This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the WCNC 2010 proceedings.

A reduced complexity Soft-Input Soft-Output MIMO detector combining a Sphere Decoder with a Hopfield Network

Daniel J. Louw, Philip R. Botha, Bodhaswar T.J. Maharaj
Department of Electrical, Electronic and Computer Engineering, University of Pretoria, Pretoria, South-Africa
Email: djlouw@tuks.co.za, philip.botha@mweb.co.za, sunil.maharaj@up.ac.za

***Abstract*—In this paper, a reduced complexity soft-input soft-output MIMO detector is presented. The detector is intended to be used in conjunction with an error correction code. The detector combines a Sphere Decoder with a Hopfield network to calculate a max-log-map approximation. It is then combined with the error correction code in an iterative structure (turbo). The code used is a quasi-cyclic non-binary LDPC code. The simulation results demonstrate that with less computational complexity, the proposed system's performance equals that of an optimal sphere decoder based detector.**

***Index Terms*—Sphere Decoder, Hopfield Network, MIMO, Space-time codes, Space-frequency codes, Space-time-frequency codes**

I. Introduction

The increasing demand for cheap, high speed, reliable wireless communications has led to many innovations over the past several years. The use of multiple antennas at both the transmitter and the receiver has resulted in substantially higher channel capacity. These multiple input multiple output (MIMO) wireless systems combined with Orthogonal Frequency Division Multiplexing (OFDM) have allowed for the easy transmission of symbols over time, space and frequency.

The performance of systems in a multipath fading environment can be improved by exploiting the diversity available in the channel. In order to extract diversity from the channel, different coding schemes have been developed. The seminal example is the Alamouti Space Time Block (STB) code [1] which could extract the spatial diversity. Many other codes have also been proposed [2], [3] which have been able to achieve some or all of the available diversity in the channel at various transmission rates.

The most advanced codes that have been proposed are able to extract full diversity while achieving a transmission rate equal to the number of transmit antennas [4]. These codes use Linear Pre-coding (LP) and the received symbols must be jointly decoded at the receiver. Generally a Sphere Decoder (SD) is used. In order to approach capacity an error correction code (ECC) is often added. The ECC requires soft outputs from the SD. Methods to produce soft outputs for SD's fall into two categories: List Sphere Decoding [5] and Max-log-map Hard-to-Soft decoding [6]. In this paper Hard-to-Soft decoding is considered. This method uses a hard output SD to produce a MAP solution. A hard SD iteration is then run for each bit in the symbol to produce a counter hypothesis to the MAP solution.

In this paper a Hopfield network is used in place of the SD to create the counter hypothesis. While the Hopfield network is sub-optimal, it is significantly less complex than the SD. The receiver is then linked in an iterative turbo structure, but due to the low complexity of the Hopfield network the overall complexity of the receiver is still less than when the SD is used. The turbo structure allows this SD-Hopfield based detector to equal the performance of the pure SD detector (that has not been linked in a turbo structure) at a reduced complexity.

This paper is structured as follows: In Section II the MIMO detector is described and in Section III the Hopfield Network is described. Section IV discusses the Turbo structure and the general system. A complexity analysis is presented in Section V. The simulation parameters and results are given in Section VI. Section VII concludes the paper.

Notation: In this paper we use the following notation. Column vectors (matrices) are denoted by boldface lower (upper) case letters. Superscripts $\mathcal{T}$ and $\mathcal{H}$ denote the transpose and Hermitian transpose operations, respectively; $diag(d_1 \dots d_P)$ denotes a $P \times P$ diagonal matrix with diagonal entries $d_1 \dots d_P$. $\mathbf{F_P}$ is the $P \times P$ discrete Fourier transform (DFT) matrix. $\lceil x \rceil$ represents the smallest integer larger than or equal to x.

II. MIMO Detection

The received signal in a MIMO system may be described as:

$$\mathbf{y} \quad = \quad \mathbf{Hx} + \mathbf{n} \qquad (1)$$

where $\mathbf{H}$ is the equivalent channel matrix, $\mathbf{x}$ is the transmitted data vector and $\mathbf{n}$ is the noise vector. In general $\mathbf{H}$, $\mathbf{x}$, $\mathbf{y}$ and $\mathbf{n}$ are complex. However, a complex to real conversion may be performed to allow the SD and the Hopfield network to consider only real quantities. The optimal detection of the transmitted data requires the calculation of the maximum a posteriori probability (MAP) that a bit equals a one (or a zero). This probability is generally expressed as a log likelihood ratio (LLR). Using the max-log-map approximation, this LLR (Λ_i^p)

