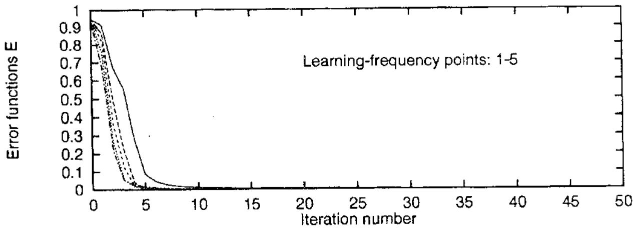

Fig. 2. Error functions E for five teacher-signal-frequency points versus iteration number of the learning process.

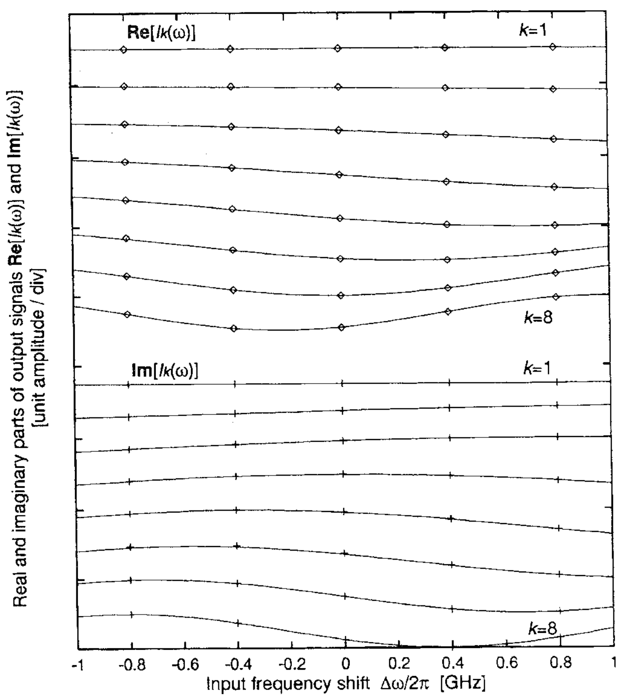

Fig. 3. Real and imaginary parts of the output signals of the neural system $\mathbf{Re}[I_k(\omega)]$ and $\mathbf{Im}[I_k(\omega)]$ versus input frequency shift $\Delta\omega$. Real and imaginary parts of teacher signals are also indicated by $\Diamond$ and $+$, respectively.

Fig. 2 shows the learning curves for five teacher-signal-frequency points. (They are dotted in Fig. 3 with $\Diamond$ and $+$ for real and imaginary parts, respectively.) The teacher signals are chosen to express various sinusoidal profiles in the frequency domain depending on the output-neuron index k. It is found in Fig. 2 that all the error functions reduce immediately. Fig. 3 shows the real and imaginary parts of output signals, $Re[I_k(\omega)]$ and $Im[I_k(\omega)]$, respectively, after the learning is completed. It is found that the output curves trace the teacher-signal points smoothly. The behavior of the neural network system is successfully controlled by the input-frequency modulation.

IV. Conclusion

Coherent-type artificial neural networks whose behavior is controlled by the carrier-frequency modulation have been proposed. The network learns teacher signals associated with the information-carrier frequency as a network parameter. The total network system forms a self-homodyne circuit. The learning process is realized by adjusting delay time and conductance of complex-valued neural connections. A simulation experiment has demonstrated that the network behavior is controlled by the frequency modulation successfully. This result will be applicable not only to signal processing but also to frequency-multiplexed optical neural computing and quantum neural devices such as carrier-energy-controlled neurons in the future.

References

[1] B. Widrow, J. McCool, and M. Ball, "The complex LMS algorithm," *Proc. IEEE*, vol. 63, pp. 719–720, 1975.

[2] G. M. Georgiou and C. Koutsugeras, "Complex-domain backpropagation," *IEEE Trans. Circuits Syst. II: Analog and Digital Signal Processing*, vol. 39, pp. 330–334, 1992.

[3] D. L. Birx and S. J. Pipenberg, "A complex mapping network for phase sensitive classification," *IEEE Trans. Neural Networks*, vol. 4, pp. 127–135, 1993.

