

# Adaptive Near-Optimal Multiuser Detection Using a Stochastic and Hysteretic Hopfield Net Receiver

## Gábor Jeney

*Mobile Communications Laboratory, Department of Telecommunications, Budapest University of Technology and Economics, Budapest, Hungary*  
Email: [jeneyg@hit.bme.hu](mailto:jeneyg@hit.bme.hu)

## János Levendovszky

*Signal Processing Laboratory, Department of Telecommunications, Budapest University of Technology and Economics, Budapest, Hungary*  
Email: [levendov@hit.bme.hu](mailto:levendov@hit.bme.hu)

## László Pap

*Mobile Communications Laboratory, Department of Telecommunications, Budapest University of Technology and Economics, Budapest, Hungary*  
Email: [pap@hit.bme.hu](mailto:pap@hit.bme.hu)

## E. C. van der Meulen

*Department of Mathematics, Katholieke Universiteit Leuven, Leuven, Belgium*  
Email: [ecvdm@gauss.wis.kuleuven.ac.be](mailto:ecvdm@gauss.wis.kuleuven.ac.be)

Received 31 January 2002 and in revised form 20 June 2002

This paper proposes a novel adaptive MUD algorithm for a wide variety (practically any kind) of interference limited systems, for example, code division multiple access (CDMA). The algorithm is based on recently developed neural network techniques and can perform near optimal detection in the case of unknown channel characteristics. The proposed algorithm consists of two main blocks: one estimates the symbols sent by the transmitters and the other identifies each channel of the corresponding communication links. The estimation of symbols is carried out either by a stochastic Hopfield net (SHN), by a hysteretic neural network (HyNN) or by both. The channel identification is based on either the self-organizing feature map (SOM) or the learning vector quantization (LVQ). The combination of these two blocks yields a powerful real-time detector with near optimal performance. The performance is analyzed by extensive simulations.

**Keywords and phrases:** CDMA system, adaptive detection, recurrent neural network, self-organizing maps, learning vector quantization.

## 1. INTRODUCTION

Recently, multiuser detection (MUD) has gained much attention in the world of telecommunication research. The claim for MUD primarily arises in systems suffering from the limitation of interference, such as code division multiple access (CDMA) which has been adopted as the main multiple access method of the third generation universal mobile telecommunication system (UMTS). Without novel MUD techniques, conventional receiver structures suffer severe performance degradation in high bitrate applications [1, 2].

MUD carries out joint detection for a group of users or single-user detection for a specific user in the presence of other users in the channel. In the 90's, a large number of articles were published focusing on this field. Many different approaches to MUD have been proposed [1] (e.g., some authors regard this field as a task of joint detection, others implement signal processing methods to get rid of unwanted interference, while a third group of authors still regard it as a classification or hypothesis testing problem). Nevertheless, we should keep in mind that the purpose of MUD is to provide a robust, low cost, reliable, and fast method to separate signals arriving from different sources over the same

medium. In this paper, we propose an alternative method which unites the fast convergence property of hysteretic neural networks (HyNN) [3] with the optimization power of stochastic recurrent neural networks (SHN) [4]. Furthermore, an adaptive detection technique is applied in the neural network receiver, though still including the existing RAKE receiver structure in the detector.

This paper is organized as follows: in Section 2, the applied baseband equivalent system model is introduced; in Section 3, some neural network-based MUD techniques are discussed; in Section 4, some novel multiuser detector structures are introduced based on the modified recurrent neural network model; in Section 5, the applied adaptation technique is demonstrated; in Section 6, the structure of the proposed adaptive multiuser detector is described; in Section 7, the performance of the novel algorithm is analyzed by extensive simulations; and finally in Section 8, its computational complexity is discussed focusing on real time applications.

## 2. SYSTEM MODEL

One of the major attributes of CDMA systems is the multiple usage of the same frequency band and time slot. Although the system is susceptible to multiple access interference, theoretically the users are not jammed by each other due to the uncorrelated waveforms. However, in practice, this property cannot be sustained because of multipath propagation or asynchronous transmission which makes the waveforms correlated. Two different channel models of the uplink are introduced. In the first part of this section, the synchronous channel is described, whereas in the second part the asynchronous multipath environment is considered.

However, in both cases, the baseband equivalent output signal of the  $k$ th user, denoted by  $q_k(t)$ , can be written in the same form. For the sake of simplicity, we apply BPSK modulation although the equations can also describe more sophisticated multivalued modulation schemes (e.g., QPSK, 16QAM). User  $k$  transmits  $b_k[i] \in \{-1, +1\}$  binary symbols where  $i$  refers to the time instant. The output signal is given as

$$q_k(t) = A_k \sum_{i=1}^{N_B} b_k[i] s_k(t - iT - \theta_k), \quad (1)$$

where  $A_k$  denotes the signal amplitude associated with the  $k$ th user,  $T$  refers to the time period of one symbol,  $\theta_k$  denotes the delay of user  $k$ , and  $N_B$  is the block size, respectively. The spreading waveform of user  $k$  is denoted by  $s_k(t)$ . In the case of direct sequence (DS) multiple access, it can be written as follows:

$$s_k(t) = \sum_{i=0}^{PG-1} S_k[i] e^{j(t - iT_c)}. \quad (2)$$

Here  $S_k[i]$  denotes the  $i$ th time chip of user  $k$ , and  $e(t)$  is the elementary waveform in the system. The chip duration is denoted by  $T_c$ . Each user is transmitting over a specific channel which can be characterized by its impulse response function

denoted by  $h_k(t)$ . The received signal is the sum of the arriving signals plus a Gaussian noise which can be written as

$$r(t) = \sum_{k=1}^K h_k(t) * q_k(t) + n(t), \quad (3)$$

where  $K$  refers to the number of users and  $n(t)$  is a white Gaussian noise with a constant one-sided  $N_0$  spectral density. Depending on the properties of  $h_k(t)$ , we get different received signals ( $r(t)$ ), but the resulting channel matched filter (CMF) output can be expressed in the same mathematical form, as will be pointed out in the following sections.

### 2.1. Synchronous model

In a symbol synchronous channel, all users are forced to transmit at the same time instant ( $\theta_k = 0$ ), their channels are characterized by single-path attenuation factors  $h_k^s(t) = \alpha_k \delta(t)$ , where  $\delta(t)$  is the Dirac delta function. The received signal in (3) is given as

$$r^s(t) = \sum_{k=1}^K \sum_{i=1}^{N_B} A_k \alpha_k b_k[i] s_k(t - iT) + n(t), \quad (4)$$

where the “ $s$ ” in the superscript refers to synchronous transmission. In the case of signatures limited to one symbol length ( $s_k(t) = 0$  if  $t \notin [0, T)$ ), there is no intersymbol interference, consequently, without loss of generality, index  $i$  can be omitted which yields

$$r^s(t) = \sum_{k=1}^K A_k \alpha_k b_k s_k(t) + n(t), \quad t \in [0, T). \quad (5)$$

The conventional detector consists of  $k = 1, 2, \dots, K$  filters matched with the signature waveforms and channels, generating the following output for the  $k$ th user:

$$\tilde{b}_k^s = \alpha_k^* \int_0^T r^s(t) s_k^*(t) dt. \quad (6)$$

In the case of BPSK modulation, the traditional single-user detector (SUD) simply calculates the signum of expression (6) yielding  $\hat{b}_k^{\text{SUD}} = \text{sign}\{\tilde{b}_k^s\}$ . This will, however, severely deteriorate the detector's performance, as can be seen from the expanded version of formula (6)

$$\tilde{b}_k^s = \overbrace{A_k R_{kk}^s b_k}^{\text{signal}} + \overbrace{\sum_{m=1, m \neq k}^K A_m R_{km}^s b_m}^{\text{multiple access interference}} + \overbrace{\tilde{n}_k^s}^{\text{noise}}, \quad (7)$$

where  $R_{km}^s$  is defined as follows:

$$R_{km}^s = \alpha_k^* \alpha_m \int_0^T s_k^*(t) s_m(t) dt, \quad (8)$$

and  $\tilde{n}_k^s = \alpha_k^* \int_0^T s_k^*(t) n(t) dt$  is a zero mean colored Gaussian noise due to linear transformation. The output of the

matched filter in vector form is

$$\tilde{\mathbf{b}}^s = \mathbf{R}^s \mathbf{A}^s \mathbf{b} + \tilde{\mathbf{n}}^s, \quad (9)$$

where  $\mathbf{A}^s = \text{diag}[A_1, A_2, \dots, A_K]$  is a diagonal matrix,  $\mathbf{R}^s = [R_{km}^s]$  is a  $K \times K$  Toeplitz, diagonal dominant matrix, and Hermitian ( $R_{km}^s = R_{mk}^{s*}$ ). The vectors are given as  $\tilde{\mathbf{b}}^s = [\tilde{b}_k^s]$ ,  $\mathbf{b} = [b_m]$ , and  $\tilde{\mathbf{n}}^s = [\tilde{n}_m^s]$ , respectively.