978-1-4244-6398-5/10/$26.00 ©2010 IEEE

can be given as [6]:

$$\begin{aligned}\Lambda_i^p &= LLR(x_i|\mathbf{H},\mathbf{y}) \quad (2)\\ &\approx \min_{x\in X_i^1}\left\{\frac{||\mathbf{y}+\tilde{\mathbf{y}}-\mathbf{H}\mathbf{x}||^2}{N_0}\right\} - \\ &\quad \min_{x\in X_i^{-1}}\left\{\frac{||\mathbf{y}+\tilde{\mathbf{y}}-\mathbf{H}\mathbf{x}||^2}{N_0}\right\} \quad (3)\end{aligned}$$

where $\tilde{\mathbf{y}}$ is the vector which solves equation 4:

$$\mathbf{\Lambda^a} = 2\mathbf{H}^T\tilde{\mathbf{y}}, \quad (4)$$

where $\mathbf{\Lambda^a}$ is the a-priori information on the sequence $\mathbf{x}$ in the LLR format and X_i^s is the set of all possible $\mathbf{x}$ vectors for which $x_i = s$. Solving this equation is done by first calculating the hard ML solution for the entire sequence using a SD. The SD is then run again, once for each bit, to calculate a counter hypothesis. A counter hypothesis is the most likely sequence with the bit in question fixed at the opposite of the ML solution. The ML sequence and the counter hypotheses are then used in equation 3 to calculate the soft output. This method will be referred to as the Soft-Output SD (SO-SD).

III. Hopfield Network

Hopfield networks consist of interconnected neurons. The neurons compute an output $v_i(t)$ from a signal $u_i(t)$ as follows:

$$v_i(t) = g[\beta u_i(t)] \quad (5)$$

where i is the indice of the neuron, β is the gain and $g(x)$ is a smooth monotonically increasing sigmoid function. The signal $u_i(t)$ obeys the following equation [7]:

$$\frac{du_i(t)}{dt} = -\frac{u_i(t)}{\tau} + \sum_{j=1}^{N} T_{ij}v_j(t) + I_i \quad (6)$$

where τ is a constant that is set equal to unity and I_i is a constant bias added to the input of a neuron. T_{ij} are elements of a symmetric connectivity matrix that are zero when $i = j$ because each neuron connects to all other neurons but not to itself. Thus:

$$\begin{aligned}T_{ij} &= T_{ji} \quad (7)\\ T_{ii} &= 0 \ \forall\ i\end{aligned}$$

Hopfield showed that when (7) is met and the neurons are operating in a high gain mode (β is large) the stable states of the network are the local minima of the computational energy (E) of the network:

$$E = -\frac{1}{2}\mathbf{v}^T\mathbf{T}\mathbf{v} - \mathbf{v}^T\mathbf{I} \quad (8)$$

Using a Hopfield network to calculate a minimisation of some kind (for example ML detection) is done by writing the ML equation in such a way as to link the variables in the ML equation with terms in equation 8 [8], [9].

A. Derivation for the use of a Hopfield network in MIMO detection.

For ML detection, the transmitted vector $\mathbf{x}$ must be found which minimises the following equation:

$$ML(\mathbf{x}) = (\mathbf{y}-\mathbf{H}\mathbf{x})^T(\mathbf{y}-\mathbf{H}\mathbf{x}) \quad (9)$$

This equation can be expanded as follows:

$$ML(\mathbf{x}) = \mathbf{y}^T\mathbf{y} + \mathbf{x}^T\mathbf{H}^T\mathbf{H}\mathbf{x} - 2\mathbf{x}^T\mathbf{H}^T\mathbf{y} \quad (10)$$

The first term in the above equation is a constant for all possible transmitted sequences and may thus be removed from the ML equation. In order to match the ML equation to equation 8 the $\mathbf{H}$ matrix will be expanded with summations to yield the following:

$$ML(\mathbf{x}) = -2\sum_{i=1}^{N} x_i\mathbf{H_i^T}\mathbf{y} + \sum_{i=1}^{N}\sum_{j=1}^{N}\mathbf{H_i^T}\mathbf{H_j}x_ix_j \quad (11)$$

where $\mathbf{H_i}$ represents the i^{th} column of $\mathbf{H}$, x_i represents the i^{th} transmitted symbol and N is the number of transmitted symbols. For BPSK type modulation, the values of $\mathbf{x}$ must be either 1 or -1. The neuron function must thus meet the requirement:

$$\begin{aligned}g(-\infty) &= -1 \quad (12)\\ g(0) &= 0\\ g(\infty) &= 1\end{aligned}$$