[4] M. Takeda and T. Kishigami, "Energy function of complex neural fields generated by optical gain and feedback," in *Proc. SPIE*, vol. 1806, 1993, pp. 279–287.

[5] A. Hirose, "Applications of complex-valued neural networks to coherent optical computing using phase-sensitive detection scheme," *Inform. Sci. Applicat.*, vol. 2, pp. 103–117, 1994.

[6] T. Okoshi and K. Kikuchi, *Coherent Optical Fiber Communications*. Dordrecht, Holland: Kluwer, 1988.

Analog Decoding Using a Gradient-Type Neural Network

Iulian B. Ciocoiu

***Abstract*—The problem of analog (soft) decision decoding of block codes by means of neural networks is addressed. The proposed solution is based on a recurrent high-order network implementing a special gradient-type system. Simulation results for two different codes are reported, showing improved performances over the classical hard decision decoder.**

I. Introduction

Artificial neural networks have already been used with significant success in data transmission applications. Examples include channel equalization [1], code design [2], and decoding/error correction of data [3].

The present contribution addresses the problem of analog (sometimes called soft) decision decoding of block codes. To generate such a code the source encoder maps blocks of k binary information symbols into output codewords of n bits $(n > k)$. The mapping must be carefully chosen to improve the reliability of transmission. There are 2^k codewords to be selected from a set of 2^n possible combinations in a way that maximizes the Hamming distance between the chosen words (the Hamming distance between two binary numbers equals the number of positions they differ). If d is the minimum Hamming distance between any two codewords then it can be shown that up to e errors can be corrected if $d \geq 2e + 1$ [4]. A block code is usually denoted (n, k) and the ratio $R = (k \backslash n)$ is called the code rate.

Let us note that when the channel input and output alphabets are identical the channel is called a hard decision channel and when

Manuscript received August 31, 1993; revised February 21, 1996.

The author is with the Faculty of Electronics and Telecommunications, Technical University Iasi, 6600 Iasi, Romania.

Publisher Item Identifier S 1045-9227(96)04383-4.

the output alphabet is larger (possibly continuous) it is called a soft decision channel. The decoding strategy in the case of hard decision channels usually consists in thresholding the received message, followed by an algebraic decoding/error correcting algorithm (the threshold is set at half distance between the signalling levels corresponding to the binary alphabet). Soft decision decoding acts directly on the analog distorted noisy received message. Soft decision decoding has much better performances over the hard decision one in terms of post-decoding bit-error rate (BER), but soft decoding algorithms for most block codes do not exist. The reason for using a neural network to perform (soft decision) decoding will be explained by means of a state-space approach. Analog recurrent neural networks are, in fact, examples of multidimensional continuous nonlinear dynamical systems. When certain conditions are met the dynamics evolve from any initial state towards one particular stable equilibrium point, and no other complex behavior can occur. Such systems are called globally stable and are usually analyzed by means of the second method of Lyapunov [5]. The idea is now straightforward: the received distorted noisy analog codeword will act as an initial condition for such a (neural) dynamical system which will eventually settle down to one of the stable equilibrium points which should be predefined to coincide with the correct versions of the codewords used by the source, hopefully to the closest in terms of Hamming distance.

II. The Proposed Architecture

There are some requirements the system should meet to reliably perform the decoding and error correcting tasks [6]:

- each codeword should be stored as a point in a multidimensional state-space where the Lyapunov function of the system has a minimum;
- the shape of the basin of attraction around such a point should be controllable;
- the number of spurious states (stable states which do not correspond to desired stable equilibria) must vanish;
- the number of desired stable equilibrium points should be arbitrarily large;
- the addition/elimination of a particular equilibrium should be performed without redesigning the whole system.

The present approach makes use of a gradient type system [5], specially designed to meet the conditions listed above

$$\frac{dx_i}{dt} = -\frac{\partial V(X)}{\partial x_i},\ i = 1 \cdots N \tag{1}$$

where $V(\mathbf{X})$ is the so-called Lyapunov function (that is a positive definite function having negative time derivative along a solution trajectory in the state space) and N is the order of the system. A well-known result states that all isolated minima of $V(\mathbf{X})$ are asymptotically stable states of system (1) [5].

