Models of Oscillatory Nonlinear Mappings

Wieslaw Citko, Wieslaw Sienko
Department of Electrical Engineering
Gdynia Maritime University
81-225 Gdynia, Morska 83, Poland
emails: sienko@am.gdynia.pl, wcitko@am.gdynia.pl

***Abstract*— In this paper we present how to design very large scale oscillatory nonlinear mappings by using orthogonal filters which, due to spectrum based information processing, can be seen as implementations of holographic-like structures.**

***Keywords*— nonlinear mappings, oscilatory neural networks**

I. Introduction

A number of experiments show that some cognitive functions of biological brains could be seen as holographic processes (see for example [1]). Hence, we believe that biologically motivated structures of artificial neural networks cannot rely on dissipative dynamical networks, with their different type of attractors, (e.g. chaotic attractors) or on multilayer feedforward neural networks trained by backpropagation algorithms. It seems that, at least for very large scale associative memories needed to implement cognition functions in great projects like Ersatz-Brain [2] and Cognitive Memory [3], the types of neural networks mentioned above are not adequate, as the attractor objects are too 'fragile". Moreover, multilayer neural networks, seen as implementations of nonlinear mappings, are not suitable for large scale problems. It is, however, worth noting, that the implementation of nonlinear mappings proposed in [4] and known as Regularized Least Squares Classification (RLSC), could be used for realization of very large scale associative memories. Relying on an RLSC approach, some novel structures of classifiers have been considered [5, 6, 7]. These structures are specific to using the Hamiltonian Neural Networks (HNN) based spectrum analysis, recognition and memorization, giving rise to the mapping of implementations with skew-symmetric kernels, as well. In this paper we present how to design very large scale nonlinear mappings of oscillator type, by using HNN, which, due to spectrum based information processing, can be seen as an implementation of holographic-like structures.

II. On Modelling of the Oscillatory Neural Networks

To our knowledge, the fundamental research in the field of oscillatory implementation of neural networks has been done by Hoppenstead [9, 10, 11, 12]. Let us briefly review that an oscillator can by described be the following state equation:

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}),\ \mathbf{x} \in \mathrm{R}^{m}, \tag{1}$$

and it is a nonlinear dynamical system with a limit cycle. Hence, a net of weakly coupled oscillators is given by:

$$\dot{\mathbf{x}}_i = \mathbf{f}_i(\mathbf{x}_i) + \varepsilon \mathbf{g}_i(\mathbf{x}_1, \ldots, \mathbf{x}_n, \varepsilon),\ \varepsilon << 1,\ i = 1, \ldots, n \tag{2}$$

Synchronization phenomenon in such a network is one of the most challenging mathematical and engineering problems. According to [11], the sufficient conditions for synchronization in the net (2) can be formulated as follows:

Transform the state space equation (2) onto phase equations:

$$\dot{\varphi}_i = \Omega_i + \varepsilon \mathrm{h}_i(\varphi_1, \ldots, \varphi_n, \varepsilon), \quad \varphi_i \in \mathrm{S}^1 \tag{3}$$

where: Ω_i – natural frequency of i-th oscillator (i.e. for $\varepsilon = 0$).

Assuming a weak coupling of oscillators, the state equation and phase equation can be simplified, as follows:

$$\dot{\mathbf{x}}_i = \mathbf{f}_i(\mathbf{x}_i) + \varepsilon \sum_{j=1}^{n} \mathbf{g}_{ij}(\mathbf{x}_i, \mathbf{x}_j) \quad \mathbf{x}_i \in \mathrm{R}^m \tag{4}$$

and

$$\dot{\varphi}_i = \Omega_i + \varepsilon \sum_{j=1}^{n} \mathrm{h}_{ij}(\varphi_i, \varphi_j) \quad i = 1, \ldots, n. \tag{5}$$

