Quantum Signal Processing via Hamiltonian Neural Networks

Wieslaw Sienko, Wieslaw Citko
Gdynia Maritime Academy
81-225 Gdynia, Morska 83, Poland
email: wcitko@am.gdynia.pl

Abstract
We show that projectors proposed in the framework of Quantum Signal Processing can be implemented as Hamiltonian Neural Network based orthogonal filters. Moreover, such filters can be used as Universal Signal Processors. The structures of such processors rely on a family of Hurwitz-Radon matrices. To illustrate, we propose a procedure of nonlinear mapping synthesis. The problem of anticipation is reformulated as a problem of supervised learning.
Keywords: Quantum Signal Processing, Signal Processing via Neural Networks, Hamiltonian Systems

1 Introduction

In the literature on signal processing, a new mathematical tool borrowed from quantum theory, named Quantum Signal Processing (QSP), has recently been proposed [1]. QSP can be seen as a complementary notion for Quantum Computing, Quantum Computers and Quantum Information Theory. The objective of QSP is the algorithmic implementation of signal processing using a von Neumann interpretation of quantum mechanics principles and, specifically, using Quantum Measurement Theory based on von Neumann projection operators. It is well known that in the standard Quantum Mechanics each subspace **M** of ***H*** (a Hilbert space) specifies a property of a quantum system. There is a one-to-one correspondence between properties and orthogonal projection operators (E^2=E), i.e., properties are identified with projectors. However, the projectors proposed by QSP are not fulfilling the general constraints imposed by Quantum Mechanics.

We point out that, the projection operators can be alternatively implemented by Hamiltonian Neural Networks, i.e., lossless neural networks with weight matrices given by **J**-matrices (orthogonal and skew-symmetric). We show that Hamiltonian Neural Networks can create the architecture of universal projectors. Moreover, a family of projectors with different possible applications in signal processing can be conceived, based on a family of Hurwitz-Radon matrices. We point out that these projectors can be seen as Haar-Walsh spectrum analyzers, analog associative memories, classifiers and different types of mappings. Finally, we tackle the notion of weak anticipation, showing how to design anticipative systems by using supervised learning.

International Journal of Computing Anticipatory Systems, Volume 14, 2004
Edited by D. M. Dubois, CHAOS, Liège, Belgium, ISSN 1373-5411 ISBN 2-930396-00-8

2 On Family of Hurwitz-Radon Matrices

The algebraic problem of founding a set of orthogonal, skew-symmetric matrices $\mathbf{A}_k$ with following properties:

$$\mathbf{A}_j \mathbf{A}_k + \mathbf{A}_k\mathbf{A}_j = \mathbf{0},\ \mathbf{A}_j^2 = \mathbf{-1} \text{ for } j \neq k,\ k = 1, \dots, s \qquad (1)$$

has been independently solved by Hurwitz and Radon [2]. These matrices are known as Hurwitz-Radon matrices. The maximum number of matrices for a given N, where N is matrix dimension, has been determined by Hurwitz and Radon. Let $N = 2^a b$, where b is an odd number and $a = 4c + d$; $0 \leq d < 4$; $c \geq 0$.
Any family (1) of Hurwitz-Radon matrices (NxN) consists of s_{max} matrices, where

$$s_{max} = 8c + 2^d - 1 \qquad (2)$$

Number $\rho(N) = 8c + 2^d = s_{max} + 1$ is known as a Radon number. Hence $s_{max} = \rho(N) - 1$ and $\rho(N) \leq N$. Radon number fulfills $\rho(N) = N$ for N = 2, 4, 8. Moreover, for any N the maximum number s_{max} of family members exists for matrices with integer elements, i.e., { -1, 0, 1}. For our purposes, the following statements on family of Hurwitz-Radon matrices could be interesting:

1. The maximum number of continuous orthogonal tangent vector fields on sphere $S^{N-1} \subset R^N$ is $\rho(N) - 1$.
2. Let $\mathbf{A}_1, \dots, \mathbf{A}_s$ be a set of orthogonal Hurwitz-Radon matrices (1) and $\mathbf{A}_o = \mathbf{1}$. Let $\alpha_0, \dots, \alpha_s$ be real numbers with:

   $$\sum_{i=0}^{s} \alpha_i = 1 \qquad (3)$$

   Hence, matrix:

   $$\mathbf{A}(\boldsymbol{\alpha}) = \sum_{i=0}^{s} \alpha_i \mathbf{A}_i$$

   is orthogonal. Since $\alpha = (\alpha_0, \alpha_1, \dots, \alpha_s) \in S^s \subset R^{s+1}$, then eq. (3) can be seen as a map of sphere S^s into the orthogonal group O(N). Moreover, the eq. (1) can be treated as a problem of finding the maximum number of orthogonal tangent vector fields on S^{N-1}. It is worth noting that Hurwitz-Radon matrices can be used for creating the weight matrices of Hamiltonian Neural Networks.