Our intention is to have these minima placed in predefined positions, corresponding to the correct codewords of a specific block code. To do so, we shall use an idea previously presented in [7] to solve a pattern recognition problem, namely we construct our Lyapunov function as a sum of individual functions exhibiting good space localization properties, having deep minima at the desired locations and been practically constant in rest

$$V(X) = \sum_{s=1}^{M} g_s(X) \tag{2}$$

where M is the number of codewords to be stored and $g_s(\mathbf{X})$ are functions satisfying the requirement above. The basic idea is to choose the distance between the current position in the state space and one of the desired equilibria as the argument of function g, so we will be interested in finding appropriate G_s functions with sharp minima at the origin

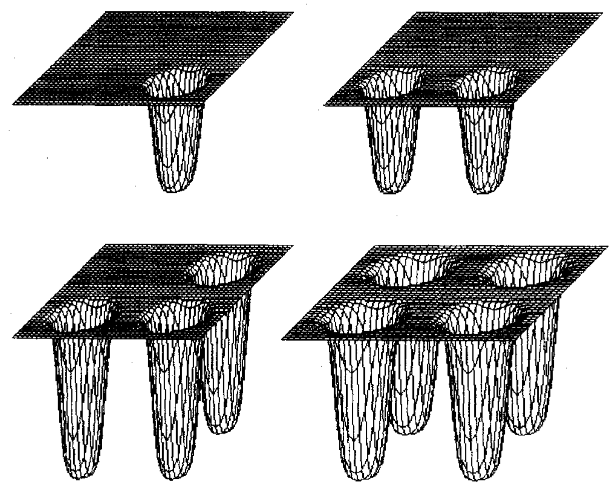

Fig. 1. Example of the special Lyapunov function ($M = 4$ and $N = 2$).

$$V(X) = \sum_{s=1}^{M} G_s(d_p(X, X^s)) \tag{3}$$

where d_p, $p \in N$ is the distance induced by the L_p measure defined on the multidimensional vector space to which the codewords belong and $\mathbf{X}^s$ is a codeword to be stored. There are many ways to adopt the functions G_s and the distance d_p. The choice should take into consideration the specific communication environment (that is the modulation scheme, the characteristics of the transmission channel, the nature of the noise) and should offer implementation advantages.

We decided to use a Gaussian type function and two different distance measures, the Euclidean ($p = 2$) and Manhattan ($p = 1$) distance

$$G_s(X) = 1 - e^{-\frac{d_p^p(X, X^s)}{2\sigma_s^2}}. \tag{4}$$

Besides its remarkable space selectivity this choice for function G_s is strongly motivated by the implementation advantages it offers (related to the fact that it is factorizable) [8].

Remark: It is worth noting that (3) using a Gaussian-type function can be regarded as a special radial basis functions (RBF's) expansion of the Lyapunov function $V(\mathbf{X})$ [9].

To offer an intuitive idea of the proposed approach we present in Fig. 1 an example of a Lyapunov function synthesized by using relation (3) for a system of order $N = 2$ with $M = 4$ stable equilibrium points, namely: $(-1, -1)$; $(-1, 1)$; $(1, -1)$; $(1, 1)$. Function G_s from (4) was employed, with $\sigma_s = 0.5$.

In Table I we present the equations describing the dynamical systems for the two types of distances where, for simplicity, all σ_s are considered equal. Special care must be taken in the case of Manhattan ($p = 1$) distance, since this function is not differentiable. A proper approximation for the modulus function has been used

$$f(x) = \frac{1}{\alpha} \ln(\cosh(\alpha x));\ f'(x) = \tanh(\alpha x) \tag{5}$$

where parameter α controls the slope of the derivative around origin (in Table I the $|\cdot|$ symbol for the argument of the exponential was maintained for the ease of notation).

The relation between the minima of the Lyapunov function in (3) and those corresponding to individual functions G_s is analyzed by