## 2.2. Asynchronous model

Synchronous high bit rate communication cannot be ensured in practice due to multipath propagation. Thus, it is required to develop more sophisticated channel models for either analytical investigations or computer simulations. For the sake of generality, we assume a general multipath propagation channel. In this case, the channel impulse response function for the  $k$ th user  $h_k^a(t)$  is a general continuous function as opposed to the previous case where  $h_k^s(t)$  was a simple attenuation factor. For instance

$$h_k^a(t) = \sum_{j=1}^5 \alpha_{kj} \delta(t - \tau_{kj}) \quad (10)$$

is a five-path propagation model where the  $j$ th path's attenuation and delay of user  $k$  are  $\alpha_{kj}$  and  $\tau_{kj}$ , respectively. Here the superscript "a" refers to the asynchronous transmission. To ensure simple notations, a new function is defined

$$v_k(t) = h_k^a(t) * s_k(t - \theta_k). \quad (11)$$

Substituting into (3), we obtain

$$r^a(t) = \sum_{k=1}^K \sum_{i=1}^{N_B} A_k v_k(t - iT) b_k[i] + n(t). \quad (12)$$

The  $k$ th matched filter performs a convolutional product on the incoming stream with  $v_k^*(-t)$ , which results in a continuous signal  $\tilde{r}_k^a(t)$ , which can be written as  $\tilde{r}_k^a(t) = v_k^*(-t) * r^a(t)$ . The output of the channel matched filters is given then in the form of

$$\tilde{\mathbf{r}}^a(t) = \sum_{i=1}^{N_B} \Phi_{vv}^a(t - iT) \mathbf{A}^s \mathbf{b}[i] + \tilde{\mathbf{n}}^a(t), \quad (13)$$

where  $\mathbf{A}^s$  is defined previously, and

$$\begin{aligned} \tilde{\mathbf{r}}^a(t) &= [\tilde{r}_1^a(t), \tilde{r}_2^a(t), \dots, \tilde{r}_K^a(t)]^T, \\ \mathbf{b}[i] &= [b_1[i], b_2[i], \dots, b_K[i]]^T. \end{aligned} \quad (14)$$

Furthermore,  $\tilde{\mathbf{n}}^a(t) = \mathbf{v}^*(-t) * n(t)$  and  $\Phi_{vv}^a(t)$  is a  $K \times K$  matrix with all correlation functions between  $v_i(t)$  and  $v_j(t)$ . It can be written as a dyadic convolution product

$$\Phi_{vv}^a(t) = \mathbf{v}^*(-t) * \mathbf{v}^T(t), \quad (15)$$

where  $\mathbf{v}(t) = [v_i(t)]$ . Sampling with  $(iT)$ , expression (13) results in a discrete-time model for mapping  $\mathbf{b}[i]$  into  $\tilde{\mathbf{b}}[i]$ ,

namely

$$\tilde{\mathbf{b}}^a[i] = \mathbf{R}^a[i] * \mathbf{A}^s \mathbf{b}[i] + \tilde{\mathbf{n}}^a[i], \quad (16)$$

where  $\tilde{\mathbf{b}}[i] = \tilde{\mathbf{r}}(iT)$ ,  $\tilde{\mathbf{n}}^a[i] = \tilde{\mathbf{n}}^a(iT)$ , and  $\mathbf{R}^a[i] = \Phi_{vv}^a(iT)$ , which is the discrete-time channel matrix. To describe the different detection schemes in Section 3, it is helpful to use the following block notation where the components of all vectors  $\tilde{\mathbf{b}}^a[i]$ ,  $\mathbf{b}[i]$ , and  $\tilde{\mathbf{n}}^a[i]$  are written into column vectors with  $KN_B$  elements

$$\begin{aligned} \tilde{\mathbf{b}}^a &= [\tilde{b}_1^a[1], \tilde{b}_2^a[1], \dots, \tilde{b}_K^a[1], \tilde{b}_1^a[2], \dots, \tilde{b}_K^a[N_B]]^T, \\ \mathbf{b} &= [b_1[1], b_2[1], \dots, b_K[1], b_1[2], \dots, b_K[N_B]]^T, \\ \tilde{\mathbf{n}}^a &= [\tilde{n}_1^a[1], \tilde{n}_2^a[1], \dots, \tilde{n}_K^a[1], \tilde{n}_1^a[2], \dots, \tilde{n}_K^a[N_B]]^T. \end{aligned} \quad (17)$$

The convolutional form in (16) can be rewritten as a multiplication

$$\tilde{\mathbf{b}}^a = \mathbf{R}^a \mathbf{A}^a \mathbf{b} + \tilde{\mathbf{n}}^a, \quad (18)$$

resulting in the same form as in (9) but with different parameters. Here  $\mathbf{A}^a = \text{diag}[\mathbf{A}^s, \mathbf{A}^s, \dots, \mathbf{A}^s]$  is a  $KN_B \times KN_B$  dimensional diagonal matrix, containing  $N_B$  sub-matrices of  $\mathbf{A}^s$ . Matrix

$$\mathbf{R}^a = \begin{bmatrix} \mathbf{R}^a[0] & \mathbf{R}^a[-1] & \dots & \mathbf{R}^a[-N_B+1] \\ \mathbf{R}^a[1] & \mathbf{R}^a[0] & \dots & \mathbf{R}^a[-N_B+2] \\ \vdots & & \ddots & \\ \mathbf{R}^a[N_B-1] & \dots & \mathbf{R}^a[1] & \mathbf{R}^a[0] \end{bmatrix} \quad (19)$$

is a  $KN_B \times KN_B$  dimensional block Toeplitz matrix and Hermitian ( $\mathbf{R}^a[i] = (\mathbf{R}^a[-i])^H$  for all  $i$ ) with diagonal dominance ( $|R_{ii}^a| > |R_{ij}^a|$  for all  $(i, j)$ ;  $i \neq j$ ). Moreover, it contains many zero elements, in most practical situations only  $\mathbf{R}^a[\ell]$ , for all  $|\ell| \leq 6$  differ from zero.

Since both the synchronous and asynchronous case can be treated by the same mathematical formula (see (9) and (18)) in the forthcoming discussion, we use  $\tilde{\mathbf{b}}$ ,  $\tilde{\mathbf{n}}$ ,  $\mathbf{A}$ , and  $\mathbf{R}$  without indices which can either refer to the synchronous or the asynchronous case.

## 3. MULTIUSER DETECTION

Based on the linear model introduced in (9) and in (18) applying BPSK modulation (only the real part of all matrices and vectors are taken into account), we can derive the optimal MUD as a maximum likelihood sequence estimation (MLSE) problem which can then be obtained in the following fashion:

$$\hat{\mathbf{b}}^{\text{opt}} = \arg \max_{\mathbf{x} \in \{-1, +1\}^K} [\Pr\{\mathbf{x} | \tilde{\mathbf{b}}\}], \quad (20)$$

which yields [1, 2]

$$\hat{\mathbf{b}}^{\text{opt}} = \arg \min_{\mathbf{x} \in \{-1, +1\}^K} [\mathbf{x}^T \mathbf{A} \mathbf{R} \mathbf{A} \mathbf{x} - 2 \mathbf{x}^T \mathbf{A} \tilde{\mathbf{b}}]. \quad (21)$$

This equation involves the global minimization of a quadratic form where the perfect knowledge of system parameters  $\mathbf{A}$  and  $\mathbf{R}$  is required. In practice, the RAKE receiver is used to keep track with the incoming signal and to estimate the components of matrix  $\mathbf{R}$ . A power control mechanism is applied to ensure that  $\text{diag}[\mathbf{RA}]$  tends to the unit matrix.

The optimal solution can be found, for example, by exhaustive search using (21), however, it implies exponentially increasing computational complexity as the number of users grows. In real life implementation, we cannot afford to use such a time wasting mechanism, thus many sub-optimal MUD schemes have been studied in the recent past [1]. Among the investigated ones, neural networks, which are widely accepted and used for classification and optimization tasks, have received considerable interest in sub-optimal MUD algorithms too. Analyzing (20), we can apply a feed-forward network for MUD. The first feed-forward neural network-based MUD was proposed by Aazhang et al. in [5]. Since that time, dozens of articles have appeared, for example, [6, 7, 8, 9, 10, 11], including further studies on the subject. The papers treating feed-forward neural network-based MUD can be distinguished mainly by the applied training method.

In this paper, we focus on recurrent neural network detection which can be characterized by strong feedback mechanisms. Recurrent neural networks are built up by computing elements termed as neurons. In each neuron a simple updating process is realized

$$Y_l[\ell + 1] = \text{sign} \left\{ \frac{1}{W_{ll}} \left( \sum_{j=1}^{\ell-1} W_{lj} Y_j[\ell + 1] + \sum_{j=\ell+1}^M W_{lj} Y_j[\ell] + V_l \right) \right\}, \quad (22)$$

where  $Y_j[\ell]$  is the output of the  $j$ th neuron at the  $\ell$ th iteration. Parameter  $V_l$  denotes the threshold of the  $l$ th neuron and  $W_{lj}$  is the connection strength between the output of the  $j$ th neuron and the input of the  $l$ th neuron. The size of the network is denoted by  $M$ . It has been proven [12, 13] that the Hopfield net will drive the following quadratic form (often referred to as Lyapunov function) into local minimum in the  $\{-1, +1\}^M$  state space

$$J(\mathbf{y}[\ell]) = -(\mathbf{y}[\ell])^T \mathbf{W} \mathbf{y}[\ell] - 2(\mathbf{y}[\ell])^T \mathbf{v}, \quad (23)$$

where  $\mathbf{y}[\ell] = [Y_i[\ell]]$ ,  $\mathbf{v} = [V_j]$ , and  $\mathbf{W} = [W_{ij}]$ , respectively, [14].

For a two-dimensional example see Figure 1. In the figure, the Lyapunov function is depicted as a discrete function of the neuron outputs. We assume that the two dimensional neural network runs from the initial state  $\mathbf{y}[0] = [-1, -1]^T$ . In each step, the net decreases the Lyapunov function, thus the first iteration produces  $\mathbf{y}[1] = [-1, +1]^T$  and in the second iteration, the network arrives at the steady state  $\mathbf{y}[2] = [+1, +1]^T$ .