3 Hamiltonian Neural Networks

Inspired by results known from classical and quantum mechanics we aim to show that very large scale artificial neural networks should be implemented as passive or, particularly, as lossless structures. These technical notions mean that from a mathematical point of view they should be Hamiltonian systems [3], [4]. It is worth noting that:

1. passivity implies BIBO stability of the structure.
2. passivity of the structure can be attained by a compatible connection of elementary passive building blocks, i.e., neurons.

A general form of an autonomous Hamiltonian system is given by the following state-space equation:

$$\dot{\mathbf{x}} = \mathbf{J}\mathbf{H}'(\mathbf{x}) = \mathbf{v}(\mathbf{x}) \tag{4}$$

where: $\mathbf{v}(\mathbf{x})$ – a nonlinear vector field
and $-\mathbf{J} = \mathbf{J}^T = \mathbf{J}^{-1}$ ($\mathbf{J}$ – skew-symmetric matrix) (5)

There is such a basis in R^{2n} where matrix $\mathbf{J}$ has a form:

$$\mathbf{J} = \begin{bmatrix} \mathbf{0} & \mathbf{1} \\ \mathbf{-1} & \mathbf{0} \end{bmatrix}; \ \mathbf{0}, \mathbf{1}, \mathbf{-1} \text{ are (n x n) diagonal matrices} \tag{6}$$

Function $\mathbf{H}(\mathbf{x})$ is a hamiltonian (energy) of the system. Since in Hamiltonian systems there is no dissipation of energy, their trajectories in the state space can be very complicated for $t \rightarrow \pm\infty$. Therefore, the basic method of the "movement" description is to find periodic solutions, using, for example, the Maupertuis principle. Equation (4) has constant solutions, i.e., every point $\mathbf{x}_0 \in R^{2n}$ such that $\mathbf{H}'(\mathbf{x}_0) = \mathbf{0}$ is the equilibrium and $\mathbf{x}(t) \equiv \mathbf{x}_0$ is the solution. It can be shown that the structure of the Hamiltonian neural network (HNN) can be obtained as a compatible connection of N lossless neuron pairs. An example of such a pair is shown in Fig.1.

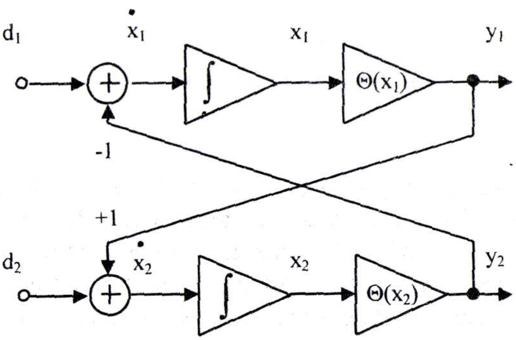

Figure 1. Two compatibly connected neurons - a two neuron lossless network.

It can be seen that the state-space description of the network from Fig.1 is as follows:

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} \Theta(x_1) \\ \Theta(x_2) \end{bmatrix} + \begin{bmatrix} d_1 \\ d_2 \end{bmatrix} \quad \text{where: } \mu_1 \le \frac{\Theta(x)}{x} \le \mu_2,\ \mu_1, \mu_2 \in [0, \infty] \tag{7}$$

The two neuron lossless network from Fig.1 can be treated as an elementary building block to create very large scale neural networks. Note, however, that in addition to the continuous time neurons (Fig.1), discrete time and PLL (Phase-Lock Loop) structures can be designed as well. Generally, a lossless neural network composed of N neuron pairs is described by the following state-space equation:

$$\dot{\mathbf{x}} = \mathbf{W}\boldsymbol{\Theta}(\mathbf{x}) + \mathbf{d} \tag{8}$$

where: **W**-matrix of information flow connections (weight matrix) and $\mathbf{W} = -\mathbf{W}^T$ (skew-symmetry), **d** represents an input signal (data)

Thus, a neural network composed of N elementary neuron pairs (from Fig. 1) with orthogonal weight matrix **W,** i.e.,

$$\mathbf{W}\,\mathbf{W}^T = \mathbf{1} \tag{9}$$

is a Hamiltonian system, with activation function $\boldsymbol{\Theta}(\mathbf{x}) = \mathbf{H}'(\mathbf{x})$, since

$$\mathbf{W}^2 = -\mathbf{1} \text{ i.e. } \mathbf{W}^{-1} = \mathbf{W}^T = -\mathbf{W} \tag{10}$$

the Hamiltonian neural network can be seen as an involutional operator. The weight matrix of HNN can be formulated as follows:

$$\mathbf{W}_{2^n} = \begin{bmatrix} \mathbf{W}_{2^{n-1}} & \mathbf{W}_C \\ -\mathbf{W}_C^T & -\mathbf{W}_{2^{n-1}} \end{bmatrix};\ n = 1, 2, \ldots \tag{11}$$

where:

$$\mathbf{W}_1 = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}$$

$$\text{and } \mathbf{W}_C \cdot \mathbf{W}_C^T = \mathbf{1} \ \left(\dim \mathbf{W}_C = \dim \mathbf{W}_{2^{n-1}}\right) \tag{12}$$

$$\mathbf{W}_{2^{n-1}} \cdot \mathbf{W}_C - \mathbf{W}_C \cdot \mathbf{W}_{2^{n-1}} = \mathbf{0}$$