One example of such a function is the hyperbolic $\tan$ function. This requirement allows one to add the following term to equation 11:

$$-\sum_{i=1}^{N}\mathbf{H_i^T}\mathbf{H_i}(x_i^2-1) \quad (13)$$

This term is created by finding a polynomial with roots equal to -1 and 1. Since x_i is always 1 or -1 this term will thus always be zero and will not affect the ML equation. We also remove the $i = j$ case from the double summation term in equation 11 and the equation is then simplified as follows:

$$\begin{aligned}ML(\mathbf{x}) &= -2\sum_{i=1}^{N} x_i\mathbf{H_i^T}\mathbf{y} + \sum_{i=1}^{N}\sum_{\substack{j=0\\ j\neq i}}^{N}\mathbf{H_i^T}\mathbf{H_j}x_ix_j \quad (14)\\ &\quad -\sum_{i=1}^{N}\mathbf{H_i^T}\mathbf{H_i}x_i^2 + \sum_{i=1}^{N}\mathbf{H_i^T}\mathbf{H_i} + \sum_{i=1}^{N}\mathbf{H_i^T}\mathbf{H_i}x_i^2\end{aligned}$$

The third and fifth terms cancel in equation 14 and the fourth term is a constant for all possible $\mathbf{x}$ and can thus be omitted. The final ML equation is thus :

$$ML(\mathbf{x}) = \sum_{i=1}^{N}\sum_{\substack{j=0\\ j\neq i}}^{N}\mathbf{H_i^T}\mathbf{H_i}x_ix_j - 2\sum_{i=1}^{N}x_i\mathbf{H_i^T}\mathbf{y} \quad (15)$$

One can now match the parameters in the ML equation (15) to those in the energy equation (8). This is done by defining the following:

$$T_{ij} = \begin{cases} -2\mathbf{H_i^T H_j} & if \quad i \neq j \\ 0 & if \quad i = j \end{cases} \tag{16}$$

$$\mathbf{I} = 2\mathbf{H^T y}. \tag{17}$$

This definition results in a symmetric $\mathbf{T}$ matrix where $T_{ij} = T_{ji}$. Substituting these definitions into equation 15 yields the following:

$$ML(\mathbf{x}) = -\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} T_{ij}x_i x_j - \sum_{i=1}^{N} x_i I_i \tag{18}$$

$$= -\frac{1}{2}\mathbf{x}^T\mathbf{T}\mathbf{x} - \mathbf{x}^T\mathbf{I}, \tag{19}$$

which is exactly the same as the energy equation when $\mathbf{x} = \mathbf{v}$. Thus, once the Hopfield network has converged, the transmitted data sequence can be read as the output of the neurons. The equation describing the convergence of the network (equation 6) is in continuous time. A discrete time version for use in a digital system is given below:

$$u_i[t+1] = \mathbf{T}_i\mathbf{v}[t] + I_i \tag{20}$$

$$v_i[t+1] = g(\beta u_i[t+1]), \tag{21}$$

where $\mathbf{T}_i$ represents the i^{th} row of $\mathbf{T}$. While this derivation was done for BPSK, the same equations hold for QPSK if a complex to real conversion is performed. Higher order modulation techniques can also be used with the method described in [6].

B. Heuristic Improvements to the Hopfield network

The performance of the Hopfield network may be improved by several heuristic methods. The first considers the update procedure of the network. In equation 20 all the neurons are updated at the same time (batch update). Better performance can be achieved using sequential update where the neurons are updated one at a time and the new outputs are used for updates of the other neurons. Thus, equation 20 is calculated for i from 1 to N and $v_j[t+1]$ is used instead of $v_j[t]$ if $j < i$. The second heuristic is the inclusion of a momentum term. Normally $\tau = 1$ (see equation 6) which means that $u_i[t]$ does not affect $u_i[t+1]$ directly. A momentum term is added by not setting τ equal to 1 and results in $u_i[t+1]$ depending on $u_i[t]$. This changes equation 20 to be:

$$u_i[t+1] = \alpha u_i[t] + \mathbf{T}_i\mathbf{v}[\mathbf{t}] + \gamma I_i \tag{22}$$