Due to the similarity between (21) and (23), we can use the Hopfield net for MUD (as was done in [15, 16, 17]). In

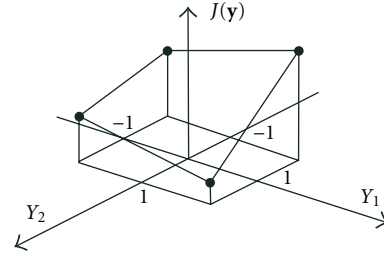


FIGURE 1: Two-dimensional Lyapunov function.

order to do this, we have to choose the parameters in (22) in the following manner:

$$\mathbf{W} = -\mathbf{A}\mathbf{R}\mathbf{A}, \quad (24)$$

$$\mathbf{v} = \mathbf{A}\tilde{\mathbf{b}}, \quad (25)$$

where matrix  $\mathbf{A}$  and matrix  $\mathbf{R}$  are the previously defined amplitude and discrete-time channel matrices, respectively. The number of performed iterations is denoted by  $T_r$ , which is limited by the speed of hardware implementation and the symbol period  $T$ . Thus, the estimated symbols are given by the outputs of the neural network in the last iteration

$$\hat{\mathbf{b}}^{\text{RNN}} = \mathbf{y}[T_r]. \quad (26)$$

The first *implicit* Hopfield neural network-based realization of MUD was introduced by Varanasi and Aazhang [18]. They defined a multistage detection scheme without mentioning the term *neural*. Later many authors, for example, Miyajima et al. [15], Kechriotis and Manolakos [16], and Teich and Seidl [17] have shown that this structure is identical to—and can be replaced with—the recurrent neural network detector, often referred to as Hopfield net detector.

Since the Lyapunov function in (23) can only be locally maximized by traditional Hopfield neural networks, the optimal solution is not reached in most cases. Avoiding the local maxima of the Lyapunov function results in a better performance. Many articles introduced modified recurrent neural network structures to achieve improved performance in MUD by avoiding local minima. For instance, Yoon and Rao [19] and Chen et al. [20] have proposed an annealed neural network multiuser receiver, which is based on normal Hopfield network, but they applied optimized sigmoid function as nonlinearity in (22). This modification resulted in remarkable performance enhancement, although the optimality of this strategy has not yet been verified analytically. Kechriotis and Manolakos proposed another modified structure which was named as hybrid detector. It is built up by a reduced detector followed by a normal Hopfield neural network [21]. Although there exists some analytical description on the behavior of this hybrid receiver structure, its optimality has not yet been proven. Wang et al. [22] proposed a transiently chaotic neural network-based multiuser receiver scheme which originates from chaos theory. This deterministic network can reach the global optimum, however

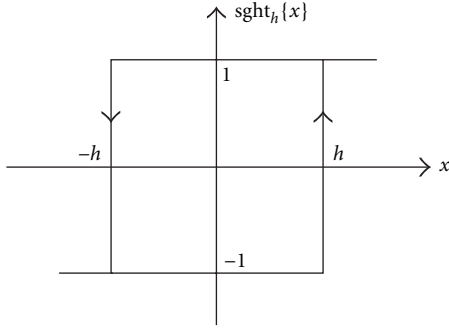


FIGURE 2: The hysteretic decision function.

its performance is deeply influenced by the choice of initial parameters which are set on the basis of experimental results.

To achieve performance enhancement, we use stochastic sources inside the neurons and to speed up the network, we apply hysteretic type nonlinearity. These two methods are introduced in the next section.

#### 4. THE MODIFIED HOPFIELD NETWORK

For the sake of generality, we use the following updating function instead of (22):

$$Y_l[\ell + 1] = \text{sgth}_{h_l} \left\{ \frac{1}{W_{ll}} \left( \sum_{j=1}^{l-1} W_{lj} Y_j[\ell + 1] + \sum_{j=l+1}^M W_{lj} Y_j[\ell] + V_l + v_l[\ell] \right) \right\}, \quad (27)$$

where  $v_l[\ell]$  is a stochastic term with  $F(x, \ell)$  distribution at time instant  $\ell$ , that is,  $\Pr\{v_l[\ell] \leq x\} = F(x, \ell)$ . Parameter  $h_l$  denotes the bound of the hysteresis function related to the  $l$ th neuron,  $\text{sgth}_{h_l}\{\cdot\}$  is a hysteretic decision function, with decision boundary  $h$ , see Figure 2 for better comprehension. If one set  $h_l = 0$  and  $v_l[\ell] = 0$ , then the operation of the original Hopfield net is obtained.

In Figure 3, the above defined modified recurrent neural network is depicted. This structure combines the advantages of the two previously investigated methods, namely the stochastic and the hysteretic type recurrent neural networks. The former is able to escape from local maxima due to the stochastic nature and to find the optimal solution of the corresponding quadratic form. In exchange, additional iterations are required [4]. Whereas the latter is able to reduce the number of iterations to provide faster detection [3]. Both are detailed in the forthcoming two subsections.

##### 4.1. Stochastic recurrent neural net

The additional  $v[\ell]$  term is expected to be symmetric, to provide equiprobability of movements in both directions

$$\Pr\{v[\ell] < x\} = F(x, \ell) = 1 - F(-x, \ell). \quad (28)$$

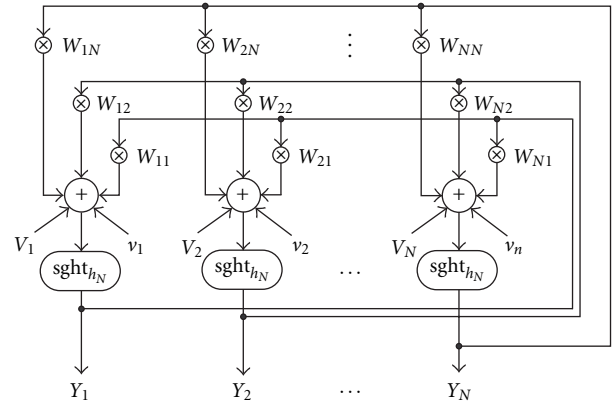


FIGURE 3: The structure of the modified recurrent neural network.

Furthermore, in order to obtain a stable solution, the variance of  $v[\ell]$  should decrease with  $\ell$ .

The authors claim to prove that if we use

$$F(x, \ell) = \frac{1}{1 + e^{\gamma[\ell]x}} \quad (29)$$

distribution—termed as logistic distribution—to generate the values of  $v[\ell]$ , where  $\gamma[\ell]$  is a negative parameter, decreased with  $\ell$ , then the recurrent neural network defined in (27) finds the optimum of the corresponding quadratic form defined in (21) with the highest probability. This statement is a subject of a separate paper and is not proven here.

However, the cooling schedule defining the function  $\gamma[\ell]$  remains open for further investigation. Previously, we used  $\gamma[\ell] = -1.5 \cdot \ell$ , which is applicable in the case of randomly generated channel matrices [23]. In other cases, where channel matrices are constant, the resulting performance is slow [24]. In general, parameter  $\gamma[0]$  and the function of  $\gamma[\ell]$  is a trade-off between performance and speed; quickly decreased  $\gamma[\ell]$  values results in quick detection, but worse performance. On the contrary, slowly decreased  $\gamma[\ell]$  values can yield almost optimal performance but the network may require more iterations to find the steady state.

##### 4.2. Neural net with hysteretic type nonlinearity

Recurrent neural networks with hysteretic type nonlinearities can provide fast convergence since hysteresis prevents output changes in the case of small input values. Hysteretic type recurrent neural networks was introduced by Levendovszky et al. [3]. Here it is taken into consideration that the parameters are still defined based on rule (24) and (25) but the power control is perfect, that is,  $\mathbf{A} = \mathbf{I}$ . Later,  $\mathbf{R}$  is used instead of  $\mathbf{W}$  and  $\mathbf{b}$  is considered instead of  $\mathbf{v}$ . For special  $\mathbf{R}$  matrices, the uniqueness of the steady state, which is at the global optimum of the corresponding Lyapunov function, can be proven [3]. First, we define the so-called openness parameter ( $D$ ) in the following form:

$$|R_{ii}| \geq D \cdot \sum_{j; j \neq i} |R_{ij}| \quad \forall i, \quad (30)$$

TABLE 1: Probability of being in the DC.

$E_b/N_0$	$\Pr\{\mathbf{m} \in \text{DC}_\varepsilon\}$	$\Pr\{\mathbf{m} \notin \text{DC}_\varepsilon\}$
10 dB	0.631878	$3.681 \cdot 10^{-1}$
15 dB	0.910229	$8.977 \cdot 10^{-2}$
20 dB	0.998473	$1.527 \cdot 10^{-3}$

where  $D$  must be positive obviously. Secondly, we define the internal double cube (DC) with parameter  $\varepsilon$  ( $0 < \varepsilon < 1$ ), which is denoted by  $\text{DC}_\varepsilon$ ,

$$\text{DC}_\varepsilon \equiv \{\mathbf{x} = [x_i] : \forall i, \varepsilon \leq |x_i| \leq 2 - \varepsilon\}. \quad (31)$$

Finally, we introduce vector  $\mathbf{m}$  as the continuous solution of (21), that is,  $\mathbf{m}$  satisfies  $\dot{\mathbf{b}} = \mathbf{R}\mathbf{m}$ .

**Theorem 1.** *If there exists  $0 < \varepsilon < 1$  parameter such that  $\mathbf{m} \in \text{DC}_\varepsilon$ , and if*

$$\kappa = (\varepsilon(1 + D) - 3) \min_i \sum_{j:j \neq i} |R_{ij}| \quad (32)$$

*is positive, then the corresponding Hopfield recursion converges to the global optimum. Moreover, reaching the optimal state can be accelerated by using hysteresis decision function with parameter*

$$h_i = |R_{ii}| - R_{ii} + \kappa. \quad (33)$$

*Proof.* For detailed proof see [3]. The outline of the proof can be summarized as follows: firstly, it can be seen that if  $\mathbf{m}$  is in the double cube  $\text{DC}_\varepsilon$ , then  $\mathbf{y} = \text{sign}\{\mathbf{m}\}$  is one steady state solution of the corresponding quadratic form. Secondly, there is no other  $\mathbf{y} \neq \text{sign}\{\mathbf{m}\}$  which could be a steady state. The third proof shows that  $\mathbf{y} = \text{sign}\{\mathbf{m}\}$  gives a stable solution.  $\square$

Note that introducing hysteretic nonlinearity results in less state changes, thus the hysteretic neural network is more stable than the original Hopfield neural network. As a multiuser detector, the Hopfield network reaches the steady state in less than ten iterations (even if the number of users and package sizes are large). Thus, the application of hysteretic Hopfield network alone, as a multiuser detector, is not advisable. The need for application of hysteretic Hopfield network alone, as a multiuser detector primarily, arises in overloaded systems where computation time must be minimized to handle all connections.