TABLE I
EQUATIONS DESCRIBING THE GRADIENT-TYPE SYSTEMS FOR EUCLIDEAN ($p = 2$) AND MANHATTAN ($p = 1$) DISTANCE MEASURES

Distance:	Dynamic system equations:
$d_2(X,X^s) = \sum_{i=1}^{N} (x_i - x_i^s)^2$	$\frac{dx_i}{dt} = -\frac{\partial V}{\partial x_i} = -\frac{\partial}{\partial x_i}\left[\sum_{s=1}^{M}(1-e^{\frac{-\sum_{j=1}^{N}(x_j-x_j^s)^2}{2\sigma^2}})\right]$ $= -\frac{1}{\sigma^2}\sum_{s=1}^{M}(x_i-x_i^s)e^{\frac{-\sum_{j=1}^{N}(x_j-x_j^s)^2}{2\sigma^2}}$ $= -\frac{x_i}{\sigma^2}\sum_{s=1}^{M} e^{\frac{-\sum_{j=1}^{N}(x_j-x_j^s)^2}{2\sigma^2}} + \frac{1}{\sigma^2}\sum_{s=1}^{M} x_i^s e^{\frac{-\sum_{j=1}^{N}(x_j-x_j^s)^2}{2\sigma^2}}$
$d_1(X,X^s) = \sum_{i=1}^{N} \lvert x_i - x_i^s \rvert$	$\frac{dx_i}{dt} = -\frac{\partial V}{\partial x_i} = -\frac{\partial}{\partial x_i}\left[\sum_{s=1}^{M}(1-e^{\frac{-\sum_{j=1}^{N}\lvert x_j-x_j^s\rvert}{2\sigma^2}})\right]$ $= -\frac{1}{2\sigma^2}\sum_{s=1}^{M}\tanh\left[\alpha(x_i-x_i^s)\right]e^{\frac{-\sum_{j=1}^{N}\lvert x_j-x_j^s\rvert}{2\sigma^2}}$ ----

means of a simple example similar to that presented in [7]. Let us consider the equation corresponding to the Euclidean distance measure ($p = 2$) from Table I. The equilibrium points (both stable and unstable) are obtained by imposing $dx_i/dt = 0$, for all $i = 1 \cdots N$. Let $\mathbf{X}^s$ be the individual minimum introduced by function G_s and let restrict our analysis to a small neighborhood around it. If we denote by $d_{\min}$ the minimum distance between the codewords to be stored, we may write $\|\mathbf{X} - \mathbf{X}^t\| \geq d_{\min}$, for every $\mathbf{X}^t \neq \mathbf{X}^s$. For the two block codes which were used in the simulations, we have $d_{\min} = 3$ and $M = 2^4$ (since we have $k = 4$ information bits in both cases). Choosing $d_{\min}/\sigma \geq 6$ we get

$$\frac{dx_i}{dt} = 0 \Leftrightarrow (x_i - x_i^s)e^{\frac{-\sum_{j=1}^{N}(x_j - x_j^s)^2}{2\sigma^2}} + \sum_{\substack{t=1 \\ t\neq s}}^{M}(x_i - x_i^t)e^{\frac{-\sum_{j=1}^{N}(x_j - x_j^t)^2}{2\sigma^2}}$$

$$= 0 \Leftrightarrow (x_i - x_i^s)e^{\frac{-\sum_{j=1}^{N}(x_j - x_j^x)^2}{2\sigma^2}} + 10^{-2} = 0$$

which shows that the minima of the Lyapunov function practically coincide with those of the individual functions G_s when imposing restrictions on the values of the σ parameter. Moreover, it is easy to see that the number of desired minima can be arbitrarily large if σ is sufficiently small. In Table II we give the actual minima of the function $V(\mathbf{X})$ corresponding to several codewords selected from the (7,4) Hamming code, for different values of σ.

The problem of the extension of the basin of attraction around such a stable equilibrium point is a more subtle one. We only cite here an efficient constructive method for estimating it proposed in [10]. To use it is somehow simpler in the case of gradient-type systems, since the method requires as a starting point the availability of a Lyapunov function.

In Fig. 2 the block diagram of the system for the Euclidean distance is presented. The code dependent linear combiner outputs factors of the form

$$\frac{1}{\sigma^2}\sum_{s=1}^{M} x_i^s e^{\frac{-\sum_{j=1}^{N}(x_j - x_j^s)^2}{2\sigma^2}} \tag{6}$$

which represent the sums of its (Gaussian) inputs corresponding to nonzero x_i^s elements. There are N such outputs. The output denoted by "0," which is common to all N cells, implements a special sum of the above type, namely the one for all x_i^s equal to one. The issue of distance computing cell has been addressed in [8]. The presence of multipliers is still an important problem, which can be alleviated by using the following approximation for the derivative of the Gaussian function:

$$\frac{d}{dx}G(x) = \frac{d}{dx}\left[e^{-\frac{x^2}{2\sigma^2}}\right] = -\frac{x}{\sigma^2}e^{-\frac{x^2}{2\sigma^2}} \simeq G\left(x + \frac{\sigma}{2}\right) - G\left(x - \frac{\sigma}{2}\right). \tag{7}$$