Possible solutions of (12) are:

$$\mathbf{W}_C = \mathbf{1} \quad \text{and} \quad \mathbf{W}_C = \mathbf{W}_{2^{n-1}} + \mathbf{1}$$

It can be seen that rows and columns of matrix $\mathbf{W}_{2^n}$ constitute a Haar-basis. Moreover, the weight matrix **W** can be obtained by using a family of matrices (3).

4 Hamiltonian Neural Network as a Spectrum Analyzer

Some basic properties of the HNN can be derived from (4). First, as mentioned above, the structure of such a network creates a nonlinear vector field: $\mathbf{W\Theta(x)}=\nu(\mathbf{x})$ with a single equilibrium point for $\mathbf{x} = \mathbf{0}$. Hence, HNN determines a type of orthogonal transformation, namely:

$$\mathbf{W}\,\Theta(\mathbf{x}) + \mathbf{d} = \mathbf{0} \tag{13}$$

where: **d** – input vector (input data or signal)

It is worth noting that (13) gives the steady state solution of the network under constant excitation. Hence, the output of the network, i.e.,

$$\Theta(\mathbf{x}) = \mathbf{W}\,\mathbf{d} \tag{14}$$

where the rows and columns of **W** constitute the orthogonal Haar basis can be seen as a Haar spectrum of input vectors. Since, however, $\Theta(\mathbf{x})$ is the output of a nonlinear, dynamical Hamiltonian system, (14) is true only for such a bounded input where $|\Theta(\mathbf{x})| \le 1$ (for saturated (± 1) activation functions).
Since $\mathbf{W}^2 = \mathbf{-1}$, it is clear that this Haar analysis sets up the following relationships:

a) $(\Theta\,,\mathbf{d}) = 0$, where $(.\,,.)$ denotes scalar product in l_2
b) The components of vector $\Theta(\mathbf{x})$ are Haar coefficients. Thus, the HNN performs a decomposition of the input vector into the sum of orthogonal patterns (columns or rows of weight matrix **W**). If the input vector consists of discrete samples of a time function, then these patterns can be treated as Haar-like wavelets.
c) If, for a given input data set, there are large Haar coefficients, then the spectrum analysis fixes a number of principal components of the input data.

The Haar analysis using HNN can be schematically shown as in Fig 2.

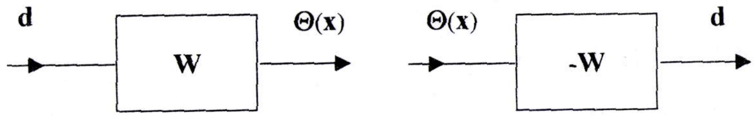

Figure 2. Haar analysis using HNN

Note: The Haar analysis illustrated in Fig. 2 assumes a physical neural network (a signal processor) with an orthogonal, skew-symmetric weight matrix **W** solving the following, ill-conditioned, differential equations:

$$\dot{\mathbf{x}} = \mathbf{W}\Theta(\mathbf{x}) + \mathbf{d}$$

Such a signal processor cannot be practically realized (in silicon) due to the pure imaginary eigenvalues of matrix **W**. On the contrary, the Haar transformation given by algebraic equation (14) is ready to use as an algorithm. In the following section we show how to solve the problem of physical realizability.

5 HNN as an Orthogonal Filter

Orthogonal filtering is one of the basic operations in modern signal processing. For example, in digital communication, transmitted messages are encoded in the form of orthogonal symbols. Thus, a transmitted signal consists of such symbols corrupted by additive noise. Hence, a primary function of any receiver is to perform such an orthogonal filtering. The basic structure of an orthogonal filter, using the structure of the HNN, is shown in Fig.3.

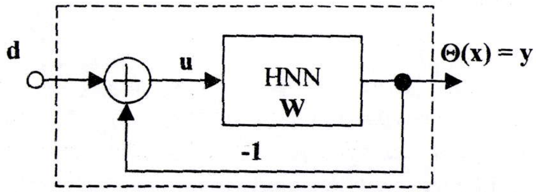

Figure 3. Structure of an orthogonal filter.

It can be seen that such a filter performs the following decomposition:

$$\mathbf{d} = \mathbf{u} + \mathbf{y}\,; \quad \mathbf{y} = \Theta(\mathbf{x}) \tag{15}$$

where: **u** and **y** are orthogonal, i.e., $\mathbf{y} = \mathbf{W}\mathbf{u}$ and $(\mathbf{u}, \mathbf{y}) = 0$.