Sometimes, however,  $\kappa > 0$  could not be satisfied. Since in (32) only the term  $(\varepsilon(1 + D) - 3)$  can be negative, we investigate its behavior. Two variables are considered:  $D$  and  $\varepsilon$ . Parameter  $D$  is deterministic, it depends on the structure of the discrete-time channel matrix  $\mathbf{R}$ , and can be computed easily. On the other hand, due to the existence of unbounded thermal noise,  $\dot{\mathbf{b}}$  can take an arbitrarily large value (see (9) and (18)). Thus, the vector  $\mathbf{m}$  can also be infinitely large, the boundary of the DC tends to infinity, hence  $\varepsilon$  cannot be kept in the interval  $(0, 1)$ . Nevertheless, if some error is allowed

in the system, we can choose parameter  $\varepsilon$  to provide sufficiently small fault probability. In the next section, an example is shown on how to compute the fault probability, that is, the probability of  $\mathbf{m}$  escapes from the double cube  $\text{DC}_\varepsilon$ .

### 4.3. A simple example

Let the following system be investigated, which is defined by its discrete-time channel matrix

$$\mathbf{R} = \begin{bmatrix} 1.2 & 0.2 & -0.15 \\ 0.2 & 1.1 & 0.1 \\ -0.15 & 0.1 & 1 \end{bmatrix}, \quad (34)$$

and  $\mathbf{b} = [-1, +1, -1]^T$ . It is easy to see that  $D = 24/7$  is the eye-openness parameter. To sustain positive  $\kappa$ , we must choose at least  $\varepsilon = 21/31$  (see (32)). The output of the RAKE receiver (9) is given as  $\dot{\mathbf{b}} = [-0.85, 0.8, -0.75]^T + \dot{\mathbf{n}}$ , where  $\dot{\mathbf{n}}$  is a zero mean Gaussian noise with covariance matrix  $\mathbf{R}N_0$ . Based on the above statements  $\mathbf{m} = \mathbf{R}^{-1}\dot{\mathbf{b}}$ , where  $\mathbf{m}$  is still a noise term, with  $\mathbf{b}$  mean and  $\mathbf{R}^{-1}N_0$  covariance. Taking into account the expression of three-dimensional normal distribution, the probability of satisfying the condition defined in Theorem 1 can be expressed as

$$\Pr\{\mathbf{m} \in \text{DC}_\varepsilon\} = \int_{\mathbf{x} \in \text{DC}_\varepsilon} \frac{\exp\{-\frac{(\mathbf{x} - \mathbf{b})^T \mathbf{R}(\mathbf{x} - \mathbf{b})}{2N_0}\}}{\sqrt{2\pi N_0}^3 \sqrt{\det \mathbf{R}^{-1}}} d\mathbf{x}. \quad (35)$$

The results for different  $E_b/N_0$  scenarios is shown in Table 1. The event  $\mathbf{m} \in \text{DC}_\varepsilon$  gives a sufficient—but not necessary—condition that the optimal solution is found. Thus, the value of  $\Pr\{\mathbf{m} \in \text{DC}_\varepsilon\}$  gives a lower bound to the probability of finding the global optimum. For instance, if  $E_b/N_0 = 15$  dB, then the probability of not being in the double cube with  $\mathbf{m}$  is equal to 0.08977. In other words, for the discrete time channel matrix  $\mathbf{R}$  defined in (34), the optimal solution is not found at a maximum probability of 9%. We must note that the probability of falling into the DC is not equal to the probability of finding the global optimum. In most cases, under faulty circumstances, it is still possible to start from a neighboring state and reach the global optimum.

## 5. ADAPTATION METHOD

In this paper, the adaptation is regarded as the approximation of discrete-time channel matrix  $\mathbf{R}$ , whereas the amplitude matrix  $\mathbf{A}$  is assumed to be known in the receiver. Furthermore, for the sake of simplicity, the amplitude matrix is assumed to be a unit matrix ( $\mathbf{A} = \mathbf{I}$ ) in the sequel. However, the equations can be easily expanded to the more general case.

RAKE receiver is designed to synchronize and to keep track of the incoming signal. The fingers of the RAKE receiver are assigned to different multipath components. These fingers are locked to the signal when sensing synchronization takes place. Assuming perfect knowledge of system parameters ( $\mathbf{R}$ ), the optimal decision rule for single-user scenarios is

to apply a signum function on (6) and (9). However, perfect estimation of the channel parameters may not be ensured due to thermal noise and multiuser interference. In our case, the RAKE receiver is modeled as an absolutely erroneous estimator—this situation is termed as noncoherent scenario—yielding  $\hat{\mathbf{R}} = \mathbf{I}$  (i.e., the unit matrix instead of  $\mathbf{R}$ ) as the input of the adaptation method. In spite of the erroneous estimator, we still consider the output of the channel matched filter to be  $\hat{\mathbf{b}}$  which is the perfect case. This may seem to be in contradiction with the previous statement and could invite criticism. Nevertheless, the explanation is simple: in this paper, the proposed method is analyzed in its simplest way (taking into account that a faulty channel matched filter remains as a subject of further investigation).

To carry out the detection, the detector algorithm needs the exact form of matrix  $\mathbf{R}$  (see, e.g., (24)). Due to the unknown channel and noise, the elements of this matrix must be estimated recursively from the incoming signal by the means of adaptation. The adaptation method applied in the paper is based on Kohonen's self-organizing map (SOM) [25, 26]. This method was first implemented for the purpose of MUD by Hottinen [27] who used SOM followed by a two-stage detector. We investigate two types of adaptation which are commonly used in mobile communication (see [25, 26]) supervised and unsupervised algorithms. When the decisions are assumed to be correct at the detector (near optimal performance), a training set can be obtained by using the detected symbols matched against the received sequence at the input of the receiver. In this case, supervised methods such as learning vector quantization (LVQ) can be of use. However, when the detected sequence contains errors with high probability (the detector is far from optimal performance), blind equalization techniques, such as SOM, are needed.

### 5.1. Supervised learning based on LVQ

To apply supervised equalization, we assume that a training set is given

$$\tau^{(N)} = \left\{ (\mathbf{b}^{[z]}, \tilde{\mathbf{b}}^{[z]}), z = 1, \dots, N \right\}, \quad (36)$$

where  $\tilde{\mathbf{b}}^{[z]}$  represents the received block number  $z$  (see (9) and (18)). In the case of communication session (following the training period), we can replace  $\mathbf{b}^{[z]}$  with the detected sequence  $\hat{\mathbf{b}}^{[z]}$ . If the detector works in a near optimal fashion, we can track further changes in the channel or noise characteristics based on this set. LVQ equalizer operates in the following way (see [27]):

$$\begin{aligned} \hat{\mathbf{R}}^{[z+1]} &= \hat{\mathbf{R}}^{[z]} + \delta^{[z]} (\tilde{\mathbf{b}}^{[z]} - \hat{\mathbf{R}}^{[z]} \hat{\mathbf{b}}^{[z]}) (\hat{\mathbf{b}}^{[z]})^T, & \text{if } \tilde{\mathbf{b}}^{[z]} \in \omega_{\hat{\mathbf{b}}^{[z]}}, \\ \hat{\mathbf{R}}^{[z+1]} &= \hat{\mathbf{R}}^{[z]} - \delta^{[z]} (\tilde{\mathbf{b}}^{[z]} - \hat{\mathbf{R}}^{[z]} \hat{\mathbf{b}}^{[z]}) (\hat{\mathbf{b}}^{[z]})^T, & \text{if } \tilde{\mathbf{b}}^{[z]} \notin \omega_{\hat{\mathbf{b}}^{[z]}}, \end{aligned} \quad (37)$$

where  $\omega_{\hat{\mathbf{b}}^{[z]}}$  is the Voronoi tessellation of vector  $\hat{\mathbf{b}}^{[z]}$ . The

learning rate is defined as

$$\delta^{[z+1]} = \frac{\delta^{[z]}}{1 + c^{[z]} \delta^{[z]}}, \quad (38)$$

where  $c^{[z]} = +1$  in the case of correct classification, that is,  $\hat{\mathbf{b}}^{[z]} = \mathbf{b}^{[z]}$ , and  $c^{[z]} = -1$  otherwise. In the expression above,  $\tilde{\mathbf{b}}^{[z]} \notin \omega_{\hat{\mathbf{b}}^{[z]}}$  denotes misclassification when  $\tilde{\mathbf{b}}^{[z]}$  does not fall into the corresponding Voronoi cell denoted by  $\omega_{\hat{\mathbf{b}}^{[z]}}$ .

### 5.2. Blind learning based on SOM

In the case of blind learning, we can apply the SOM algorithm described as follows:

$$\begin{aligned} \hat{\mathbf{R}}^{[z+1]} &= \hat{\mathbf{R}}^{[z]} + \eta^{[z]} (\tilde{\mathbf{b}}^{[z]} - \hat{\mathbf{R}}^{[z]} \hat{\mathbf{b}}^{[z]}) (\hat{\mathbf{b}}^{[z]})^T \\ &+ \sum_{\mathbf{x} \in A_{\hat{\mathbf{b}}^{[z]}}} \beta^{[z]} (\tilde{\mathbf{b}}^{[z]} - \hat{\mathbf{R}}^{[z]} \mathbf{x}) \mathbf{x}^T, \end{aligned} \quad (39)$$

where  $\hat{\mathbf{b}}^{[z]}$  represents the detected binary vector at the output of MUD,  $\mathbf{x}$  denotes a generic binary vector and  $A_{\hat{\mathbf{b}}^{[z]}}$  denotes a set of rival decisions. Rival decisions refer to a set of possible classifications excluding the chosen one (which is  $\hat{\mathbf{b}}^{[z]}$ ). Choosing the weight sequence  $\eta^{[z]}$  should satisfy the conditions

$$\sum_i (\eta^{[i]})^2 < \infty, \quad \sum_i \eta^{[i]} = \infty \quad (40)$$

to adapt to a stationary distribution. Similar conditions are imposed on  $\beta^{[z]}$  [27].