TABLE II
RELATION BETWEEN SEVERAL CODEWORDS FROM THE (7,4) HAMMING CODE AND THE ACTUAL MINIMA OF THE LYAPUNOV FUNCTION IN (3) (EUCLIDEAN DISTANCE)

Codewords:	0000000	1101000	1111111
$\sigma^2 = 0.5$	0.5, 0.5, ... 0.5	0.5, 0.5, ... 0.5	0.5, 0.5, ... 0.5
$\sigma^2 = 0.33$	0.057, 0.057, ... 0.057	0.92, 0.92, 0.07, 0.92, 0.07, 0.07, 0.07	0.92, 0.92, ... 0.92
$\sigma^2 = 0.25$	0.009, 0.009, ... 0.009	0.99, 0.99, 0.009, 0.99, 0.009, 0.009, 0.009	0.99, 0.99, ... 0.99

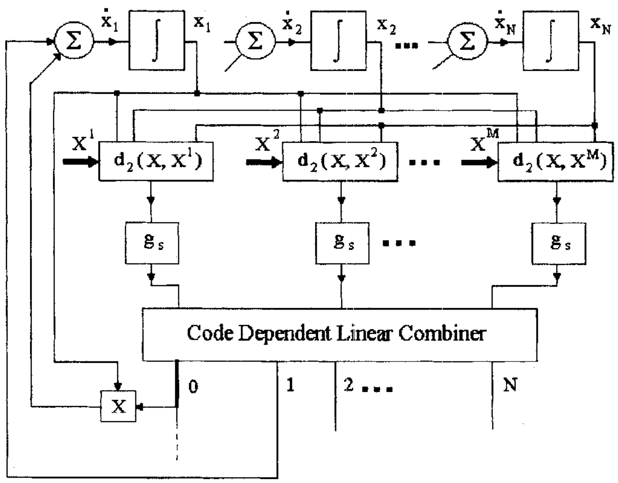

Fig. 2. Block diagram of the network (Euclidean distance measure).

The multipliers can be eliminated by slightly complicating the argument of the exponential.

III. COMPUTER SIMULATIONS

To test the efficiency of the proposed approach intensive computer simulations have been performed using two different block codes, namely (7,4) Hamming code and (7,4) cyclic redundancy code. They both are one error correcting codes. The decoding performances are usually analyzed in terms of the post-decoding BER versus the signal-to-noise ratio per information bit E_b/N_0. The last term strongly depends on the modulation scheme and the type of channel which is used. For additive white Gaussian noise (AWGN channel), baseband unipolar transmission and for optimal demodulator (matched filter followed by a sampler) the ratio E_b/N_0 can be expressed as a function of the signal amplitude A_s and the noise variance σ_n^2 as [4]

$$\frac{E_b}{N_0} = \frac{n}{k}\frac{A_s^2}{8\sigma_n^2}. \tag{8}$$

The factor n/k indicates that E_b is the energy per information bit and not per channel symbol.

Many simulations have been performed by selecting one of the codewords and adding Gaussian noise with zero mean and σ_n^2 variance to each bit then delivering this analog vector to the neural network, as an initial condition. The network will converge towards one of the predefined equilibrium points (which practically coincide with the correct codewords), ideally to the closest in terms of Hamming distance.

