



Conference Paper

Hopfield neural networks for vector precoding

Author(s):

Gardašević, Vesna; Müller, Ralf R.; Øien, Geir E.

Publication Date:

2010

Permanent Link:

<https://doi.org/10.3929/ethz-a-006000870> →

Rights / License:

[In Copyright - Non-Commercial Use Permitted](#) →

This page was generated automatically upon download from the [ETH Zurich Research Collection](#). For more information please consult the [Terms of use](#).

Hopfield Neural Networks for Vector Precoding

Vesna Gardašević, Ralf R. Müller and Geir E. Øien
 Department of Electronics and Telecommunications
 The Norwegian University of Science and Technology
 7491 Trondheim, Norway
 Email: {vesna.gardasevic, mueller, oien}@iet.ntnu.no

Abstract—We investigate the application of Hopfield neural networks (HNN) for vector precoding in wireless multiple-input multiple-output (MIMO) systems. We apply the HNN to vector precoding with N transmit and K receive antennas, and obtain simulation results for the average transmit energy optimization as a function of the system load $\alpha = K/N$. We compare these results with lattice search based precoding performances, and show that the proposed method for nonlinear vector precoding with complexity $O(K^3)$ achieves competitive performances in the range $0 < \alpha \leq 0.9$ in comparison to lattice search based precoders. The proposed method is of a polynomial complexity and therefore, it is an attractive suboptimal approach for vector precoding.

I. INTRODUCTION

In a broadcast MIMO system a transmitter, typically a base station, communicates with a number of receivers. We assume that the receivers cannot cooperate with each other, and that the channel state information is known at the transmitter side. In this scenario, our aim is to delegate most of the signal processing work to the transmitter side. The signal processing at the transmitter includes precoding techniques [1] for predistortion of the transmitted symbols. In this way the transmit energy is reduced and signal detection at the receivers is simplified.

The capacity region of Gaussian broadcast MIMO channels [2] can be achieved by dirty paper coding (DPC) [3]. Since DPC has high complexity demands for implementation, research in the area of precoding techniques has been focused on different linear and nonlinear sub-optimal methods. In linear precoding (e.g. [4]) the transmitted symbols are premultiplied by the pseudo-inverse $\mathbf{H}^\dagger(\mathbf{H}\mathbf{H}^\dagger)^{-1}$ of the channel matrix \mathbf{H} and the receiver applies simple symbol-by-symbol detection. This method is known as zero-forcing precoding (ZF) (e.g. [5]) and the main advantage of this method is its low complexity.

A drawback occurs when the channel matrix has small singular values, such that an inversion operation causes severe transmit power amplification. One method proposed to control power amplification due to ill-conditioned channel matrices is channel regularization [6]. However, this approach does not cancel all the interference at the receiver.

A nonlinear precoding approach employs nonlinear predistortion of the transmitted symbols before the linear operation. In a nonlinear predistortion step the alphabet of transmitted symbols is increased to a larger redundant set, such that the symbols to be sent are subject to optimization. A vector perturbation method [7], for example, modifies the idea of THP.

The vector-space search for the closest point in the lattice set that minimizes the energy is performed by an exhaustive search [7]. Often search is performed by a sphere encoder (SE), which is known to reduce the complexity, but still keeping it exponential [8].

Lattice-basis reduction algorithms have been proposed for further reduction of the SE complexity. Lower complexity can be achieved by searching for the approximately closest point in the lattice. There are implementations of vector precoding algorithms based on lattice-basis reduction [9], for example the LLL (Lenstra, Lenstra, and Lovasz) algorithm [10], that provide good performance.

Another approach to nonlinear predistortion is to apply a convex relaxation of the input symbol alphabet. A novel vector precoding method that applies a convex relaxed symbol alphabet instead of a discrete set is presented in [11]. In [12] it is shown that with a convex precoding approach, spectral efficiency can be higher than with lattice precoding at low to moderate signal-to-noise ratios.

The field of artificial neural networks (ANN) has been an active research area with periods of both dynamic and stagnation phases, from the early 1940s. Research has resulted in a great number of publications e.g. [13], and ANNs have been applied to the realization of, for example: content-addressable memory, pattern classifiers, pattern recognition, vector optimization, multiuser detection. ANNs have also been applied for solving optimization problems in different areas.