where $\alpha = \left(1 - \frac{1}{\tau}\right)$, and γ is a scaling term which may be varied along with α to change the relative contributions of the three terms in order to improve performance [10]. The third heuristic used to improve performance is called annealing. Annealing has been shown to improve the performance of many kinds of network based optimisation techniques and specifically Hopfield networks [11], [12]. In an annealing procedure, variable values are changed from one iteration to the next. The most important parameter to anneal is the gain of the neuron (β). Initially, the gain is set low to allow the neurons to stay in a less determined zone (and to keep the algorithm from getting stuck in a local minima). The gain is then increased with each iteration to force the algorithm to converge. Annealing may also be used on the other scaling factors introduced above. In order to find good initial values for the parameters as well as an annealing procedure, the parameters were optimised by using a genetic algorithm to minimise the metric:

$$\|\mathbf{y} - \mathbf{H}\tilde{\mathbf{x}}\|, \tag{23}$$

where $\tilde{\mathbf{x}}$ is $\mathbf{x}$ with bit $\tilde{x}_i = -x_i$. The optimised values of the parameters and the annealing procedure were found to be:

$$\gamma = 0.672 + 0.2797\exp\left(0.0575t - 0.426\right) \tag{24}$$

$$\beta = 0.0267 + 0.1363\exp\left(0.0442t - 1.86\right) \tag{25}$$

$$\alpha = -0.137 + 0.369\exp\left(0.131t - 1.863\right), \tag{26}$$

with t being the iteration number.

IV. System Specifications

Linear pre-coding of the symbol constellation encodes the data symbols in a manner that extracts the most benefit from the diversity available in the channel. It does this by mapping a set of data symbols to a new set of encoded symbols that are transmitted. This process can be expressed as a matrix multiplication. Let $\mathbf{x} = [\mathbf{x_1}, \ldots, \mathbf{x_M}]^{\mathcal{T}}$ be a data vector of length M complex symbols from a modulation alphabet (eg. QPSK, M-QAM). Let $\mathbf{\Theta}$ be a unitary matrix of size $M \times M$. $\mathbf{\Theta}$ may be defined as [13]:

$$\mathbf{\Theta} = \mathbf{F_M^{\mathcal{H}}diag}(\mathbf{1}, \varphi, \ldots, \varphi^{\mathbf{M-1}}), \tag{27}$$

where $\varphi = \exp(j2\pi/4M)$. The linear pre-coding may now be expressed as:

$$\mathbf{s} = \mathbf{\Theta x} \tag{28}$$

where $\mathbf{s}$ is the resultant encoded vector of complex symbols. A linear pre-coded (LP) data vector ($\mathbf{s}$) permits the correct decoding of all of the data symbols ($\mathbf{x}$) encoded by the linear pre-coder upon the reception of a single encoded symbol (s_i), thus yielding diversity equal to the rank of the rotation matrix (R_Θ) [13]. For the remainder of this paper, we will assume the following:

- The channel will be characterised by frequency-selective block-fading conditions, where the channel remains constant for the duration of an OFDM symbol, but changes from one symbol to the next.
- The number of independent paths in the channel (L) is the same for every transmitter receiver pair.
- Perfect Channel State Information (CSI) is available at the receiver and no CSI is available at the transmitter.
- Perfect carrier and symbol synchronisation at the receiver.

The MIMO-OFDM system under consideration has N_T transmit and N_R receive antennas. For a given SF code, codewords will be placed over N_F frequency tones. Data in binary form is mapped to a symbol in a complex modulation constellation

This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the WCNC 2010 proceedings.

(eg. QPSK). Let $\mathbf{x} = [\mathbf{x_1}, \ldots, \mathbf{x_M}]^{\mathbf{T}}$ be a data vector of length M complex modulation symbols, where $M = N_T N_L$ and $N_L = 2^{\lceil \log_2 L \rceil}$ as in [4]. The data, $\mathbf{x}$, then undergoes linear pre-coding as described in equation 28. Placing these encoded symbols appropriately over space and frequency will result in a rate 1 code. However, a rate N_T code may be created by layering N_T such encoded vectors. Layering is achieved by multiplying the i^{th} vector by the diophantine number: ϕ^{i-1} [4]. For this paper we used