One further structure should be considered, which is a modification of the SOM based blind learning. Here (39) is used with  $\beta^{[z]} = 0$ . This method is often termed adaptive vector quantization (AVQ) or the on-line  $k$ -means method. It is worth emphasizing that AVQ is still a blind method while LVQ was a supervised one. The reader should not be confused by the phrase. For both the synchronous and asynchronous case, to avoid the increment in the computational complexity, we focus only on the AVQ technique, that is,  $\beta^{[z]} = 0$ , but we term it SOM to avoid misunderstandings.

## 6. THE STRUCTURE OF THE PROPOSED DETECTOR

In Figure 4, the structure of the proposed detector is depicted. The RAKE receiver block is assumed to be incorporated, it receives the incoming signal and as an output gives the values  $\hat{\mathbf{b}}$  and  $\hat{\mathbf{R}}$ . At the top of the figure, a random number generator is shown, it generates noise values to the neurons of the stochastic recurrent neural network-based on the distribution function (29). This block has two inputs,  $\ell$  indicates the iteration instant, and  $A$  is the initial variance parameter of noise:  $\gamma[\ell] = -A \cdot \ell$ . On the right, the modified neural network is shown which has already been depicted by Figure 3. The input matrix  $\hat{\mathbf{R}}$  and vector  $\hat{\mathbf{b}}$  are used to set the network parameters based on the rule (24) and (25), vector  $\mathbf{h} = [h_1, h_2, \dots, h_N]^T$  determines the hysteresis parameters of the network. The output  $\hat{\mathbf{b}}^{\text{MNN}}$  is the estimation of the

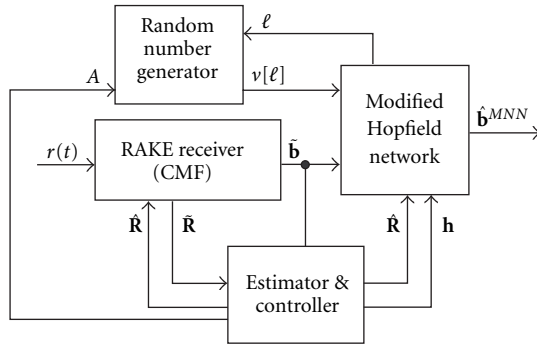


FIGURE 4: The structure of the proposed detector.

sequence  $\mathbf{b}$ . The role of the block at the bottom of the figure is twofold; on one hand, it is an adaptive equalizer to fix the estimation error of the RAKE receiver in the sense of  $\hat{\mathbf{R}}$ . On the other hand, it controls the entire detector. Depending on the structure of matrix  $\mathbf{R}$ , this unit decides what operation should the detector follow: normal, stochastic, hysteretic type or the latter two. In the following subsections, these four different operations are detailed.

### 6.1. Normal operation

If the eye-openness parameter is close to—or less than—one, and there is a limited amount of time to iterate, then normal operation is performed, thus both  $A$  and vector  $\mathbf{h}$  are set to zero. Neither stochastic nor hysteretic properties are exhibited in this case. The controller unit is ordered to focus on adaptive estimation of matrix  $\hat{\mathbf{R}}$ .

### 6.2. Stochastic operation

If there is enough computational capacity (i.e., the mobile network is rarely loaded) or the discrete-time channel matrix  $\mathbf{R}$  shows poor eye-openness ( $D$  is small), then the controller can switch to stochastic operation. The limit vector  $\mathbf{h}$  is set to contain zero elements, thus hysteresis in the decision function of the neurons does not appear. Parameter  $A$  is set to be the proper value which can be pre-programmed constant (e.g., 2.5), or can be adaptively updated in the course of operation based on bit error ratio (BER) measurements.

### 6.3. Hysteretic operation

To perform hysteretic operation, matrix  $\mathbf{R}$  must show *wide* eye-openness. If the receiver serves many users, and there is lack of computational time, the controller drives the detector to perform hysteretic operation. Vector  $\mathbf{h}$  depends on the time the receiver has to demodulate, larger  $h$  value produces higher speed detection but worse bit error ratio. Note that the minimal value of  $h$  is zero, a negative value drives the detector to malfunction. Parameter  $A$  is set to zero, thus the stochastic nature disappears.

### 6.4. Stochastic and hysteretic operation

The best performance can be achieved by applying both stochastic and hysteretic properties in the state transition rule

(27). The advantage of hysteresis truly lies in streamlining the convergence time of the detector. The controller unit determines what  $\mathbf{h}$  and  $A$  are to be used. This operation is the most effective although there is no general recipe on how to set the corresponding parameters. A separate paper could be dedicated to this problem, thus it is not considered here. When simulating the detector, we focus on the stochastic operation.

## 7. SIMULATION RESULTS

To show the applicability of the proposed structure, simulations of both synchronous and asynchronous uplink are considered in the following two subsections. In the synchronous case, we show the results for all adaptive scenarios. For the SOM structure, we have omitted the terms which take into consideration the possible rival decisions ( $\beta^{[z]} = 0$  in (39)). In this way, a considerable computational simplification became possible. Despite the fact that this simplification results in performance degradation, the simplified structure is still able to yield a performance close to the coherent case, as will be demonstrated later.

### 7.1. Synchronous case

Uplink with synchronous transmission can be regarded as a system where intersymbol interference (ISI) is somehow avoided. If we assign limited bandwidth for the communication link, it will result in infinite waveform in the time domain which gives rise to the effect of ISI. Thus, synchronous transmission models are not realistic, but they provide an easy way for simulation and get insight into the applicability of different methods. Section 7.2 deals with more sophisticated channel models.

Here the signal is modeled as a pulse amplitude modulated (PAM) stream, so  $e(t) = \Delta_{T_c}(t)$ , where  $\Delta_{T_c}(t)$  is equal to one, if  $0 \leq t < T_c$  and zero otherwise. AWGN scenario is investigated ( $\alpha^{[z]} = 1$ , for all  $k$ ) and the amplitude of the users are equal ( $\mathbf{A} = \mathbf{I}$ ). Seventeen users are considered ( $K = 17$ ) only with Gold codes of length  $PG = 15$ . Due to synchronous transmission, all  $\theta_k$  are equal to zero. The resulting discrete-time channel matrix  $\mathbf{R}$  from (8) is visualized in Figure 5. Each row and each column is assigned for one user, and the boxes represent the absolute value of cross correlation between the corresponding users. Dark tones refer to stronger correlations while light tones show weaker correlations between the corresponding users. The greatest elements of the matrix are located in the diagonal which are the auto correlation values and are equal to one.

For the synchronous scenario, the SHN was ordered to perform 1000 iterations with initial noise value  $A = 0.02$  with linearly decreased  $\gamma$ . For the noncoherent methods, 300 iterations were performed and the initial learning weight was set to  $\delta^{[0]} = 0.01$ ,  $\eta^{[0]} = 0.02$ , and  $\eta^{[z]} = \eta^{[0]}(1 - \tau/300)$ , where  $\tau$  refers to the iteration instance. The initial estimate of channel matrix  $\mathbf{R}$  was set to  $\hat{\mathbf{R}} = \mathbf{I}$  which corresponds to the worst case. In Figure 6, BER is exhibited as a function of  $E_B/N_0$ , the case of different detector structures and adaptation methods. The optimal performance can be lower



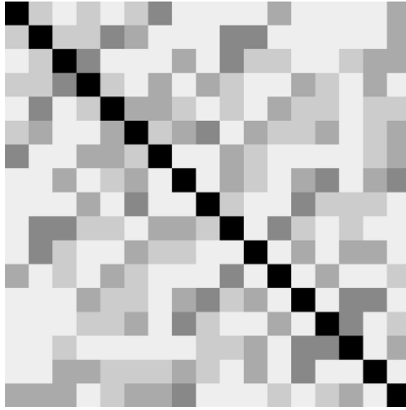


FIGURE 5: The structure of the channel matrix in the synchronous case.

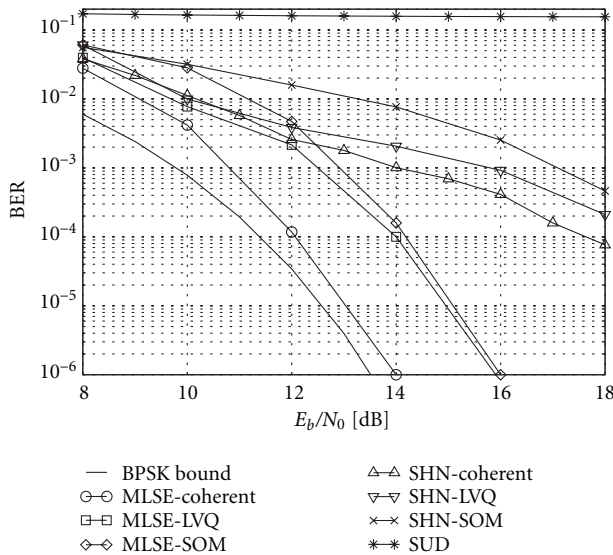


FIGURE 6: The synchronous case: BER versus  $E_b/N_0$ .

bounded by the theoretical BPSK limit

$$\text{BER}^{\text{optimal}} \geq \text{BER}^{\text{BPSK}} = \frac{1}{2} \text{erfc} \left( \sqrt{\frac{E_b}{2N_0}} \right), \quad (41)$$

where

$$\text{erfc}(x) = \int_x^\infty \frac{2}{\sqrt{\pi}} e^{-y^2} dy \quad (42)$$

denotes the complementary error function [28]. This BPSK bound is the lowest curve on the figure. The optimal detector is also simulated (assuming the full knowledge of  $\mathbf{R}$ ) denoted by MLSE-coherent on the figure. The corresponding curve is just beyond the BPSK bound. Next, the performance of the stochastic recurrent neural network was simulated assuming known channel characteristics. It is denoted by SHN-coherent on Figure 6. The corresponding curve is located above the MLSE-coherent as expected. Hereby, we must note