At the same time (15) sets up two types of orthogonal transformations:

$$\mathbf{y} = 0.5\,(\mathbf{1}+\mathbf{W})\,\mathbf{d}\,; \; (\mathbf{d}, \mathbf{y}) \neq 0 \tag{16}$$

and

$$\mathbf{u} = 0.5\,(\mathbf{1}\text{-}\mathbf{W})\,\mathbf{d} \tag{17}$$

According to (16) the input signal **d** is decomposed into a Haar or Walsh basis (Walsh functions take only the values 1 and –1), and the output signal **y** constitutes the Haar or Walsh spectrum, respectively. Assuming, that the above symbols are encoded by columns or rows of matrix ($\mathbf{1} + \mathbf{W}$), the largest Haar coefficient of the spectrum at the output of the orthogonal filter can be used as a measure of presence of information hidden in noise. Indeed, one obtains

$$\mathbf{d} = \mathbf{w}_i + \mathbf{n} = \mathbf{u} + \mathbf{y} \tag{18}$$

where: $\mathbf{w}_i$ - a symbol (a column or row of matrix $(\mathbf{1} + \mathbf{W})$), $i \in [1, \ldots, N]$,
$\mathbf{n}$ - additive noise, $N = \dim \mathbf{W}$

Hence, if there is such a minimal S/N ratio, that

$$\Theta_i = (\mathbf{w}_i + \mathbf{n}, \mathbf{w}_i) > \Theta_k = (\mathbf{w}_i + \mathbf{n}, \mathbf{w}_k), \forall k \neq i \tag{19}$$

where: Θ_i – the largest Haar coefficient of the output spectrum, then the orthogonal filtering can be performed.

6 Realizability of HNN

As mentioned above the HNN described by (8) cannot be realized as a technical object. On the contrary, the orthogonal filter shown in Fig. 3 could be implemented in silicon, even if the weight matrix **W** is not exactly skew-symmetric. It is clear that this implementation is guaranteed by the stabilizing action of negative feedback loops. Moreover, by cascading two such orthogonal filters one obtains a Haar spectrum analyzer equivalent to that shown in Fig. 2. In Fig. 4 such a cascade is schematically presented.

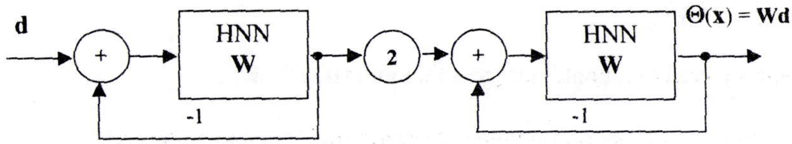

Figure 4. Haar spectrum analyzer by using two orthogonal filters.

Thus, without going into detail on the technological issues, one can state that very large scale artificial neural networks can be realized as physical objects by the structure of orthogonal filters.

7 HNN as a Universal Signal Processor

It is not the main goal of this paper to present practical applications of signal processing using HNN. Nevertheless, it seems to be clear that, by having such networks implemented in the form of orthogonal filters, one could realize, in real-time, most of the signal processing known from advanced wavelet analysis, for example:

- signal denoising
- coherent structure extraction-orthogonal filtering
- pattern recognition and classification

- multiscale and multiresolution signal analysis
- image processing
- data compression

It is worth noting that the problem of orthogonal Hurwitz-Radon matrices, mentioned above, and the solution given by Hurwitz-Radon theorem can be used to formulate two essential issues in signal processing performed by HNN, namely:

- finding the best-adapted bases for a given class of signals
- decomposition of a given signal (pattern/image) into orthogonal components.

Indeed, let $\mathbf{W}_1, \ldots, \mathbf{W}_s$ be e set of orthogonal skew-symmetric Hurwitz-Radon matrices i.e.

$\mathbf{W}_j \mathbf{W}_k + \mathbf{W}_k \mathbf{W}_j = \mathbf{0}$ for $j \neq k$; $j, k = 1, \ldots, s$

Let $\alpha_1, \ldots, \alpha_s$ be real numbers with $\Sigma \alpha_j^2 = 1$. Then:

$$\mathbf{W}(\boldsymbol{\alpha}) = \sum_{j=0}^{s} \alpha_j \mathbf{W}_j \quad ; \mathbf{W}_0 = \mathbf{1} \tag{20}$$

is orthogonal, where $s_{max} = \rho(n) - 1$; $\rho(n)$ - Radon number of n. Hence the following adaptation rule:

Find such a vector of parameters $\boldsymbol{\alpha}$ that the weight matrix $\mathbf{W}(\boldsymbol{\alpha})$ of HNN sets up the best-adapted basis.

8 Design of HNN Based Associative Memories and Mappings

It is worth noting that, using the Hamiltonian Neural Network (HNN) based spectral analyzers, the typical functions of neural networks can be implemented as well. This means, for example, that one can implement content-addressable or associative memories and, hence, different types of selectors, classifiers and nonlinear mappings. The purpose of this note is to show how the mapping can be implemented in the form of an associative memory.