$$\phi = \varphi^{1/N_T}, \tag{29}$$

where $\varphi = \exp(j2\pi/4M)$. The number of frequency tones required can be given by $N_F = 2^{\lceil \log_2 N_T + \log_2 L \rceil}$. The OFDM time symbols are then created and transmitted. In this paper N_T and N_R were fixed at $N_T = N_R = 2$. The codes were designed for $L = 2$ resulting in $N_F = 4$ and the codes are represented using matrices. The rows correspond to frequency tones, while the columns represent the transmit antennas. The matrix representing the specific code used is given below:

$$\begin{pmatrix} s_1(1) & \phi s_2(1) \\ \phi s_2(2) & s_1(2) \\ s_1(3) & \phi s_2(3) \\ \phi s_2(4) & s_1(4) \end{pmatrix} \tag{30}$$

where $[s_i(1), \ldots, s_i(4)] = \Theta[x_i(1), \ldots, x_i(4)]$, $i = 1, 2$. Θ represents the 4×4 linear pre-coding matrix and i represents the i^{th} layer. This code may also be found in [4]. For this paper the complex symbols were taken from a QPSK modulation alphabet. An SF symbol thus contains 16 bits.

A. Error Correction Code

In principle any ECC could be used to demonstrate the performance of the SD-Hopfield decoder. In this paper a non-binary quasi-cyclic low density parity check (NB-QC-LDPC) code was chosen. The structure of the parity check matrix of the code was taken from [14]. However, the non-binary elements were chosen randomly from the nonzero elements of the finite field. The code parameters are given in table I. The decoder used is the fast fourier transform belief propagation (FFT-BP) algorithm [15]–[17].

TABLE I
NB-QC-LDPC CODE PARAMETERS

Code parameters	
Code Rate	$\frac{1}{2}$
Code Length (n) in bits	2304
Data Length (k) in bits	1152
Field size	256 $[GF(2^8)]$

B. Turbo System

The performance of the Hopfield network is suboptimal, thus the authors use an iterative joint detector and decoder (Turbo) structure to improve the performance of the system. A block diagram of the turbo receiver can be seen in Figure 1. The turbo process works by passing extrinsic information generated by one decoder to the other decoder to be used as a-priori information. In the first iteration the SD decodes the received signal and assumes the a-priori information is zero. The SD then passes the MAP solution to the Hopfield network which calculates a counter hypothesis solution for each bit. Using these counter hypotheses in the max-log-map equation given in 3 produces soft outputs which are passed to the FFT-BP decoder. The FFT-BP decoder runs a set amount of decoding iterations. Extrinsic information is calculated from the output of the FFT-BP decoder, which is then passed back to the SD. The SD then re-decodes the received information using the extrinsic information as a-priori information. The resultant soft outputs then represents the a-posteriori information. From this information the extrinsic information is calculated and passed to the BP decoder. This iterative process may be repeated as often as required. The extrinsic information ($\mathbf{\Lambda}_e$) is calculated as:

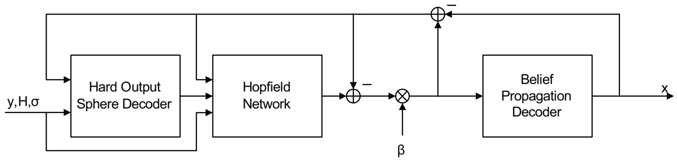

Fig. 1. SD-Hopfield Turbo System

$$\mathbf{\Lambda}_e = \mathbf{\Lambda}_p - \mathbf{\Lambda}_a. \tag{31}$$

Since the soft MIMO detector is sub-optimal, the extrinsic information coming from the Hopfield network will tend to be overoptimistic (too large). A simple method to counter this and improve the performance is to introduce a scaling factor. This has been shown to work in many turbo coded systems. An analysis to find a good scaling factor was done. The result can be seen in Figure 2 where the E_b/N_0 required to achieve a BER of 10^{-4} is plotted against different scaling factors. From the figure one can see that a scaling factor of 0.4 yields the best results.

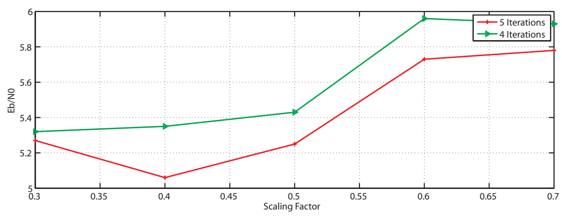

Fig. 2. SNR required to achieve a BER of 10^{-4} for different scaling factors.