In this paper we will apply the Hopfield Neural Network (HNN) [13] which belongs to the class of recurrent ANNs as the algorithm for optimization in vector precoding. The structure of extensive parallelism makes the computational capabilities of the HNN very powerful and attractive. We will provide numerical simulation results for loads within $0 < \alpha \leq 1$, for $K = 8$, $K = 16$, $K = 27$ and $K = 64$, and compare the results with the performances of the SE lattice precoding, where the number of redundant representations of each information bit is $L = 2$ and the number of receive antennas are $K = 27$ and $K = 64$. We also compare the HNN precoder performances with the analytical solution for the SE vector precoding [12] and to the performance of convex relaxation (CR) [11] of the input symbol alphabet.

Our simulation results show that for loads $\alpha \leq 0.7$, the performance of the HNN vector precoding is very close to the SE performance. Up to $\alpha \approx 0.9$ there is a controlled increase in the transmit energy of the HNN algorithm compared with

the SE vector precoding performances. For loads $\alpha \approx 0.9$ and greater there is a substantial increase in the transmit energy of the HNN solution. Comparing the HNN performances with the analytical result for CR we show that up to $\alpha \approx 0.9$ the HNN precoding outperforms CR, which for loads $0.9 \leq \alpha < 1$ the energies provided by the HNN start to increase considerably.

It is known that the expected complexity of the SE algorithm is exponential [8]. Therefore, due to the fast computational capabilities and robustness of the HNN, our approach is an efficient suboptimal way for vector precoding within a wide load range at low complexity.

II. SYSTEM MODEL

A narrowband multi-user MIMO system is modeled by

$$\mathbf{r} = \mathbf{H}\mathbf{t} + \mathbf{n} \quad (1)$$

where $\mathbf{t} = [t_1, t_2, \dots, t_N]^T$ and $\mathbf{r} = [r_1, r_2, \dots, r_K]^T$ denote the transmit and receive signal vectors, respectively, the $K \times N$ channel matrix \mathbf{H} is assumed to have independent and identically distributed (i.i.d.) Gaussian random variables with zero mean and unit variance, and \mathbf{n} is the white Gaussian noise. We consider (1) as a MIMO system with a single transmitter with N antennas, and K receivers, each with a single antenna, that cannot cooperate with each other.

Now, assume that the number of transmit antennas is greater or equal to the number of receive antennas ($K \leq N$). The $K \times 1$ data symbol vector is denoted by \mathbf{s} , and for binary phase-shift keying (BPSK), the elements of the vector $\mathbf{s} = [s_1, s_2, \dots, s_K]^T$ belong to the set $\mathbf{S} = \{-1, +1\}$. Let the union of the sets $\mathcal{B}_{-1} = \{-1, +3\}$ and $\mathcal{B}_{+1} = \{+1, -3\}$ be the relaxed alphabet. In a nonlinear predistortion step the data vector \mathbf{s} is mapped onto a vector $\mathbf{x} = [x_1, x_2, \dots, x_K]^T$, with the vector elements chosen to minimize the transmit energy, where $\mathbf{x}_k \in \mathcal{B}_{s_k}$, for $k = 1, 2, \dots, K$. The following linear predistortion matrix \mathbf{T} is assumed to be

$$\mathbf{T} = \mathbf{H}^+ = \mathbf{H}^\dagger (\mathbf{H}\mathbf{H}^\dagger)^{-1} \quad (2)$$

The optimization problem can now be formulated as follows:

$$\mathbf{x}^* = \min_{\mathbf{x} \in \mathcal{B}_{s_1} \times \dots \times \mathcal{B}_{s_K}} \|\mathbf{T}\mathbf{x}\|^2 = \min_{\mathbf{x} \in \mathcal{B}_{s_1} \times \dots \times \mathcal{B}_{s_K}} \mathbf{x}^\dagger (\mathbf{H}\mathbf{H}^\dagger)^{-1} \mathbf{x} \quad (3)$$

This problem is difficult to solve since it is a nonconvex optimization problem in a high dimensional space, and we therefore investigate the application of the HNN for solving (3).