A design of such a mapping is based on the following procedure:

Let a mapping $\mathbf{y} = \mathbf{F}(\mathbf{u})$ be given by the following trainings points:

$$\mathbf{u}_k \rightarrow \mathbf{y}_k, \quad \text{for } k = 1, \ldots, n \tag{21}$$

where $\mathbf{u}_k$ and $\mathbf{y}_k$ belong to input and output vector spaces (generally with different dimensions), respectively.

In the first step of the design procedure, let us create the following memory matrix $\mathbf{M}$:

$$\mathbf{M} = [\ \mathbf{m}_1,\ \mathbf{m}_2\ , \dots ,\ \mathbf{m}_n\ ] \tag{22}$$

where:

$$\mathbf{m}_k = (\mathbf{W}-\mathbf{1})\begin{bmatrix}\mathbf{u}_k\\ \mathbf{y}_k\end{bmatrix} = (\mathbf{W}-\mathbf{1})\mathbf{d}_k \quad \text{for } k = 1, \dots , n; \quad \mathbf{d}_k = \begin{bmatrix}\mathbf{u}_k\\ \mathbf{y}_k\end{bmatrix} \text{ - given input vector} \tag{23}$$

It is easy to see that memory vectors $\mathbf{m}_k$ consist of a Haar spectrum of training points $\mathbf{u}_k$ and $\mathbf{y}_k$ (Fig.5.)

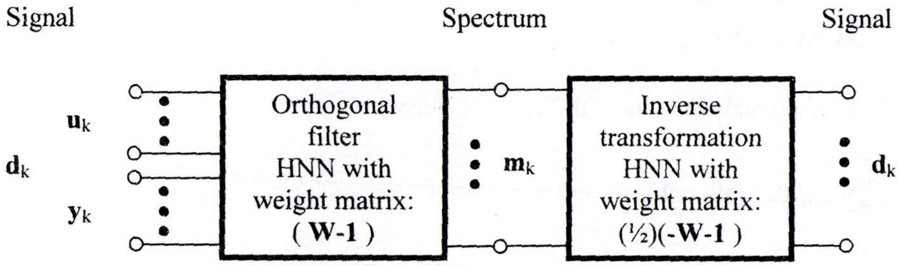

Fig.5. Spectrum analysis of training points

The proposed structure of an associative memory (fulfilling the relation (21)) is shown in Fig.6

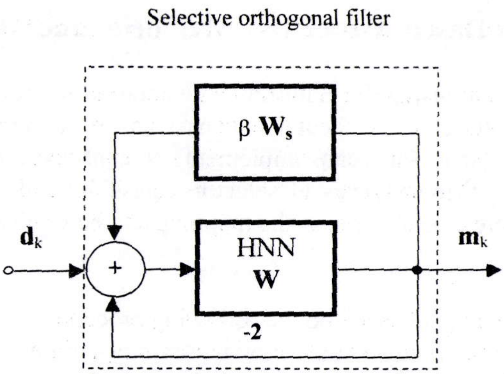

Fig. 6. The structure of an associative memory –selective orthogonal filter.

The block denoted as $\mathbf{W}_s$ in Fig.6. represents a symmetrical part of the weight matrix, where

$$\mathbf{W}_s = \mathbf{M}\ (\ \mathbf{M}^T\ \mathbf{M})^{-1}\ \mathbf{M}^T \quad \text{(pseudoinverse of } \mathbf{M}) \tag{24}$$

and $\beta > 0$ (25)

This structure can be seen as a neural network with a weight matrix $\mathbf{W}_M$ containing two components: antisymmetric (orthogonal) **W** and symmetric:

$$-\mathbf{2} + \beta\, \mathbf{W}_s$$

i.e. $\mathbf{W}_M = \mathbf{W} - \mathbf{2} + \beta\, \mathbf{W}_s$ (26)

Hence, one obtains the following statements:

1. For $\beta \leq 1$ the associative memory from Fig.6. constitutes a selective orthogonal filter, i.e. ,

   $\mathbf{m}_k = (\mathbf{W-1})\, \mathbf{d}_k$ for $k = 1, 2, \dots , n$

2. The state space of this associative memory is a set of attractors with attraction centers $\mathbf{m}_k$, $k = 1, 2, \dots , n$. Due to such an attractor structure, this associative memory (being passive) can reconstruct the input points belonging to the neighborhood of trainings points.

Finally, the structure of mapping **F(•)** implementation by successive approximation is given in Fig.7.