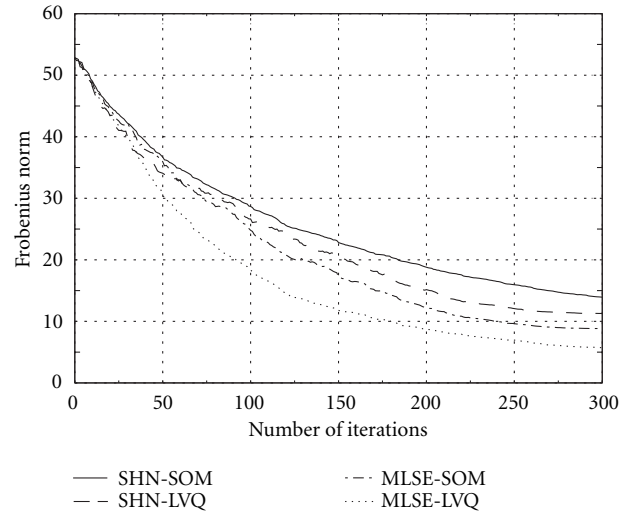


FIGURE 7: The synchronous case: Frobenius norm versus number of iterations.

that the SHN-coherent detection algorithm provides the best performance among all suboptimal detection methods [24]. For example, if we want to achieve  $\text{BER} = 10^{-3}$ , then only 3 dB additional signal-to-noise ratio (SNR) is needed compared with the MLSE-coherent, although the system is overloaded (serving 17 users with processing gain of length 15). It may be noteworthy that the SUD method cannot reach  $\text{BER} = 10^{-1}$  which is grossly unacceptable.

Some simulation results related to the noncoherent case are also depicted in Figure 6. Namely, the MLSE method was implemented with the LVQ algorithm to adaptively identify matrix  $\mathbf{R}$ . We see that the corresponding curve termed as MLSE-LVQ goes above the one of MLSE-coherent scenario due to the information loss on  $\mathbf{R}$ . Next, the MLSE detector with the self-organizing map is denoted by MLSE-SOM. Using an unsupervised adaptation method results in slight loss of performance (which can only be significant in the case of powerful noise). The performance of SHN was also evaluated by using both LVQ and SOM algorithms. Note that both curves are very close to the corresponding MLSE curves. An important conclusion can be drawn from the figure: for applying SHN with SOM, we pay only approximately 4 dB additional SNR related to the coherent SHN case.

In Figure 7, the Frobenius norm between the discrete-time channel matrix  $\mathbf{R}$  and its estimated counterpart  $\hat{\mathbf{R}}$  is given as a function of iterations (performed by the corresponding learning algorithms). The Frobenius norm is defined as follows:

$$\|\mathbf{R} - \hat{\mathbf{R}}\|_F = \sqrt{\sum_i \sum_j |R_{ij} - \hat{R}_{ij}|^2}. \quad (43)$$

In this case, the  $E_b/N_0 = 12$  dB scenario is considered. As can be seen, the LVQ algorithm yields a faster convergence and smaller deviation from the true channel matrix  $\mathbf{R}$  because this is a supervised method. The decision rule results in further difference in the learning performance as the detected

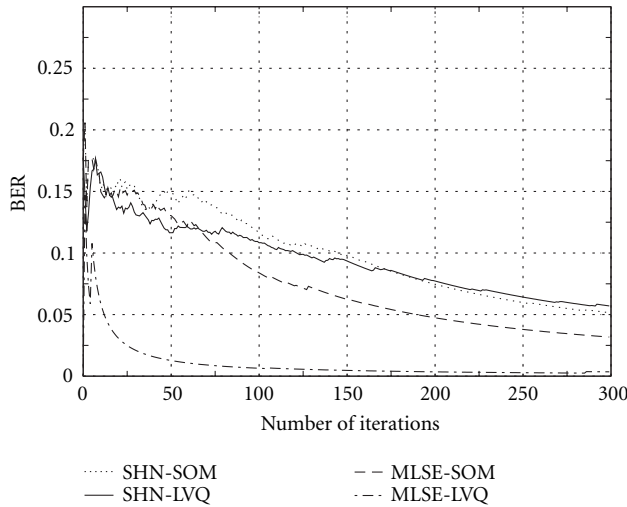


FIGURE 8: The synchronous case: cumulated BER versus number of iterations.

symbols are needed both by the SOM algorithm (39) and to reconstruct the Voronoi cells in the case of LVQ (37). Because of this effect, the learning algorithm (let it be either LVQ or SOM) exhibits slower convergence when working with SHN than working with MLSE.

Figure 8 illustrates the cumulated BER as the function of iterations for  $E_b/N_0 = 12$  dB. As the figure demonstrates, the BER decreases with time as the Frobenius norm of deviation becomes smaller and smaller. We conclude that LVQ algorithm provides the fastest adaptation, however, there is only a slight difference between SOM and LVQ in the performance of SHN. This implies that unsupervised learning can yield almost as good performance as the supervised one in the case of SHN. In the sequel, the asynchronous scenario is considered where the MLSE method is impossible to simulate. To save computational time, only the unsupervised SHN-SOM method is simulated which is expected to give almost similar performance as the one of the supervised SHN-LVQ algorithm.

## 7.2. Asynchronous case

The asynchronous uplink of a DS-CDMA transmission system is considered. BPSK modulation with carrier frequency 5.2 GHz, chip duration  $T_c = 31.25$  ns, and measurement bandwidth 128 MHz (which is equal to four samples per chip), we assumed mobile velocity 3 m/s, time invariant channel impulse response. For  $e(t)$ , we used root raised cosine with roll-off parameter 0.22, the processing gain was set to  $PG = 16$  in all the simulations. We simulated  $K = 16$  users with both Walsh-Hadamard and extended Gold code sets. The delay configuration was  $\theta_k = (k - 1)T_c$  for the  $k$ th user.

We have investigated the behavior of this complex structure in two typical types of indoor channel models [29]:

- (i) HPLOS: hard partitioned (HP) office scenario with line of sight (LOS) propagation, average RMS delay

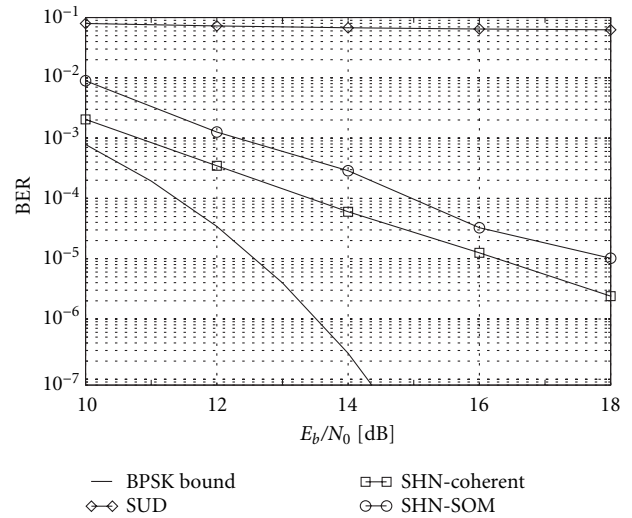


FIGURE 9: The HPLOS channel with extended Gold codes.

- spread 27 ns, maximum RMS delay spread 144 ns,
- (ii) OPOBS: open plan (OP) scenario without line of sight (obstructed topography—OBS) propagation, average RMS delay spread 109 ns, maximum RMS delay spread 191 ns.

The simple AWGN was omitted here as the other models present more relevant channel scenarios. Based on (15) calculating the discrete-time channel matrices for these cases, the  $\mathbf{R}[i]$  values are not equal to zero for any  $i$  due to the infinite nature of  $e(t)$ . However, for  $|i| > 6$  they are negligible, and thus they can be omitted. The resulting discrete-time channel matrices can be checked at [30]. For all four scenarios, only the SOM adaptation method is considered with 100 iterations and  $\eta^{[z]} = (1 - \tau/100)\eta^{[0]}$ , where  $\tau$  refers to the iteration instance. For the detector part, only SHN was investigated due to the unmanageable computational requirements of the MLSE. In the neurons, the same  $F(x)$  distribution function was used (29) but with parameter  $\gamma[\ell] = -2.5 \cdot \ell$ , and only 100 iterations were performed.

In Figure 9, BER is depicted as a function of  $E_b/N_0$  in the case of asynchronous HPLOS channel with extended Gold codes. We can see that SHN with SOM and SHN-coherent yields a reasonable detection performance. Of course, SHN-SOM is slightly worse than SHN-coherent, it needs an additional 2 dB payoff in SNR to achieve the same performance. This stresses the advantage of SHN detection with SOM, as only a marginal increase in SNR yields the same performance as SHN-coherent. Thus, LVQ adaptation has not been simulated for the asynchronous scenarios to save computational time. As can be seen in the figure, SUD also results in a very poor performance in this case. Other detection schemes such as MLSE were not implemented due to the high computational complexity required by the algorithm.

Figure 10 depicts the same as Figure 9, only the channel model is different (OPOBS instead of HPLOS). The results show the same tendencies as elaborated in explaining the previous figure.

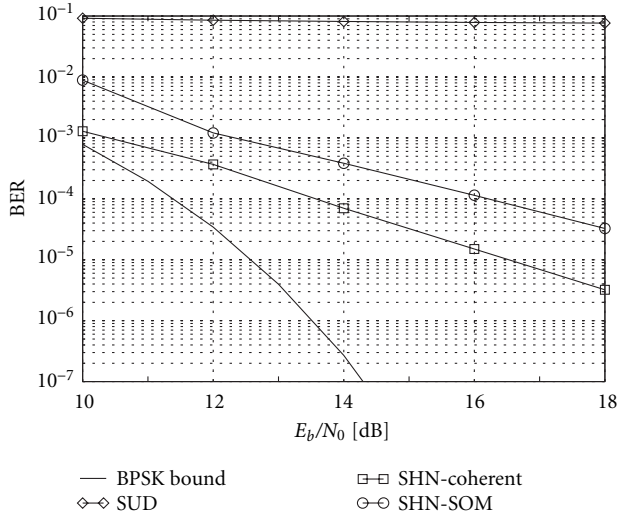


FIGURE 10: The OPOBS channel with extended Gold codes.