Introducing a phase deviation Ψ_i of i-th oscillator i.e.:

$$\varphi_i = \Omega_i t + \Psi_i \tag{6}$$

and averaging over a period $T = 2\pi/\Omega$, the phase equation (5) can be formulated as:

$$\dot{\Psi}_i = \varepsilon \sum_{j=1}^{n} H_{ij}(\Psi_i - \Psi_j), \qquad i = 1, \dots, n \qquad (7)$$

where nonlinear functions H_{ij}; i, j = 1, … n determine time evolution of momentary frequency of coupled oscillators in the net. It is clear, that state of synchronization is given by equilibria of differential equations (7), i.e. :

$$\varepsilon \sum_{j=1}^{n} H_{ij}(\Psi_i - \Psi_j) = 0, \; i = 1, \dots, n \qquad (8)$$

or

$$\Delta\omega_i + \sum_{j \neq i}^{n} H_{ij}(\Psi_i - \Psi_j) = 0; \qquad \forall\, i \qquad (9)$$

where: $\Delta\omega_i = H_{ii}(0)$ is a deviation of natural frequency Ω_i.

For steady state of synchronization the equilibria have to be asymptotically stable. Unfortunately, the general solution of Eq.(9) is a nontrivial task, for n >> 1. In special case, under assumption that $H_{ij}(\bullet)$ has a form:

$$H_{ij}(\Psi_i - \Psi_j) = H(\Psi_i - \Psi_j) = -\sin(\Psi_i - \Psi_j) \qquad (10)$$

the solution of equation (9) can be analytically found. The above case is known and celebrated as Kuramoto model [11, 13]. For example, for n =2, Kuramoto model is given by:

$$\begin{aligned} \frac{d\Psi_1}{d\tau} &= \Delta\omega_1 - \sin(\Psi_1 - \Psi_2) \\ \frac{d\Psi_2}{d\tau} &= \Delta\omega_2 + \sin(\Psi_1 - \Psi_2) \end{aligned} \qquad (11)$$

where: $\tau = \varepsilon t$.

It is worth noting that, assuming equation (1) as a model of an oscillatory neuron, state equation (4) describes an oscillatory neural network, which can be synchronized, as shown above. But, it seems that synchronization alone insufficiently determines a neural network as an information processor. We claim that neural networks, to be treated as information processors, have to function as orthogonal filters.

III. On Oscillatory Implementation of Orthogonal-Filters

The models of nonlinear mappings described in this paper rely on using Hamiltonian Neural Networks (HNN) based orthogonal filters. Let us note that HNN are nonlinear, dynamical structures composed of elementary lossless neurons. A basic d.c. model of a lossless neuron is shown in Fig.1 and its state space description is as follows:

$$\begin{bmatrix} \dot{z}_1 \\ \dot{z}_2 \end{bmatrix} = \begin{bmatrix} 0 & w \\ -w & 0 \end{bmatrix} \begin{bmatrix} \Theta(z_1) \\ \Theta(z_2) \end{bmatrix} + \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \qquad (12)$$

where activation function $\Theta(z)$ is passive and fulfills:

$$\mu_1 \le \frac{\Theta(z_i)}{z_i} \le \mu_2; \mu_1, \mu_2 \in [0, \infty), i = 1,2$$

x - input data
z – state vector

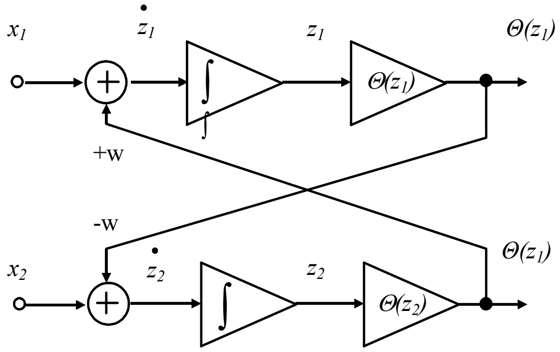

Figure 1. D.C. model of lossless neuron.