III. QUADRATIC OPTIMIZATION USING HOPFIELD NEURAL NETWORKS

Hopfield [13] proposed the application of ANNs for solving combinatorial optimization problems. A review of the HNN applications for solving mathematical programming problems is given in [14]. The optimization using the HNN is performed by constructing the energy function with the parameters that depend on a practical optimization problem. Hopfield [13] showed that the energy function constructed for an observed

problem provides convergence of the system to stable states if the matrix in the objective function is symmetric, with zero diagonal elements. The main drawback of the HNN is that its computational properties do not necessarily provide the best solution by minimizing the appropriate energy function, but the optimization can result in a local minimum. Various modifications of the HNN, for example the combination with stochastic algorithms [15] have been proposed for avoiding said local minima.

The HNN works as follows: the sum of an external threshold value θ_i and the weighted sum of input states are transformed by a nonlinear function called an activation function to become the output of the system. The HNN supports different activation functions, for example: hard limiter (threshold) transfer function, hyperbolic tangent (tanh), sigmoid and other functions. A weight matrix \mathbf{W} can be used to model various effects in an observed system, and depends on the particular problem that is solved by the HNN.

We apply the HNN model with an activation function $f(\cdot)$ described by

$$v_j^{(l+1)} = f \left(\sum_{i=1}^K w_{ji} v_i^{(l)} + \theta_j \right) \quad (4)$$

where $l = 0, 1, 2, \dots$, denotes the number of iterations run by the HNN, $i = 1, 2, \dots, K$, $j = 1, 2, \dots, K$, the network states are denoted by v_j , respectively, w_{ji} are the assigned weights between neurons j and i , and θ_j is the external input signal.

The HNN described by (4) minimizes the energy function

$$E = \frac{1}{2} \sum_{i=1}^K \sum_{j=1}^K w_{ij} v_i v_j + \sum_{i=1}^K v_i \theta_i \quad (5)$$

This HNN model can also be considered as an iterative algorithm that performs soft parallel interference cancellation [16], [17]. In our model the solution of the optimization problem defined in (3) corresponds to the minimization of the energy function in (5). The output vector \mathbf{v} of the HNN in (4) corresponds to the vector \mathbf{x} in (3), while the coefficients w_{ij} in (4) correspond to the entries of the channel matrix \mathbf{H} .

We assume that the start time of the iterations is $l = 1$, and that a soft decision $v_i^{(l)}$ is calculated in each step. This value is subtracted from the soft decision from the previous iteration and the hypothetical interference is cancelled in each iteration. The iterations can be performed until there is only a minor change between the soft decisions in successive iterations, i.e. until $\max_i |v_i^{(l)} - v_i^{(l-1)}| < \delta$ where δ is a sufficiently small value or the number of iterations exceeds the maximum number of iterations. In our simulations we have chosen the criterion that iterations are performed until the number of iterations exceeds the maximum number of iterations, denoted by I_{\max} .

It has been shown that the expected complexity [7] of the SE algorithm depends on the number of dimensions K and the signal-to-noise ratio (SNR). In [8] the expected complexity of the SE algorithm, as well as the asymptotic expression

for the complexity, were derived. The expected complexity is defined to represent the expected number of steps performed by the algorithm, and it is a function of the symbols \mathbf{x} , and the realization of the channel matrix \mathbf{H} . It has been shown that the expected complexity of the SE is exponential.

The optimization algorithm utilizing the HNN is shown in Table I.

TABLE I
HOPFIELD NEURAL NETWORK (HNN) ALGORITHM.

<p>Input: Set \mathbf{H}, $\mathbf{v} = \mathbf{s}$, $l = 1$, I_{\max}</p> <p>Set $\mathbf{W} = -\text{diag}[(\mathbf{H}\mathbf{H}^\dagger)^{-1}] + (\mathbf{H}\mathbf{H}^\dagger)^{-1}$</p> <p>Define: $f_{s_j}(y) = -s_j + 2 \tanh(2(y + s_j))$</p> <p>while $l \leq I_{\max}$</p> <p> for $1 \leq j \leq K$</p> <p> Calculate</p> <p> $v_j^{(l)} = f_j\left(-\sum_{i=1}^K w_{ij}v_i^{(l-1)}\right)$</p> <p> end</p> <p>end</p> <p>for $1 \leq j \leq K$</p> <p> $x_j = -s_j + 2 \cdot \text{sign}(v_j + s_j)$</p> <p>Output: \mathbf{x}</p>

For convex precoding, the quadratic programming solver from the MATLAB optimization toolbox [18] is applied. This algorithm has computational complexity $O(K^{3.2})$. Given a fixed number of iterations, the algorithm in Table I contains one loop that is executed K times and involves the summation of K terms. Its complexity is therefore $O(K^2)$.