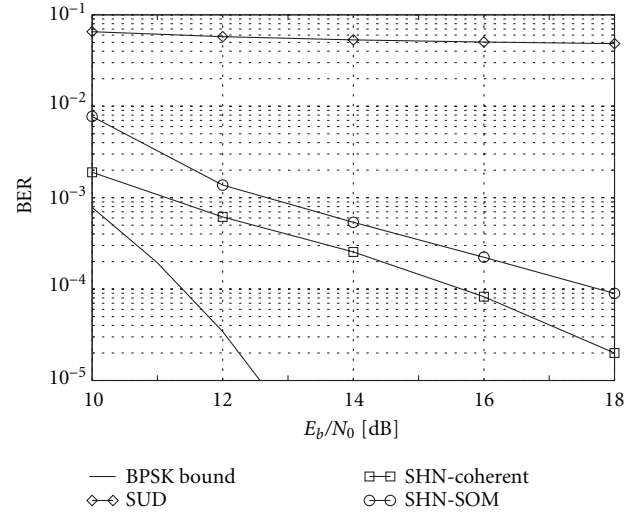


FIGURE 12: The OPOBS channel with Walsh-Hadamard codes.

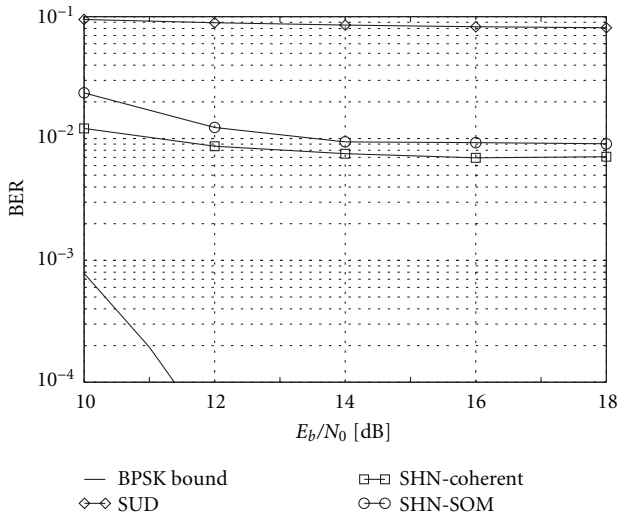


FIGURE 11: The HPLOS channel with Walsh-Hadamard codes.

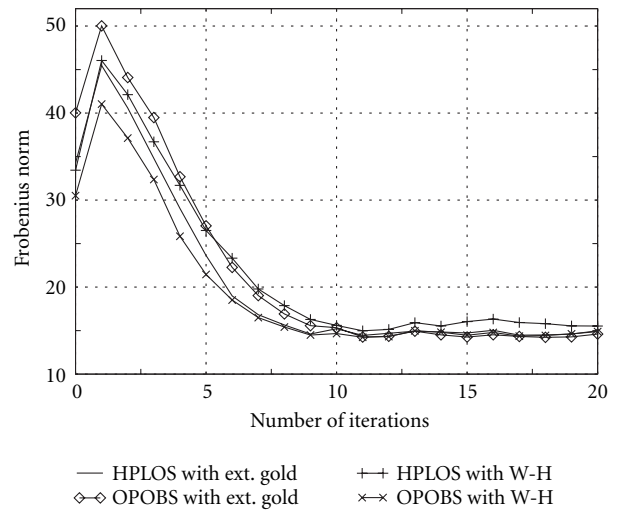


FIGURE 13: Frobenius norm versus number of iterations in the asynchronous case.

Figures 11 and 12, respectively shows the BER with respect to  $E_b/N_0$  in the case of applying Walsh-Hadamard codes. Since these codes exhibit worse correlation properties in asynchronous case than the extended Gold codes [31], the performance has deteriorated. However, the same tendency can be observed as in the previous figures (but BER is definitely higher).

Figure 13 shows the Frobenius norm of the deviation between the true discrete-time channel matrix  $\mathbf{R}$  and its estimated version  $\hat{\mathbf{R}}$  given in (43). In all cases, SHN detector with SOM adaptation method is applied. The transient behavior of the norm can be studied based on four different channel models such as HPLOS with extended Gold codes, HPLOS with Walsh-Hadamard codes, OPOBS with extended Gold codes and OPOBS with Walsh-Hadamard codes, respectively.

However, in all cases the adaptation method reaches quickly (in approximately ten iterations) a minimal Frobenius norm around 10 independently of the code sequences and the propagation model. We may wonder that the convergence of the norm (i.e., the convergence of the adaptation algorithm) seems to be faster here than in the synchronous case in Figure 7. The reason lies in the parallel update of matrix elements in a block fashion in contrast with the synchronous case. As a result, the matrix elements are updated 250 times within one iteration in the asynchronous case due to the block size  $N_B = 250$ .

Further simulation results, including dynamic behaviour of the structure (for instance, change in the number of active users), can be found in [32].

## 8. COMPUTATIONAL COMPLEXITY

The computational complexity of the proposed detector is evaluated as the complexity of different blocks in Figure 4. The RAKE receiver is already implemented and widely used in spread spectrum systems. The only additional requirement related to RAKE receiver is the accessibility of the vector  $\tilde{\mathbf{b}}$  (i.e., soft output values) which is assumed to be implemented. To determine the computational requirements of the other three blocks, we must take into consideration some system parameters, for example, bit rate of communication, chip rates, block size, etc. We consider the third generation universal mobile telecommunication system where CDMA is already standardized as the basic modulation scheme. In our model, we assume packets at the input (18), thus we focus on one slot. Following the standard, one slot is defined to last for 0.625 ms containing 2560 chips. Each user chooses the rate of communication from 32 kbps up to 2 Mbps, thus the number of bits in one slot changes from 20 up to 1280. Of course, with higher speed communication, only less active users could be accommodated in the system. Namely, only two 2 Mbps users are allowed to transmit at the same time, and the maximal number is 128 at 32 kbps. However, in a fully loaded system, the task is to demodulate 2560 symbols per slot. It infers 2560 decisions per slot, equivalent to approximately 4 million decisions per second. From [23], it is clear that 20 iterations must be sufficient. It results in approximately 80 million iterations per second. Thus, to generate random values for internal noise sources in the neurons, the noise chip must be capable of generating 80 million values per second in the upper part of Figure 4. This is an easy task for today's DSP devices.

The stochastic Hopfield network should process these noise values, that is, the SHN must compute 80 million iterations per second. In one iteration, each neuron computes the updating rule defined in (27) which entails 2559 multiplications and 2561 additions. This adds up to 5120 operations per iteration per neuron. The resulting computational requirement is approximately 410 billion operations per second which seems to be a huge number. However, it can be handled with distributed computing. Namely, we can take into account that

- (1) we can assign one specific DSP device to every symbol of each user, that is, one DSP for every neuron;
- (2) there are zero elements in the matrix  $\mathbf{W}$ , that is, the intersymbol interference occurs locally, thus only the neighboring symbols can overlap each other. Zero elements in  $\mathbf{W}$  decrease the required computation.

Depending on the circumstances (available hardware infrastructure, network load), we can apply one DSP also for more neurons, then our results must be multiplied by the number of neurons computed by the DSP device. Based on the second condition  $3 \times 128$ , symbols interfere in the worst case (128 users on the channel) which entails 383 multiplication, 385 addition, and one random number generation in each neuron equivalently 769 operations per slot per iteration. Using the above mentioned values approximately 25

million operations per second per neuron are required for SHN MUD. For instance, Texas TMS 320c 5502 neuron integrated circuit is capable of reaching 100 million iterations per second, thus four neurons fit in one IC. On the other hand, it is very cheap (approximately \$8), which yields a low cost implementation of the proposed detection scheme.

The HyNN represents the same complexity, only the decision function differs, which does not need special attention. As has already been seen, the adaptation method contains linear operations at the end of every packet, resulting in the update of matrix  $\hat{\mathbf{R}}$ , which comprises  $2560 \times 2560$  elements. For the update of each component, 2562 multiplications and two additions are required at every 0.625 ms (see, e.g., (39)). This results in approximately 4 million operations per second per matrix element, thus all elements cannot be updated using the same hardware; distributed architecture is needed for the adaptation method.

## 9. CONCLUSIONS AND FURTHER WORKS

In this paper, a novel MUD scheme was presented based on the theory of neural networks and adaptive detectors. The new detector is also capable of serving many users in heavily loaded environments. The robustness of the method lies in the application of the stochastic recurrent neural network and the self-organizing feature map. As an improvement, a control unit is also proposed which could switch between stochastic and hysteretic operations depending on the circumstances. However, the rules of how to do so have not yet been investigated. The system has been tested by extensive simulations. Simulation results are promising and they show that the performance of the proposed scheme is close to the one of the optimal detector.

For further work, the authors plan to provide a more rigorous mathematical analysis of this system, including the cooling schedule, which plays an important role in the stochastic recurrent neural network. Furthermore, a separate SOM seems to be feasible in the controller part too which could recognize what operation the detector should follow; stochastic, hysteretic, or both. For doing this, continuous adaptation will be required.

## ACKNOWLEDGMENTS

The research reported here was partly supported by the project T025964 granted by the Hungarian National Scientific Foundation and the project GOA/98/06 provided by the Katholieke Universiteit Leuven. The authors are also grateful for the COST 262 project.