This model gives rise to the following notes:

1. From the point of view of circuit theory, a lossless neuron can be treated as a loop connection of a nonlinear inductor and capacitor, forming a passive nonlinear oscillator. Moreover, looking for an analogy between circuit theory and mechanics, one could consider the above mentioned oscillator as an energy model of a relativistic particle. Indeed, taking into consideration the following classical relationships:

$$F = \dot{p} \qquad (13)$$

$$v(p) = \frac{p}{m_0 \sqrt{1 + \left(\frac{p}{m_0 c}\right)^2}} \qquad (14)$$

where: F- external force
p – momentum
c – absolute velocity
m_0 - rest mass
v – velocity (activation function)

for free particle (F = 0), one obtains the basic model of a Hamiltonian particle, as shown in Fig.2

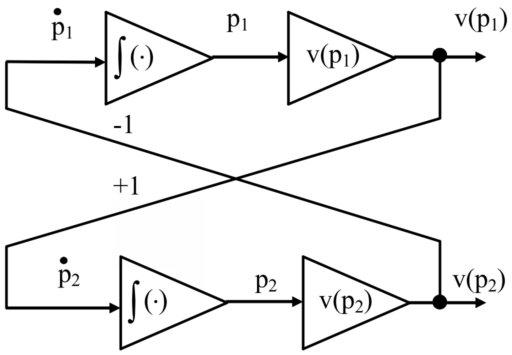

Figure 2. Model of Hamiltonian particle ("fermion")

Thus, Eq. (12) takes the following form:

$$\dot{p}_1 = v(p_2)$$
$$\dot{p}_2 = -v(p_1) \qquad (15)$$

Assuming, $p_1 = p_2 = p$, the whole internal energy stored in the particle shown in the Fig. 2. is given by Hamiltonian:

$$H = E = 2\int_{-\infty}^{t} \dot{p}\, v(p) dt = \frac{2}{m_0}\int_{0}^{p_{max}} \frac{p}{\sqrt{1+\left(\frac{p}{m_0 c}\right)^2}} dp = 2m_0c^2\left(\sqrt{1+\left(\frac{p_{max}}{m_0 c}\right)^2} - 1\right) \qquad (16)$$

Considering a linearization of function v(p), as shown in Fig.3.

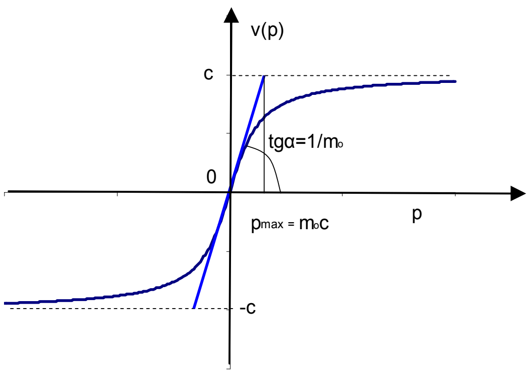

Figure 3. Velocity in SRT.

equation (16) takes the following form:

for $p_{max} = m_0c$ (see Fig..3.)

$$E = 2 \bullet 0.41 \bullet m_0c^2 = 0.82 \bullet m_0c^2 \qquad (17)$$

The classical amount of rest energy i.e., $E_0 = m_0c^2$, is obtained only under the assumption that function v(p) is piecewise linear. The model of particle used in this consideration can be seen as a Hamiltonian oscillator. Solutions of Eq.(15) are periodical and are dependent on initial values of momentum. Thus, for example, a numerical solution for initial values: $|p_{01}| = m_0c$, $p_{02} = 0$ or $|p_{02}| = m_0c$, $p_{01} = 0$ is presented in Fig. 4.