When we compare computational complexities of the HNN and CR vector precoding methods, we can observe that the dominant complexity is in the linear operation (2). Numerical computation of $(\mathbf{H}\mathbf{H}^\dagger)^{-1}$ has complexity $O(K^3)$, and the additional complexity due to the application of the HNN is thus negligible in comparison with this pseudo-inverse operation.

IV. NUMERICAL RESULTS

In the HNN vector precoding algorithm the channel matrix \mathbf{H} has been modeled with i.i.d. Gaussian entries. For each realization of the channel matrix \mathbf{H} , the number of the iterations was set to be $I_{\max} = 40$.

We simulated the performance for: $K = 8$, $K = 16$, $K = 27$ and $K = 64$. The number of different channel realizations for each system size was 1000. We plot the resulting average transmit energy as a function of the ratio $\alpha = K/N$ of the number of receive antennas K to the number of transmit antennas N , where $0 < \alpha \leq 1$, as shown in Fig. 1. In the same figure we draw for comparison the following plots: the analytical solution for the SE lattice set [12] (the number of the redundant representations of each information bit is $L = 2$), simulation results for the SE lattice precoding ($L = 2$) with the number of receive antennas $K = 27$ and $K = 64$, and the analytical solution for CR [11].

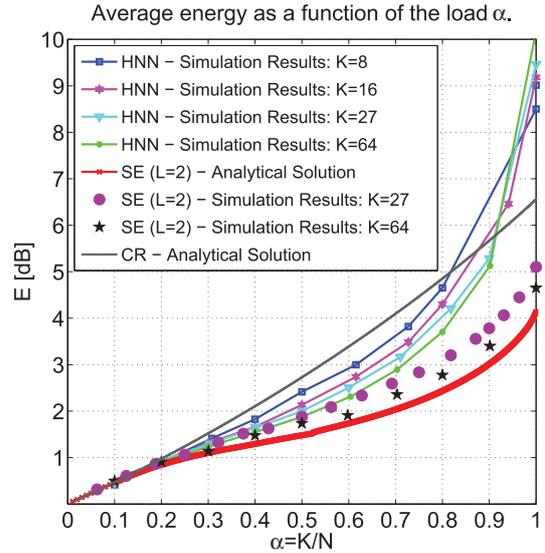


Fig. 1. The average transmit energy as a function of the load $\alpha = K/N$.

The simulation results show that precoding using the HNN provides performances very close to the SE-based precoding performances for the load $0 < \alpha \leq 0.8$. For example, for $K = 27$ and $\alpha = 0.5$ the performance of the SE is 4.88 dB and the HNN performance is 4.99 dB, while for $\alpha = 0.8$ the HNN shows performance penalty of less than 1 dB.

Almost similar results were obtained for $K = 64$; for $\alpha = 0.5$ the SE performance enhancement is 0.11 dB, while for $\alpha = 0.8$ the energy differs by 1 dB. For loads within $0.8 \leq \alpha \leq 0.9$ the average transmit energy by the HNN precoding gradually increases and for $\alpha = 0.9$, the SE outperforms the HNN by 1.5 dB and 1.69 dB for $K=27$ and $K = 64$, respectively. For loads greater than $\alpha \geq 0.9$, performance of the HNN degrades severely.

In comparison with the analytical results obtained for CR [12] we notice that up to $0 < \alpha \leq 0.9$ the HNN precoding outperforms CR for all simulated values of K , except for $K = 8$, in which case CR is outperformed up to $\alpha \leq 0.8$. The HNN performance enhancement is greatest in the range of $0.5 \leq \alpha \leq 0.8$ and increases when K gets larger. For example, for $\alpha \leq 0.7$, the HNN outperforms CR by 1.21 dB.