V. Complexity Analysis

The complexity of the SD is difficult to provide analytically since it is not always the same and can depend heavily on the heuristic methods used to speed up the tree pruning process. The results provided in literature places the complexity between $O(n^6)$ and $O(n^3)$ in the best case scenarios. The complexity of the hopfield network is however very easy to calculate. The complexity calculation can be broken into two parts: the number of operations required for the initialization process and the number of calculations required per iteration of the network. For this analysis each multiplication or addition will be considered as a floating point operation (flop). The calculation of a transcendental function will be kept seperate and referred to as a tflop. These may be calculated using look up tables.

A. Initialization Phase

Setting up the connection weights requires the calculation of equations 16 and 17. This corresponds to one matrix on matrix multiply and one vector on matrix multiply. This complexity is given in Table II.

TABLE II
HOPFIELD NETWORK INITIALIZATION COMPLEXITY

Initialization Complexity		
Description	Operations	Units
Matrix × Matrix	n^3	$flops$
Matrix × Vector	$2n^2$	$flops$
Total	$n^3 + 2n^2$	$flops$

B. Iteration

Each Hopfield iteration requires the calculation of equation 22. This requires : one matrix on vector multiply, two scaler on vector multiplies and two vector additions. The annealing of the parameters requires: three transcendental calculations, six multiplies and six additions. An iteration also requires the calculation of the neuron function. This corresponds to one scaler by vector multiplication and one vector transcendental calculation. The number of $flops$ required per iteration will be denoted by C_{fiter} and the number of $tflops$ required per iteration will be denoted by C_{titer}. The complexity is given in Table III.

TABLE III
HOPFIELD NETWORK ITERATION COMPLEXITY

Complexity per Iteration		
Description	Operations	Units
Matrix × Vector	$2n^2$	$flops$
3 - Scaler × Vector	$3n$	$flops$
2 - Vector + Vector	$2n$	$flops$
Vector Transcendental	n	$tflops$
Annealing	3	$tflops$
Annealing	12	$flops$
C_{fiter}	$2n^2 + 5n + 12$	$flops$
C_{titer}	$n + 3$	$tflops$

C. Complexity Comparison

For the SO-SD detection, 50 iterations of the FFT-BP algorithm were run. For the SD-Hopfield detector 10 FFT-BP iterations were run per turbo iteration, resulting in a total of 50 FFT-BP iterations. The complexity of the FFT-BP decoding is thus not considered in this analysis. The total complexity of the SD-Hopfield detector can be expressed as:

$$C_{Total} = i_o(C(SD) + i_i(C_{fiter} + C_{titer}) \quad (32)$$
$$= i_oC(SD) + i_oi_i(C_{fiter} + C_{titer}) \quad (33)$$

where i_o is the number of turbo iterations and i_i is the number of internal Hopfield iterations. $C(SD)$ represents the complexity of a Sphere Decoding iteration. Notice that the initialisation complexity of the Hopfield detector has been omitted as the required equations will have been calculated by the SD. The complexity of the SO-SD detector (C_{SO-SD}) can be represented as:

$$C_{SO-SD} = (n + 1)C(SD) \quad (34)$$

The specific complexity gain will thus depend on the number of iterations and n. In this paper we chose $i_i = 10$, $i_o = 5$. The complexity gain can then be expressed as:

$$C_{gain} = C_{SO-SD} - C_{Total} \quad (35)$$
$$= (n + 1)C(SD) - 5C(SD) - 50(C_{fiter} + C_{titer}) \quad (36)$$
$$= (n - 4)C(SD) - 50(C_{fiter} + C_{titer}) \quad (37)$$

Considering that the complexity of the SD is cubic in n and the complexity of the Hopfield network is square in n, the gain is heavily dependant on n. In this paper $n = 16$ and the run-time for the SD-Hopfield detector was approximately 5 times faster than the SO-SD detector.

VI. SIMULATION RESULTS

The simulations were performed on the channel simulator developed in [18]. The sphere decoder used in the simulations was taken from [19]. The simulation parameters can be found in Table IV. Simulations were performed on an ideal two tap channel and a more realistic suburban alternative channel. The suburban alternative power delay profile (PDP) in Table IV can be found, and was applied to the simulator in [20]. Whilst the FFT size used is 128, of this only 48 carriers are used for in this paper, as that corresponds to one user. Twelve SF frequency symbols are grouped together and interleaved to form one 48 carrier grouping for a user. This grouping is then combined with a second user, pilot and guard bands to form the 128 carrier OFDM symbol.