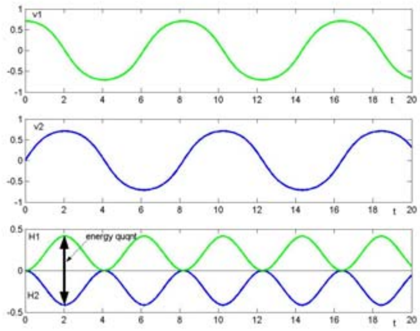

Figure. 4. A numerical solution of dynamical system Eq (15) for $p_{01} = 1$, $p_{02} = 0$ $m_0 = 1$, $c = 1$

$H_{1max} + H_{2max} = 0.82 \bullet m_0c^2$

where

$$H_{imax} = \max\left|\int_0^t \dot{p}_i v_i d\tau\right|$$

It can be seen, that the internal energy of a particle-oscillator has a form of quanta. For external observers, this energy is not visible. In the case of Eq.(15), force field is skew-symmetric, i.e., the internal forces are of "electromagnetic" type. However, this skew-symmetry can be easily changed into symmetry. Hence, the particle-oscillator is a "connection", via an internal "gravitation field" (symmetric), of matter ($m > 0$) and antimatter ($m < 0$). It is worth noting, that, by adequate interpretation of solutions (Fig.4.), one could easily obtain such objects like "spin" and "uncertainty". But, such interpretation would guide us to exotic physics.

2. A D.C. model of a lossless neuron can be, one-to-one, transformed into an oscillatory model, using two phase-locked-loops (PLLs). Such a PLL based model is shown in Fig. 5
.

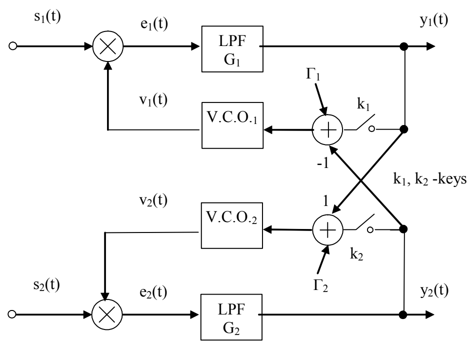

Figure. 5. Oscillatory model of lossless neuron.

It is easy to see that the model in Fig. 5 (PLL model) consists of two antisymmetrically coupled sinusoidal phase oscillators. Input signals $s_i(t)$, i = 1, 2 are sinusoidal carriers. Thus:

$$s_i(t) = A_{Ci} \sin(\Omega_i t + \Psi_{si}), \qquad (18)$$

$$v_i(t) = A_{Vi} \cos(\Omega_i t + \Psi_i);\ i = 1, 2. \qquad (19)$$

Assuming ideal transmittances of loop filters, i.e., $G_1 = G_2 \equiv 1$, the mean phase equation (Adler equation) of this model is as follows (keys k_1, k_2 open):

$$\frac{d}{dt}\begin{bmatrix}\Psi_{s1}-\Psi_1\\ \Psi_{s2}-\Psi_2\end{bmatrix} = 2\pi\begin{bmatrix}0 & \pm k_{V1}k_{m2}A_{C2}A_{V2}\\ \mp k_{V2}k_{m1}A_{C1}A_{V1} & 0\end{bmatrix}\begin{bmatrix}\sin(\Psi_{s1}-\Psi_1)\\ \sin(\Psi_{s2}-\Psi_2)\end{bmatrix} + \begin{bmatrix}\Delta\omega_1\\ \Delta\omega_2\end{bmatrix} - 2\pi\begin{bmatrix}k_{V1}\Gamma_1\\ k_{V2}\Gamma_2\end{bmatrix} \qquad (20)$$

where: $\Delta\omega_i$ - frequency deviations of input $s_i(t)$ signal
k_{Vi}, k_{mi} – sensitivity of VCO and phase-detector, respectively
Γ_i – input d.c. signal (i =1, 2)