We have thus demonstrated that the HNN-based precoding outperforms the CR theoretical results within $0 < \alpha \leq 0.8$, achieves performances very tight to the SE in the range of $0 < \alpha \leq 0.7$, and has competitive performance in comparison to the SE for $0 < \alpha \leq 0.9$. It is known that lattice precoding is a problem that at loads close to 1 exhibits strong replica symmetry breaking (RSB) [12]. RSB problems are well-known to not being well-approximated by HNNs unlike to those that do not exhibit RSB.

V. COMPUTATIONAL COMPLEXITY

The advantages, limits and computational power of neural networks (for example: [19], [20]) and, in particular, the HNN

have been studied over years. There are various realizations of HNN, for example: continuous or discrete time, feedforward or recurrent model, with discrete or analog activation function, finite or infinite network size, asynchronous or synchronous network. The HNN computational complexity has been analyzed depending on the network model and its applications. Some of the results on the computational complexity have been generalized. We will consider a symmetric HNN applied to the energy minimization problems.

The computational complexity can be considered in terms of the memory and time resources required for a particular application. The highest computational cost in terms of the memory resources is due to the allocation of the memory space for storing the weight matrix \mathbf{W} . The number of the steps performed by the algorithm before the algorithm converges is convergence time and its trivial upper bound is 2^K . The HNN convergence time may be exponential in the worst case, for both sequential and parallel networks. However, it has been shown that under some mild conditions, the binary HNN converges in only $O(\log \log K)$ parallel steps in the average case. The property allows us to set the maximum number of iterations to a moderate value in practice

VI. CONCLUSIONS

We have presented a method for vector precoding using a HNN as algorithm for combinatorial optimization, and shown that this method can be applied for precoding within a wide load range. We investigated the performance of this scheme by extensive simulations, and compared the results with the simulation results of SE precoding, where the number of redundant representations of each information bit is $L = 2$, with corresponding analytical results for the SE, and with the analytical result for the convex precoding performance. The HNN vector precoding method obtains performances close to the discrete lattice precoding for loads up to $\alpha \leq 0.8$, and for loads $0.8 \leq \alpha \leq 0.9$ there is a gradual increase in the transmit energy within 1.7 dB depending on the number of receiving antennas. When we compare the HNN performances and the CR analytical result we observe that up to $\alpha \approx 0.8$ the HNN precoding outperforms CR for $K = 16, 27$ and 64 . For $\alpha \leq 0.7$ the HNN outperforms CR for all simulated values of K . Our simulations showed that this algorithm can be applied for system loads up to $\alpha \leq 0.9$. Therefore, the HNN is an attractive solution for vector precoding of polynomial complexity, with competitive performance within a wide load range in comparison with the SE of exponential complexity. Further modification of the algorithm will be addressed to control the energy penalty for the load up to $\alpha \leq 1$. Furthermore, due to its low complexity, the HNN precoding solution can serve as a starting solution for lattice-based searches with SEs. This allows a greatly improved starting radius for the SE and aid reduction of the SE's complexity.

Finally, we would like to outline that the HNN based lattice precoding, similar to the SE, can be combined with lattice basis reduction. While lattice basis reduction helps to reduce the complexity of the SE, we conjecture that it will not reduce

the complexity of the HNN, but improve its performance, particularly at high loads as it reduces the eigenvalue spread of the weight matrix \mathbf{W} . This will be investigated in future research.