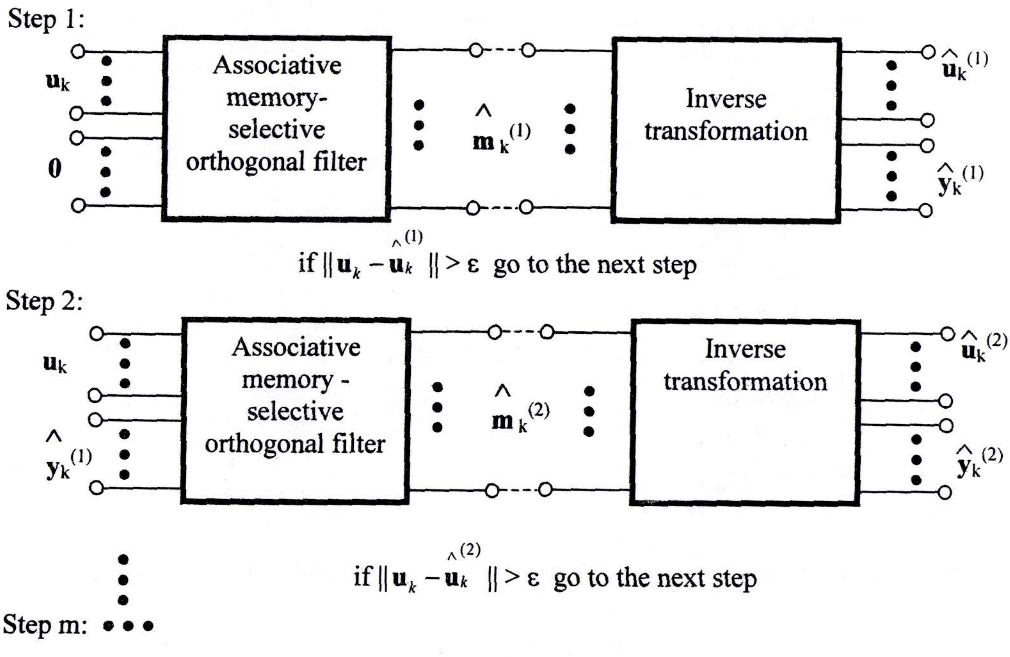

Fig.7. The structure of a nonlinear mapping **F(•)** implementation.

Example

Let us design a nonlinear mapping given by the following 3 training points:

$$\mathbf{u}_1 = \begin{bmatrix} -0.031922 \\ -0.040369 \\ -0.050288 \\ 0.011161 \\ 0.070934 \\ -0.030971 \\ 0.088485 \end{bmatrix} \rightarrow y_1 = 0.045005 \text{ i.e. } \mathbf{d}_1 = \begin{bmatrix} -0.031922 \\ -0.040369 \\ -0.050288 \\ 0.011161 \\ 0.070934 \\ -0.030971 \\ 0.088485 \\ 0.045005 \end{bmatrix}$$

$$\mathbf{u}_2 = \begin{bmatrix} -0.039411 \\ -0.033591 \\ -0.086611 \\ -0.078058 \\ 0.004112 \\ 0.017683 \\ -0.022274 \end{bmatrix} \rightarrow y_2 = 0.081688 \text{ i.e. } \mathbf{d}_2 = \begin{bmatrix} -0.031922 \\ -0.040369 \\ -0.050288 \\ 0.011161 \\ 0.070934 \\ -0.030971 \\ 0.088485 \\ 0.081688 \end{bmatrix}$$

$$\mathbf{u}_3 = \begin{bmatrix} -0.065599 \\ -0.017373 \\ -0.079100 \\ -0.055284 \\ 0.009708 \\ -0.067576 \\ -0.074152 \end{bmatrix} \rightarrow y_3 = -0.002495 \text{ i.e. } \mathbf{d}_3 = \begin{bmatrix} -0.031922 \\ -0.040369 \\ -0.050288 \\ 0.011161 \\ 0.070934 \\ -0.030971 \\ 0.088485 \\ -0.002495 \end{bmatrix}$$

The Haar analysis using orthogonal filter (23) with weight matrix **W** given by:

$$\mathbf{W} = \frac{1}{\sqrt{7}} \begin{bmatrix} 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ -1 & 0 & -1 & 1 & -1 & 1 & -1 & 1 \\ -1 & 1 & 0 & -1 & -1 & 1 & 1 & -1 \\ -1 & -1 & 1 & 0 & -1 & -1 & 1 & 1 \\ -1 & 1 & 1 & 1 & 0 & -1 & -1 & -1 \\ -1 & -1 & -1 & 1 & 1 & 0 & 1 & -1 \\ -1 & 1 & -1 & -1 & 1 & -1 & 0 & 1 \\ -1 & -1 & 1 & -1 & 1 & 1 & -1 & 0 \end{bmatrix}$$

gives the following spectrum points $\mathbf{m}_k$

$$\mathbf{m}_1 = \begin{bmatrix} 0.001777 \\ -0.030004 \\ -0.039876 \\ 0.027392 \\ 0.007130 \\ 0.031364 \\ 0.077770 \\ 0.015389 \end{bmatrix} \quad \mathbf{m}_2 = \begin{bmatrix} -0.041804 \\ 0.014449 \\ -0.044535 \\ -0.034496 \\ -0.042489 \\ 0.005388 \\ 0.033910 \\ 0.061331 \end{bmatrix} \quad \mathbf{m}_3 = \begin{bmatrix} -0.086846 \\ 0.007132 \\ -0.048127 \\ -0.030456 \\ 0.015816 \\ -0.025323 \\ 0.011519 \\ 0.012995 \end{bmatrix}$$

