991.

[2] P. A. Jansson, Ed., *Deconvolution of Images and Spectra*, 2nd ed. San Diego: Academic, 1997.

[3] V. K. Arya and J. K. Aggarwal, Eds., *Deconvolution of Seismic Data*. Stroudsburg, PA: Hutchinson Ross, 1981.

[4] J. M. Mendel, *Maximum Likelihood Deconvolution: A Journey into Model-Based Signal Processing*. New York: Springer-Verlag, 1990.

[5] P. A. Jansson, Ed., *Deconvolution with Applications to Spectroscopy*. New York: Academic, 1984.

[6] K. B. Klaassen, *Electronic Measurement and Instrumentation*. New York: Cambridge Univ. Press, 1996.

[7] B. Widrow, J. R. Glover, Jr., J. M. McCool, J. Kaunitz, C. S. Williams, R. H. Hearn, J. R. Zeidler, E. Dong, Jr., and R. C. Goodlin, "Adaptive noise cancellating: Principles and applications," *Proc. IEEE*, vol. 63, pp. 1692–1716, 1975.

[8] F. L. Luo and Z. Bao, "Neural-network approach to adaptive FIR filtering and deconvolution problems," in *Proc. Int. Conf. Ind. Electron., Contr., Instrumentation*, Kobe, Japan, Oct. 28–Nov. 1, 1991, pp. 1449–1453.

[9] J. P. Miller, T. Roska, T. Sziranyi, K. R. Crounse, L. O. Chua, and L. Nemes, "Deblurring of images by cellular neural networks," in *Proc. 3rd Int. Wkshp. Cellular Neural Networks Applicat.*, Rome, Italy, Dec. 18–21, 1994, pp. 237–241.

[10] K. Sivakumar and V. B. Desai, "Image restoration using a multilayer perceptron with a multilevel sigmoidal function," *IEEE Trans. Signal Processing*, vol. 41, pp. 2018–2022, May 1992.

[11] P. W. Hollis and J. J. Paulos, "Artificial neural networks using MOS analog multipliers," *IEEE J. Solid-State Circuits*, vol. 25, pp. 848–855, June 1990.

[12] J. J. Hopfield, "Neurons with graded response have collective computational properties like those of two-state neurons," in *Proc. Nat. Academy Sci. USA*, vol. 81, no. 10, pp. 3088–3092, May 1984.

[13] A. Hodge, W. Zhen, and R. W. Newcomb, "On the inverse of Hopfield-type dynamical neural networks," in *Proc. 29th Asilomar Conf. Signals, Syst., Comput.*, Pacific Grove, CA, Oct. 30–Nov. 2, 1995, pp. 881–884.

Louiza Sellami received the Ingeniorat d'Etat from the Ecole Polytechnique d'Alger, Algeria, in 1984, and the M.S. and Ph.D. degrees from the University of Maryland, College Park, in 1988 and 1992, respectively.

Currently, she is an Assistant Professor in the Electrical Engineering Department at the U.S. Naval Academy. She is also a Visiting Research Fellow with the Microsystems Laboratory, and a special member of the graduate faculty at the University of Maryland, College Park. Her research interests include signal processing, circuits and systems theory, neural networks, and biomedical engineering.

Robert W. Newcomb (S'52–M'56–F'72–LF'97) was born in Glandale, California, in June 1933. He received the B.S.E.E. degree from Purdue University, West Lafayette, IN, in 1955, the M.S. degree in electrical engineering from StanfordUniversity, CA, in 1957, and the Ph.D. degree from the University of California, Berkeley, in 1960.

After serving on the tenured faculty at Stanford he moved to the University of Maryland to update the graduate program in electrical engineering. He has held visiting appointments in Australia, Belgium, Malaysia, Spain, Singapore, and most recently Korea. He directs the Microsystems Laboratory at the University of Maryland where emphasis is placed upon analog VLSI realization techniques. His present research includes biologically motivated neural networks, their VLSI realization, DSP for the ear, and general systems theory.