TABLE IV
MIMO-WiMAX SIMULATION PARAMETERS

MIMO-OFDM parameters	
Transmit antennas	2
Receive antennas	2
FFT size	128
Number of sub-channels	2
Users per sub-channel	1
Mode	FUSC
Cyclic prefix length	0.25
Maximum doppler spread	$f_d = 100Hz$
Sampling Time	$T_s = 0.8\mu s$
Channel Bandwidth	1.25MHz
Transmit filter	Square root raised cosine, $\alpha = 0.5$
Receive filter	Square root raised cosine, $\alpha = 0.5$
Ideal Channel Parameters	
Frequency-Selectivity	Two-ray equal power PDP at 0 μs and 8 μs
Time-Selectivity	Slow-fading conditions with $f_d = 100Hz$
Realistic Channel Parameters	
Frequency-Selectivity	Suburban-alternative, 20 tap PDP
Time-Selectivity	Slow-fading conditions with $f_d = 100Hz$

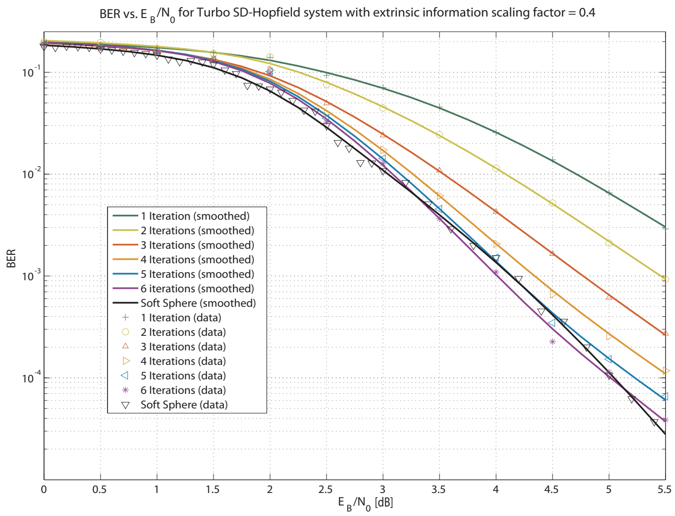

Fig. 3. Performance of SD-Hopfield detector on the rate-2 SF code with the NB-QC-LDPC code on a two tap channel.

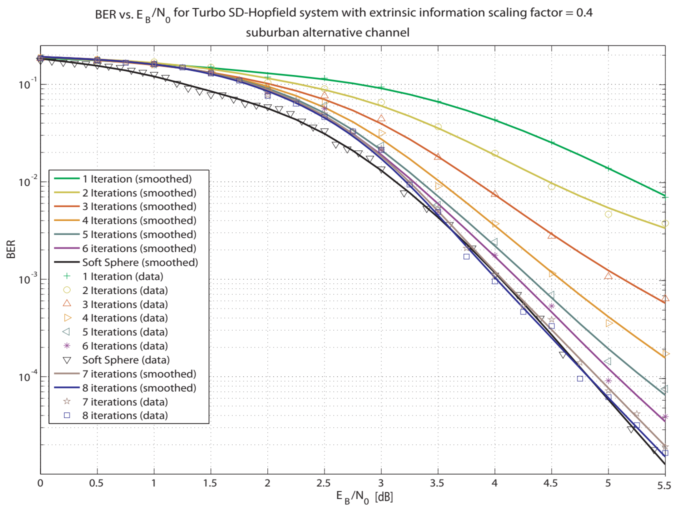

Fig. 4. Performance of SD-Hopfield detector on the rate-2 SF code with the NB-QC-LDPC code on the suburban alternative channel.

From the results in figure 3 one can see that the performance of the SD-Hopfield detector is approximately the same as that of the SO-SD detector. The performance of the SD-Hopfield detector improves with each iteration, but from the figure it can be seen that 5 iterations is sufficient to match the performance. Running more iterations allows the SD-Hopfield to beat the SO-SD detector. From Figure 4 one can see that the SD-Hopfield detector still performs well in realistic channel conditions, although more iterations are required to exactly equal the performance of the SO-SD detector.

VII. Conclusion

In conclusion, a reduced complexity soft output MIMO detector was developed which combines a SD with a Hopfield network in a turbo structure. The detector is able to achieve the same performance as a max-log-map optimal detector, using only SD's, at a lower complexity. Since the number of iterations in the turbo system can be changed, the proposed detector provides a simple mechanism to trade off performance for complexity.