REFERENCES

- [1] C. Windpassinger, R. Fischer, T. Vencel, and J. Huber, "Precoding in Multiantenna and Multiuser Communications," *IEEE Trans. Wireless Commun.*, vol. 3, no. 4, pp. 1305–1316, July 2004.
- [2] S. Vishwanath, N. Jindal, and A. Goldsmith, "Duality, Achievable rates, and Sum-Rate Capacity of Gaussian MIMO Broadcast Channels," *IEEE Trans. Inf. Theory*, vol. 49, no. 10, pp. 2658–2668, Oct. 2003.
- [3] M. Costa, "Writing on dirty paper," *IEEE Trans. Inf. Theory*, vol. 29, no. 3, pp. 439–441, May 1983.
- [4] M. Joham, W. Utschick, and J. Nosske, "Linear Transmit Processing in MIMO Communications Systems," *IEEE Trans. Signal Processing*, vol. 53, no. 8, pp. 2700–2712, Aug. 2005.
- [5] R. F. Fischer, *Precoding and Signal Shaping for Digital Transmission*. New York, NY, USA: John Wiley & Sons, Inc., 2002.
- [6] C. Peel, B. Hochwald, and A. Swindlehurst, "A Vector-Perturbation Technique for Near-Capacity Multiantenna Multiuser Communication-part I: Channel Inversion and Regularization," *IEEE Trans. Commun.*, vol. 53, no. 1, pp. 195–202, Jan. 2005.
- [7] B. Hassibi and H. Vikalo, "On the Sphere-Decoding Algorithm I. Expected Complexity," *IEEE Trans. Signal Processing*, vol. 53, no. 8, pp. 2806–2818, Aug. 2005.
- [8] J. Jalden and B. Ottersten, "On the Complexity of Sphere Decoding in Digital Communications," *IEEE Trans. Signal Processing*, vol. 53, no. 4, pp. 1474–1484, April 2005.
- [9] M. Taherzadeh, A. Mobasher, and A. K. Khandani, "Communication Over MIMO Broadcast Channels Using Lattice-Basis Reduction," *IEEE Trans. Inf. Theory*, vol. 53, no. 12, pp. 4567–4582, 2007.
- [10] A. K. Lenstra, H. W. Lenstra, Jr., and L. Lovász, "Factoring Polynomials with Rational Coefficients," *Math. Ann.*, vol. 261, no. 4, pp. 515–534, 1982.
- [11] R. R. Müller, D. Guo, and A. L. Moustakas, "Vector Precoding in High Dimensions: A Replica Analysis," *IEEE J. Sel. Areas Commun.*, vol. 26, no. 3, p. 530540, Apr. 2008.
- [12] B. M. Zaidel, R. R. Müller, R. de Miguel, and A. L. Moustakas, "On Spectral Efficiency of Vector Precoding for Gaussian MIMO Broadcast Channels," *Proc. IEEE 10th Int. Symp. Spread Spectrum Techniques and Applications (ISSSTA)*, pp. 232–236, Aug. 2008.
- [13] A. Cochocki and R. Unbehauen, *Neural Networks for Optimization and Signal Processing*. New York, NY, USA: John Wiley & Sons, Inc., 1993.
- [14] U. P. Wen, K. M. Lan, and H. S. Shih, "A Review of Hopfield Neural Networks for Solving Mathematical Programming Problems," *European Journal of Operational Research*, 2008.
- [15] W. H. Schuler, C. J. A. Bastos-Filho, and A. L. I. Oliveira, "A Hybrid Hopfield Network-Simulated Annealing Approach to Optimize Routing Processes in Telecommunications Networks," in *Proc. the Seventh International Conference on Intelligent Systems Design and Applications (ISDA)*. Washington, DC, USA: IEEE Computer Society, 2007, pp. 58–63.
- [16] R. R. Müller and J. B. Huber, "Iterated Soft-Decision Interference Cancellation for CDMA," in *Broadband Wireless Communications, Luise and Pupolin, Eds.* Springer, 1998, pp. 110–115.
- [17] T. Tanaka and M. Okada, "Approximate Belief Propagation, Density Evolution, and Statistical Neurodynamics for CDMA Multiuser Detection," *Information Theory, IEEE Transactions on*, vol. 51, no. 2, pp. 700–706, Feb. 2005.
- [18] Matlab Optimization Toolbox, Inc. v3.0.4 (r2006a). [Online]. Available: www.mathworks.com
- [19] J. Šíma, "Energy-based Computation with Symmetric Hopfield Nets," in *Limitations and Future Trends in Neural Computation, NATO Science Series: Computer Systems Sciences*, vol. 186. IOS Press, 2003, pp. 45–69.
- [20] J. Šíma, P. Orponen, and T. Antti-Poika, "On the Computational Complexity of Binary and Analog Symmetric Hopfield Nets," *Neural Comput.*, vol. 12, no. 12, pp. 2965–2989, 2000.