The similarity between equation (20) and Kuramoto model is worth noting. Closing k_1, k_2- keys in model from Fig. 5. one obtains an elementary PLL orthogonal filter described by:

$$\frac{d}{dt}\begin{bmatrix}\Psi_{s1}-\Psi_1\\ \Psi_{s2}-\Psi_2\end{bmatrix} = 2\pi\begin{bmatrix}-k_{V1}k_{m1}A_{C1}A_{V1} & k_{V1}k_{m2}A_{C2}A_{V2}\\ -k_{V2}k_{m1}A_{C1}A_{V1} & -k_{V2}k_{m2}A_{C2}A_{V2}\end{bmatrix}\begin{bmatrix}\sin(\Psi_{s1}-\Psi_1)\\ \sin(\Psi_{s2}-\Psi_2)\end{bmatrix} + \begin{bmatrix}\Delta\omega_1\\ \Delta\omega_2\end{bmatrix} - 2\pi\begin{bmatrix}k_{V1}\Gamma_1\\ k_{V2}\Gamma_2\end{bmatrix} \qquad (21)$$

where assumed that connection matrix has a form:

$$\mathbf{W}_c = \mathbf{W} - w_0\mathbf{1} \qquad (22)$$

with $\mathbf{W}^2 = -\mathbf{1},\ \mathbf{W}^T = \mathbf{W}^{-1} = -\mathbf{W}$ (23)

and $w_0 > 0$ (**W** –skew-symmetric, orthogonal)

Let us note that PLL implementation of the elementary orthogonal filter from Fig.5. can be easily scaled up to n-dimensional space. Such a generalization is shown in Fig.6. [7].

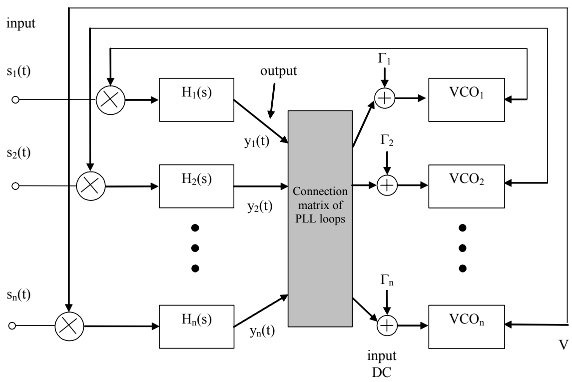

Figure 6. PLL model of n-dim neural network

The Adler equation of this model is given by:

$$\frac{d}{dt}\begin{bmatrix}\Psi_{s1}-\Psi_1\\ \Psi_{s2}-\Psi_2\\ \vdots\\ \Psi_{sn}-\Psi_n\end{bmatrix} = 2\pi\begin{bmatrix}-k_{v1}k_{m1}A_{C1}A_{v1} & \pm k_{v1}k_{m2}A_{C2}A_{v2} & \cdots & \pm k_{v1}k_{mk}A_{Ck}A_{vk}\\ \mp k_{v2}k_{m1}A_{C1}A_{v1} & \cdots & \cdots & \pm k_{v2}k_{mk}A_{Ck}A_{vk}\\ \cdots & \cdots & \cdots & \cdots\\ \mp k_{vk}k_{m1}A_{C1}A_{v1} & \mp k_{vk}k_{m2}A_{C2}A_{v2} & \cdots & -k_{vk}k_{mk}A_{Ck}A_{vk}\end{bmatrix}\begin{bmatrix}\sin(\Psi_{s1}-\Psi_1)\\ \sin(\Psi_{s2}-\Psi_2)\\ \vdots\\ \sin(\Psi_{sn}-\Psi_n)\end{bmatrix} + \begin{bmatrix}\Delta\omega_1\\ \Delta\omega_2\\ \vdots\\ \Delta\omega_n\end{bmatrix} - 2\pi\begin{bmatrix}k_{V1}\Gamma_1\\ k_{V2}\Gamma_2\\ \vdots\\ k_{Vn}\Gamma_n\end{bmatrix} \qquad (24)$$

where : $s_i(t) = A_{Ci} \sin(\Omega_i t + \Psi_{si})$
$v_i(t) = A_{Vi} \cos(\Omega_i t + \Psi_i)$
$\Delta\omega_i$ – frequency deviation
Γ_i – input d.c. signal
i = 1, … , n

Equation (24) can be rewritten as:

$$\dot{\mathbf{z}} = \mathbf{W_c} \sin \mathbf{z} + \boldsymbol{\Delta\omega} - \boldsymbol{\Gamma} \qquad (25)$$

where: $\mathbf{z} = [z_1, \dots, z_n]^T = [\Psi_{s1} - \Psi_1, \dots, \Psi_{sn} - \Psi_n]^T$
$\mathbf{W_c}$ – matrix of connections.