References

[1] S. Alamouti, "A simple transmit diversity technique for wireless communications," *IEEE J. Select. Areas Commun.*, vol. 16, no. 8, pp. 1451–1458, October 1998.

[2] H. El Gamal and J. Hammons, A. R., "On the design of algebraic space-time codes for MIMO block-fading channels," *IEEE Trans. Inform. Theory*, vol. 49, no. 1, pp. 151–163, 2003.

[3] V. Tarokh, H. Jafarkhani, and A. R. Calderbank, "Space-time block codes from orthogonal designs," *IEEE Trans. Inform. Theory*, vol. 45, no. 5, pp. 1456–1467, 1999.

[4] W. Zhang, X. G. Xia, and P. C. Ching, "High-Rate Full-Diversity Space-Time-Frequency Codes for Broadband MIMO Block-Fading Channels," *IEEE Trans. Commun.*, vol. 55, no. 1, pp. 25–34, 2007.

[5] Y. de Jong and T. Willink, "Iterative tree search detection for mimo wireless systems," *IEEE Trans. Commun.*, vol. 53, no. 6, pp. 930–935, June 2005.

[6] R. Wang and G. Giannakis, "Approaching mimo channel capacity with soft detection based on hard sphere decoding," *IEEE Trans. Commun.*, vol. 54, no. 4, pp. 587–590, April 2006.

[7] J. Hopfield and D. Tank, "Computing with neural circuits: a model," *Science*, vol. 233, no. 4764, pp. 625–633, 1986. [Online]. Available: http://www.sciencemag.org/cgi/content/abstract/233/4764/625

[8] S. Bang, B. Sheu, and C.-H. Chang, "Maximum likelihood sequence estimation of communication signals by a hopfield neural network," in *Neural Networks, 1994. IEEE World Congress on Computational Intelligence., 1994 IEEE International Conference on*, vol. 5, Jun-2 Jul 1994, pp. 3369–3374 vol.5.

[9] R. Altes, "Unconstrained minimum mean-square error parameter estimation with hopfield networks," in *Neural Networks, 1988., IEEE International Conference on*, Jul 1988, pp. 541–548 vol.2.

[10] S. Abe and A. Gee, "Global convergence of the hopfield neural network with nonzero diagonal elements," *IEEE Trans. Circuits Syst. II*, vol. 42, no. 1, pp. 39–45, Jan 1995.

[11] S. V. Balakrishnan-Ayer, "Solving combinatorial optimization problems using neural networks with applications in speech recognition," Ph.D. dissertation, Trinity College, Cambridge, 1991.

[12] R. . Duda, P. E. Hart, and D. G. Stork, *Pattern Classification*, 2nd ed. Wiley-Interscience, October 2000.

[13] Z. Liu, Y. Xin, and G. B. Giannakis, "Linear constellation precoding for OFDM with maximum multipath diversity and coding gains," *IEEE Trans. Commun.*, vol. 51, no. 3, pp. 416–427, 2003.

[14] X. Ge and S.-T. Xia, "Structured non-binary LDPC codes with large girth," *Electronics Letters*, vol. 43, no. 22, 25 2007.

[15] T. Richardson and R. Urbanke, "The capacity of low-density parity-check codes under message-passing decoding," *IEEE Trans. Inform. Theory*, vol. 47, no. 2, pp. 599–618, Feb 2001.

[16] D. J. MacKay and M. C. Davey, "Evaluation of gallager codes for short block length and high rate applications," in *In Codes, Systems and Graphical Models*. Springer-Verlag, 1999, pp. 113–130.

[17] D. Declercq and M. Fossorier, "Decoding algorithms for nonbinary ldpc codes over gf," *IEEE Trans. Commun.*, vol. 55, no. 4, pp. 633–643, April 2007.

[18] K. P. Maré and B. T. Maharaj, "Performance analysis of modern space-time codes on a mimo-wimax platform," in *Proc. IEEE International Conference on Wireless and Mobile Computing, Networking and Communications WIMOB '08*, 2008, pp. 139–144.

[19] Z. Safar, W. Su, and K. J. R. Liu, "A fast sphere decoding algorithm for space-frequency block codes," *EURASIP J. Appl. Signal Process.*, pp. 148–148, January 2006.

[20] K. Maré, B. Salmon, and B. Maharaj, "Performance gain of space-time-frequency concatenated LDPC codes," in *Proc. IEEE Internation Conference on Communications ICC '09, Dresden, Germany*, June 2009, pp. CD–ROM.