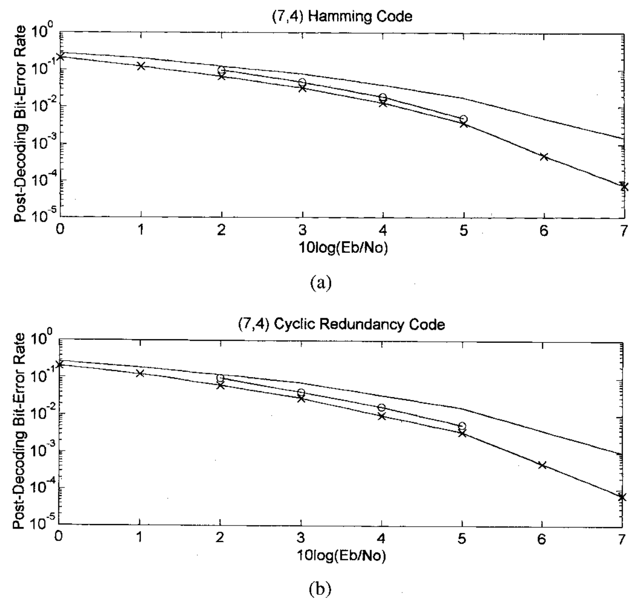

Fig. 3. (a) Simulation results for the (7,4) Hamming code ($\sigma = 0.5$): $x - d_2$ network; $o - d_1$ network ($\alpha = 50$); solid line—using the hard-decision decoder. (b) Simulation results for the (7,4) cyclic redundancy code ($\sigma = 0.5$): $x - d_2$ network; $o - d_1$ network ($\alpha = 50$); solid line—using the hard-decision decoder.

The results are presented in Fig. 3(a) for the (7,4) Hamming code and in Fig. 3(b) for the (7,4) cyclic redundancy code. For comparison, the results obtained by using a standard hard decision decoder are also presented. It is obvious that the soft decision neural decoders perform better than the hard one, which illustrates in fact a well-known principle. The post-decoding bit error rate is one order of magnitude smaller for the neural decoder using the Euclidean distance at 7 dB signal-to-noise ratio for the (7,4) Hamming code and slightly better for the cyclic code. In all simulations we have considered $A_s = 1$.

In Fig. 4 typical dynamic trajectories are shown. In Fig. 4(a) only one component of the initial state vector exceeds the threshold set at 0.5 (half distance between the two signaling levels) and the final state of the system is the correct $(0 \cdots 0)$ vector. The same result would

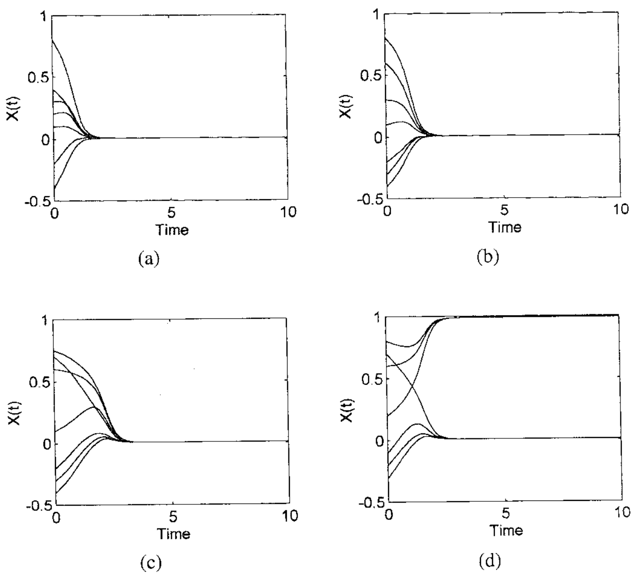

Fig. 4. Typical dynamic trajectories for d_2 network ($X(t) = \{x_i(t)\}$, $i = 1 \cdots 7$): (a) one component of the initial state vector exceeds the threshold value; (b) two components of the initial state vector exceed the threshold value; (c) three components of the initial state vector exceed the threshold value; (d) convergence towards a wrong stable state.

had been obtained if a hard-decision decoder have been used since after thresholding the Hamming distance between the resulting word and the correct one would have been one (and the (7,4) Hamming code is able to correct up to one error). Anyway, for the trajectories presented in Fig. 4(a) and (c) the resulting Hamming distance after thresholding would be two, respectively, three, and the hard decoder would not be able to reconstruct the correct codeword. In Fig. 4(d) convergence to a wrong codeword is shown. These graphics indicate that the basin of attraction of a specific codeword may include states that, after thresholding, give binary words whose Hamming distances from it exceed one (which is the limit for the hard-decision decoder to operate well).