Symmetric matrix $\mathbf{W}_s$ for these memory points $\mathbf{m}_k$ is as follows:

$$\mathbf{W}_s = \begin{bmatrix} 0.6462 & -0.0571 & 0.3078 & 0.2180 & -0.1748 & 0.2282 & -0.0050 & -0.0199 \\ -0.0571 & 0.1505 & 0.0608 & -0.1799 & -0.1650 & -0.0822 & -0.2055 & 0.1107 \\ 0.3078 & 0.0608 & 0.3157 & 0.0382 & -0.0422 & -0.0209 & -0.3086 & -0.1367 \\ 0.2180 & 0.1799 & 0.0382 & 0.2598 & 0.1815 & 0.1204 & 0.1763 & -0.1799 \\ -0.1748 & -0.1650 & -0.0422 & 0.1815 & 0.4665 & -0.1042 & 0.0211 & -0.3809 \\ 0.2282 & -0.0822 & -0.0209 & 0.1204 & -0.1042 & 0.1801 & 0.2303 & 0.1000 \\ -0.0050 & -0.2055 & -0.3086 & 0.1763 & 0.0211 & 0.2303 & 0.5763 & 0.1488 \\ -0.0199 & 0.1107 & -0.1367 & -0.1799 & -0.3809 & 0.1000 & 0.1488 & 0.4049 \end{bmatrix}$$

For example, the approximation process for point $\mathbf{d}_1$, i.e., $\mathbf{u}_1 \rightarrow \mathbf{y}_1$, using the structure from Fig.7. with $\beta = 1$, is as follows:

Step 1:

$$\text{for input } \mathbf{u}_1 = \begin{bmatrix} -0.031922 \\ -0.040369 \\ -0.050288 \\ 0.011161 \\ 0.070934 \\ -0.030971 \\ 0.088485 \end{bmatrix} \text{ and } y_1 = 0.0000000 \rightarrow \text{output } \hat{y}_1^{(1)} = 0.007630$$

Step 2:

$$\text{for input } \mathbf{u}_1 = \begin{bmatrix} -0.031922 \\ -0.040369 \\ -0.050288 \\ 0.011161 \\ 0.070934 \\ -0.030971 \\ 0.088485 \end{bmatrix} \text{ and } \hat{y}_1^{(1)} = 0.007630 \rightarrow \text{output } \hat{y}_1^{(2)} = 0.013954$$

Hence after 30 steps one has the following mapping:

$$\text{for input } \mathbf{u}_1 = \begin{bmatrix} -0.031922 \\ -0.040369 \\ -0.050288 \\ 0.011161 \\ 0.070934 \\ -0.030971 \\ 0.088485 \end{bmatrix} \text{ and } \hat{y}_1^{(29)} = 0.044388 \rightarrow \text{output } \hat{y}_1^{(30)} = 0.044421$$

$y_1 = 0.045005$ is the nominal value

Notes

1. Proposed above design is linear and well-posed (due to the passivity of structures). Hence, it can be easily extended to high dimensionality. It is also worth noting that, at the same time, the outputs $\mathbf{y} = \Theta(\mathbf{x})$ and $\mathbf{x} = \Theta^{-1}(\mathbf{y})$ can be exploited.
2. The structures shown in Fig. 5 and Fig. 6 can be seen as an implementation of different type projectors.

3. From a topological point of view, the procedure of associative memory realization can be interpreted as a „drilling of holes in the sphere".
4. The method of association proposed in the above example seems to be very humanlike, i.e., the results and quality of association depends on available time (think about chess players) or the available number of procedure steps.

9 Anticipation by Learning

The voluminous literature deals with formal definitions and different notions of anticipation. Without delving too much into philosophical matters, one can state that the notion of anticipation should be related to the possibility of time reversal. Rosen's notions of anticipation and anticipatory systems deal with time reversal as a predictive model of itself and/or its environment, which makes it possible for the system (also biological) to compute its present state as a function of the model's prediction [8], [9]. It is worth noting that such a structure of an anticipatory, technical system can be found in model reference adaptive control, especially as a fuzzy model reference-learning controller (FMRLC). The above described implementation of a nonlinear mapping $\mathbf{y}$(or $\mathbf{x}$) = $\mathbf{F}(\mathbf{u})$ can be treated as a model of supervised (by examples) learning. Indeed, learning techniques are similar to implementation of the mapping $\mathbf{F}(\bullet)$ relying on the fitting of experimental pairs { $\mathbf{u}_k$, $\mathbf{y}_k$ }. The key point is that the fitting should be predictive and uncover the underlying physical law, which is then used in a predictive way. Hence, one can easily see the relation between supervised learning and Rosen's anticipatory system. Thus, the structures of selective orthogonal filters can be seen as implementations of anticipatory systems. Our comparative study on the quality and features of these systems versus systems known as Support Vector Machines (SVM) and Regularized Least Squares Classifiers (RLSC) is being prepared [5]. The problem of real (not by model) time reversal can be considered only in framework of quantum mechanics. Within an interpretation of quantum mechanics proposed by R. Omnes [6[, [7] one can find an appropriate theoretical tool to approach the fundamental question: Is it possible to change the direction of time? The bedrock of the Omnes theory is the concept of a family of consistent histories: Any physical process, and particularly an experiment, can be described by a family of histories, where a history is a time-ordered sequence of physical properties, mathematically coded as a sequence of projectors (projection operators in Hilbert space) $E_1(t)$, $E_2(t)$, … ,$E_3(t)$ reference times being ordered according to $t_l < t_2 < \ldots < t_n$. Omnes answer to the above mentioned fundamental question is as follows:

1. The time direction is determined by an irreversible, dynamical process of decoherence i.e. $E(t_k) \Rightarrow E(t_s)$, for $t_k < t_s$ (a datum cannot be changed or contradicted by a later external action or evolution).
2. The close relation between dissipation and decoherence implies that the only possibility of avoiding decoherence is to work with nondissipative systems.

Omnes considers several examples of quantum object complementarity (e.g. EPR) resulting in peculiarities which could be interpreted as anticipation i.e. $E(t_k) \Leftarrow E(t_s)$, for

$t_k < t_s$ but quantum logic makes us to choose the direction of decoherence.

Conclusion 1

Due to decoherence, the real time reversal is impossible (with probability equal 1).

Conclusion 2
The von Neumann language of projectors and quasi-projectors is quite universal. Hence, it could be used in the framework of families of histories for describing the very complex dynamical systems alternatively to Zadeh's fuzzy logic systems and soft computing [10].
As a byproduct of these considerations, we claim that hamiltonian neural network based projectors can be physically implemented as near-lossless structures. Hence, they could be considered as models of physically implementable quantum computers

10 Concluding Remarks

In this paper, we have presented how to find the most suitable architecture for very largescale artificial neural networks. The structure of such neural networks is composed of pairs of lossless neurons. Hence, they have the form of Hamiltonian Systems with orthogonal weight matrices, **W**, particularly belonging to a family of Hurwitz-Radon matrices. The unique feature of HNN seems to be the fact that HNN can exist as algorithms or Hamiltonian physical devices performing the Haar-Walsh analysis in real-time. The family of Hurwitz-Radon matrices generates a set of HNN based signal processors or algorithms belonging to the notion of Quantum Signal Processing.

References

[1] Yonina C. Eldar, Alan V. Oppenheim, Quantum Signal Processing, IEEE Signal Processing Magazine, Vol. 19. No 6, 2002 pp. 12-32.

[2] Beno Eckmann, Topology, Algebra, Analysis-Relations and Missing Links, Notices of the AMS, Vol. 46, No 5, 1999, pp.520-527.

[3] Wieslaw Sienko, Wieslaw Citko, On Very Large Scale Hamiltonian Neural Nets, in Neural Networks, in Neural Networks and Soft Computing, Eds L. Rutkowski, J. Kacprzyk, Springer Physica-Verlag, Heidelberg, NY 2003.

[4] Wieslaw Sienko, Wieslaw Citko, Bogdan Wilamowski, Hamiltonian Neural Nets as a Universal Signal Processor, The 28th Annual Conference of the IEEE Industrial Electronics Society, IECON'2002, Seville, 2002, SF-007388.

[5] Tomaso Poggio, Steve Smale, The Mathematics of Learning: Dealing with Data, Notices of the AMS, Vol.50, No 5, 2003, pp.537-544.

[6] Roland Omnès, Understanding Quantum Mechanics, Princeton University Press, 1999.

[7] Wiliam Faris, Review of Roland Omnès, The Interpretation of Quantum Mechanics, Notices of AMS, Vol. 43, No 11, 1996, pp. 1328-1339.

[8] Daniel M. Dubois, Computing Anticipatory Systems with Incursion and Hyperincursion, Computing Anticipatory Systems: CASYS'97 First International Conference. Edited by Daniel M. Dubois, Published by The American Institute of Physics, AIP Conference Proceedings 437, pp. 3-30.
[9] Daniel M. Dubois, Hyperincursive and Anticipatory Systems-Foundation of Anticipation in Electromagnetism, Computing Anticipatory Systems: CASYS'99 Third International Conference. Edited by Daniel M. Dubois, Published by The American Institute of Physics, AIP Conference Proceedings 517, pp. 3-30.
[10] Lotfi A. Zadeh, From Computing with Numbers to Computing with Words – From Manipulation of Measurments to Manipulation of Perceptions: Computing Anticipatory Systems: CASYS'00 Fourth International Conference. Edited by Daniel M. Dubois, Published by The American Institute of Physics, AIP Conference Proceedings 573, pp. 36-58.