## REFERENCES

- [1] S. Verdú, *Multiuser Detection*, Cambridge University Press, Cambridge, UK, 1998.
- [2] S. Verdú, "Minimum probability of error for asynchronous Gaussian multiple-access channels," *IEEE Transactions on Information Theory*, vol. 32, no. 1, pp. 85–96, 1986.
- [3] J. Levendovszky, W. Mommaerts, and E. C. van der Meulen, "Hysteretic neural networks for global optimization of

- quadratic forms,” *Neural Network World*, vol. 2, no. 5, pp. 475–496, 1992.
- [4] G. Jeney and J. Leventovszky, “Stochastic hopfield network for multi-user detection,” in *European Conf. Wireless Technology*, pp. 147–150, Paris, France, 2000.
- [5] B. Aazhang, B.-P. Paris, and G. C. Orsak, “Neural networks for multi-user detection in code-division multiple-access communications,” *IEEE Trans. Communications*, vol. 40, no. 7, pp. 1212–1222, 1992.
- [6] L. Castedo and O. Macchi, “Maximizing the information transfer for adaptive unsupervised source separation,” in *Proc. IEEE Workshop on Signal Processing Advances in Wireless Communications*, pp. 65–68, Paris, France, April 1997.
- [7] G. He and A. C. Kot, “A new adaptive neural network multiuser detector in synchronous CDMA systems,” in *Proc. IEEE Int. Conf. Acoustics, Speech, Signal Processing*, vol. 4, pp. 3333–3336, Munich, Germany, April 1997.
- [8] Y. Sun, “Eliminating-highest-error and fastest-metric-descent criteria and iterative algorithms for bit-synchronous CDMA multiuser detection,” in *Proc. IEEE International Conference on Communications*, vol. 3, pp. 1576–1580, Atlanta, Ga, USA, June 1998.
- [9] W. Neng, Z. Baoyu, and Z. Weiping, “A blind multiuser detector based on neural network,” in *Int. Conf. on Comm. Techn. (ICCT’98)*, vol. 2, pp. S44-01-1–S44-01-5, Beijing, China, October 1998.
- [10] C. Valadon and R. Tafazolli, “A fast adaptive multiuser detector for DS-CDMA communications based on an artificial neural network architecture,” in *IEEE Fifth International Symposium on Spread Spectrum Techniques & Applications*, vol. 3, pp. 873–877, Sun City, Republic of South Africa, September 1998.
- [11] U. Mitra and H. V. Poor, “Neural network techniques for adaptive multi-user demodulation,” *IEEE Journal on Selected Areas of Communications*, vol. 12, no. 9, pp. 1460–1470, 1994.
- [12] J. J. Hopfield and D. W. Tank, “Neural computation of decision in optimization problems,” *Biological Cybernetics*, vol. 52, pp. 141–152, 1985.
- [13] J. J. Hopfield, “Neurons with graded response have collective computational properties like those of two-state neurons,” *Proc. Nat. Acad. Sci. U.S.A.*, vol. 81, no. 10, pp. 3088–3092, 1984.
- [14] S. Haykin, *Neural Networks: A Comprehensive Foundation*, Prentice-Hall, Upper Saddle River, NJ, USA, 2nd edition, 1999.
- [15] T. Miyajima, T. Hasegawa, and M. Haneishi, “On the multiuser detection using a Neural network in code division multiple access communications,” *IEICE Trans. on Communication*, vol. E76-B, no. 8, pp. 961–968, 1993.
- [16] G. I. Kechriotis and E. S. Manolakos, “Implementing the optimal CDMA multiuser detector with hopfield neural networks,” in *Proc. International Workshop on Applications of Neural Networks to Telecommunications*, pp. 60–66, Princeton, NJ, USA, October 1993.
- [17] W. G. Teich and M. Seidl, “Code division multiple access communications: multi-user detection based on a recurrent neural network structure,” *IEEE Trans. Vehicular Technology*, vol. 46, pp. 979–984, July 1996.
- [18] M. K. Varanasi and B. Aazhang, “Multistage detection in asynchronous code-division multiple access communications,” *IEEE Trans. Communications*, vol. 38, no. 4, pp. 509–519, 1990.
- [19] S. H. Yoon and S. Rao, “Multiuser detection in CDMA based on the annealed neural network,” in *IEEE Int. Conf. Neural Networks*, vol. 4, pp. 2124–2129, Washington, DC, USA, June 1996.
- [20] D. C. Chen, B. J. Sheu, and T. W. Berger, “A compact neural network based CDMA receiver for multimedia wireless communication,” in *Proc. IEEE International Conference on Computer Design*, pp. 99–103, Austin, Tex, USA, October 1996.
- [21] G. I. Kechriotis and E. S. Manolakos, “A hybrid digital signal processing-neural network CDMA multiuser detection scheme,” *IEEE Trans. on Circuits and Systems II: Analog and Digital Signal Processing*, vol. 43, no. 2, pp. 96–104, 1996.
- [22] B. Wang, Z. He, and J. Nie, “To implement the CDMA multiuser detector by using transiently chaotic neural network,” *IEEE Trans. on Aerospace and Electronics Systems*, vol. 33, pp. 1068–1071, July 1997.
- [23] G. Jeney, J. Leventovszky, S. Imre, and L. Pap, “Comparison of different neural network based multi-user detectors,” in *EU-NICE 2000*, pp. 117–123, Enschede, Netherlands, September 2000.
- [24] G. Jeney, S. Imre, L. Pap, A. Engelhart, T. Dogan, and W. G. Teich, “Comparison of different multiuser detectors based on recurrent neural networks,” in *COST 262 International Workshop—Multiuser Detection in Spread Spectrum Communications*, pp. 61–70, Schloss Reisenburg, Germany, January 2001.
- [25] T. Kohonen, “The self-organizing map,” *Proceedings of the IEEE*, vol. 78, no. 9, pp. 1464–1480, 1990.
- [26] J. A. Kangas, T. K. Kohonen, and J. Laaksonen, “Variants of self-organizing maps,” *IEEE Transactions on Neural Networks*, vol. 1, no. 1, pp. 93–99, 1990.
- [27] A. Hottinen, “Self-organizing multiuser detection,” in *IEEE Int. Symposium on Spread Spectrum Techniques and Applications*, pp. 152–156, Oulu, Finland, July 1994.
- [28] J. G. Proakis, *Digital Communications*, McGraw-Hill, New York, USA, 3rd edition, 1995.
- [29] D. M. Krizman, T. S. Rappaport, B. J. Ellison, and S. Y. Seidel, *SIRCIM Plus: Simulation of Indoor Radio Channel Impulse Response Models with Impulse Noise, Version 2.0*, MPRG/Virginia Tech Intellectual Properties, Blacksburg, Va, USA, 1997.
- [30] T. Dogan, J. Zheng, A. Engelhart, M. A. Dangel, and H. E. Schummer, “Visualization of the discrete-time channel matrix,” <http://it.e-technik.uni-ulm.de/Rmatrix/>.
- [31] A. Engelhart, H. E. Schummer, W. G. Teich, J. Lindner, H. D. Schotten, and H. Elders-Boll, “Investigation of selected spreading sequences for DS-CDMA systems with a nonlinear multiuser detector,” in *COST262 MCM meeting*, TD 018(99), Thessaloniki, Greece, July 1999.
- [32] G. Jeney, J. Leventovszky, and L. Kovács, “Blind adaptive stochastic neural network for multiuser detection,” in *IEEE Vehicular Technology Conference*, vol. 3, pp. 1868–1872, Rhodes, Greece, May 2001.

---

**Gábor Jeney** received his M.S. degree in 1998 from the Technical University of Budapest in the field of mobile and optical telecommunications. Since that time, he has been working on his Ph.D. at the Department of Telecommunications in the Budapest University of Technology and Economics. Since 2000, he has been an Assistant Professor at the same department. Since 1999, he is a member of the IEEE, and recently he has been involved in the COST 262 action. His main research interest covers mobile communications, wide-band systems, neural networks, MUD, and so forth.



**János Levendovszky** received the M.S. degree in electrical engineering at the Budapest University of Technology and Economics, Hungary, in 1986. He obtained his Ph.D. degree (Candidate of Science) from the Hungarian Academy of Sciences in 1989 in the field of adaptive signal processing. Since then, he has been visiting scholar at Oxford University, UK, and conducted research on neural network theory at the Department of Mathematics, Katholieke Universiteit Leuven, Belgium. Presently, he is an Associate Professor of electrical engineering at the Department of Telecommunications at the Budapest University of Technology and Economics. He teaches and researches in the area of information and communication theory, networking, and soft computing.



**László Pap** graduated in 1967 from the Telecommunication Branch of the Electrical Engineering Faculty at the Technical University of Budapest. He received the candidate of sciences and the academic doctor of sciences degrees in 1980 and 1992, respectively. Since 1992, he has been a Professor and Head of the Telecommunications Department, and since 1994, he has been the dean of the Electrical Engineering and Informatics Faculty. His main interest includes communication theory, encoding, modulation, synchronization, further mobile communications, and the theory of modern telecommunication systems.



**E. C. van der Meulen** received his M.S. degree in mathematics from the University of Leiden, The Netherlands, in 1962 and his Ph.D. degree in statistics from the University of California, Berkeley, in 1968. In 1993, he received the Doctor honoris causa degree from the Technical University of Budapest, Hungary. From 1968 to 1975, he served on the faculty of the Department of Statistics at the University of Rochester, Rochester, NY. From September 1972 to January 1974, he was on leave at the Mathematical Center, Amsterdam, The Netherlands. He has been a Professor of Mathematics at the Katholieke Universiteit Leuven, Belgium, since September 1975. His research interests include multiuser information theory and information-theoretic statistical inference. He is a member of the editorial board of the American Journal of Mathematical and Management Sciences. Dr. van der Meulen is a member of IMS, ISI, and the Dutch Society for Statistics. Since 1990, he has served as Chairman of the Information Theory Chapter in the Benelux Section of the IEEE. The American Journal of Mathematical and Management Sciences awarded him the 1997 Jacob Wolfowitz Prize. In 1998, the Czech Academy of Sciences awarded him the B. Bolzano Honorary Medal. He is a Fellow of the IEEE.