IV. Conclusions

This paper presents a novel solution for the implementation of soft decision decoders by using a special gradient type neural network. The synthesis of the network relies on the construction of a particular Lyapunov function. The proposed approach has the following advantages:

- the system needs no learning phase, since it is a hardwired one;
- the key elements of the architecture have already been implemented in VLSI (very large scale integration) structures. IBM has recently produced a neural chip including Gaussian-type functions and distance computing blocks [11];
- the system can be used with minor modifications for decoding m-ary codes (where m is the number of signaling levels, $m \geq 2$).

Using the distance between $\mathbf{X}$ and $\mathbf{X}^s$ as the argument for the space selective function G_s is only sufficient, not necessary. In fact, the argument has to be positive and vanishes only when the current state of the network equals one of the stored codewords. For binary codewords other solutions could prove more useful for implementation, for example

$$P(X, X^s) = \sum_{i=1}^{N}[x_i(1 - x_i^s) + x_i^s(1 - x_i)] \tag{9}$$

which does not represent a properly defined distance in R^n (since $P(\mathbf{X}, \mathbf{X}^s)$ is zero for $\mathbf{X} = \mathbf{X}^s$ only when $\mathbf{X}$ and $\mathbf{X}^s$ are both binary, in which case it gives the Hamming distance between the two vectors). This choice could be related to a paper by Saha *et al.* where oriented nonradial-basis functions (ONRBF's) are introduced [12].

It is worth noting that the presence of any nonlinearities other than G_s should be avoided, since they could introduce supplementary stable states which would degrade the performance of the proposed solution. If operational amplifiers are used to implement the integrators and the adders they should operate in the linear region.

There are some problems to be further investigated, for example making the system a learning one by setting the σ parameters in an adaptive manner. This feature would allow the system to cope with the (possibly nonstationary) communication environment and, more generally, to work properly when the equilibrium points are clustered.

The reported results obtained through simulation are believed to be even better for more complicated block codes. To acquire the full benefit of the proposed solution, it should be hardware implemented.

References

[1] T. Kohonen *et al.*, "Performance evaluation of self-organizing map-based neural equalizers in dynamic discrete-signal detection," in *Artificial Neural Networks*, T. Kohonen, K. Makisara, O. Simula, and J. Kangas, Eds. New York: Elsevier, 1991, pp. 1677–1680.

[2] J. C. Platt and J. J. Hopfield, "Analog decoding using neural networks," in *Proc. AIP Conf.*, Snowbird, UT, 1986, pp. 364–369.

[3] C. Jeffries, "High-order neural models for error correcting code," in *Proc. SPIE Conf. Appl. Artificial Neural Networks*, Orlando, FL, 1990, pp. 510–517.

[4] R. E. Ziemer and R. L. Peterson, *Introduction to Digital Communication.* New York: Macmillan, 1992.

[5] M. W. Hirsch and S. Smale, *Differential Equations, Dynamical Systems, and Linear Algebra.* New York: Academic, 1974.

[6] A. N. Michel and J. A. Farrell, "Associative memories via artificial neural networks," *IEEE Contr. Syst. Mag.*, vol. 10, pp. 6–17, Apr. 1990.

[7] J. Y. Han, M. R. Sayeh, and J. Zhang, "Convergence and limit points of neural networks and its application to pattern recognition," *IEEE Trans. Syst., Man, Cybern.*, pp. 1217–1222, Sept.-Oct. 1989.

[8] S. Collins, G. F. Marshall, and D. R. Brown, "An analog radial basis function circuit using a compact Euclidean distance calculator," in *Proc. IEEE ISCAS'94*, London, pp. 233–236.

[9] T. Poggio and F. Girosi, "Networks for approximation and learning," *Proc. IEEE*, vol. 78, no. 9, pp. 1481–1497, 1990.

[10] H. D. Chiang and J. S. Thorp, "Stability regions of nonlinear dynamical systems: A constructive methodology," *IEEE Trans. Automat. Contr.*, vol. 34, pp. 1229–1241, Dec. 1989.

[11] IBM Microelectronics, ZISC 036, preliminary inform., April 1994.

[12] A. Saha, J. Christian, D. S. Tang, and C.-L. Wu, "Oriented nonradial basis functions for image coding and analysis," in *Advances in Neural Information Processing Systems 3*, R. P. Lippmann, J. E. Moody, and D. S. Touretzky, Eds. San Mateo: Morgan Kaufmann, 1991, pp. 728–734.