It is worth noting that:

1. The hold range of a PLL network is determined by the stable equilibrium of Eq.(25). It means that, for a given $\boldsymbol{\Delta\omega}$ and $\boldsymbol{\Gamma}$, one can find such loop gains ($k_v k_m A_c A_v$) that PLL network attains synchronization in point: $|\sin z_i| < 1$, i = 1, … , n.
2. Under synchronization, the steady-state output of PLL network is given by:

$$\mathbf{y} = \sin \mathbf{z} = \mathbf{W_c}^{-1}(\boldsymbol{\Gamma} - \boldsymbol{\Delta\omega}). \qquad (26)$$

   Taking connection matrix $\mathbf{W}_c$ as weight matrix in orthogonal filter, output **y** gives the Haar spectrum of the input vector. Moreover, the PLL network from Fig.6 can be treated as a n-dimensional F.M. signal demodulator.

3. The PLL network from Fig. 6. can be seen as a model of a neural network with dynamical connections. The weight of connections can be changed by parameter k_v (i.e. sensitivity of VCO).

IV. Orthogonal Filters Based Nonlinear Mappings

Nonlinear functions or mappings approximation can be implemented by using HNN-based orthogonal filters, which perform spectrum analysis and memorization. Function approximation, as known from machine learning, starts with training data $(\mathbf{x}_i, y_i)_{i=1}^m$, where input vectors $\mathbf{x}_i \in X \subset R^n$ and $y_i \in Y \subset R$. One synthesizes a multivariate function that optionally represents the relation between the input $\mathbf{x}_i$ and y_i. We use here a kernel representation, i.e.:

$$f(\mathbf{x}) = \sum_{i=1}^{m} c_i \mathbf{K}_{\mathbf{X}i}(\mathbf{x}) \qquad (27)$$

where: $\mathbf{c} = [c_1, c_2, \dots, c_m]^T$, $c_i \in R$

and kernels $K_{\mathbf{X}i}(\mathbf{x})$ are definite functions continuous on X x X. The weights c_i are such, to minimize the error on the training set, i.e., they can be found from the equation:

$$\mathbf{K}\,\mathbf{c} = \mathbf{y} \qquad (28)$$

where: $\mathbf{K}$ is the square matrix with elements $K_{i,j} = K_{\mathbf{X}i}(\mathbf{x}_j)$ and $\mathbf{y}$ is the vector with coordinates y_i.

For implementation of f($\mathbf{x}$), equation (28) has to be well-posed. One of the most important positive–definite kernels is the Gaussian:

$$K_{\mathbf{x}_i}(\mathbf{x}) = e^{-\|\mathbf{x}_i - \mathbf{x}\|^2 / 2\sigma^2}$$

giving structure known as RBF.
Generally, taking kernels as positive-definite functions, matrix $\mathbf{K}$ in Eq.(28) is positive-definite and hence Eq.(28) is well-posed. Moreover, taking into account the Tikhonov regularization, Eq.(28) can be reformulated as a key algorithm for RLSC structure, as follows [4]:

$$(\, m\gamma\, \mathbf{1} + \mathbf{K})\, \mathbf{c} = \mathbf{y} \qquad (29)$$

where: $\gamma > 0$ and $(\, m\gamma\, \mathbf{1} + \mathbf{K})$ is strictly positive.
The purpose of this paper is to show, how mappings, classifiers and associative memories can be implemented using HNN based orthogonal filters, which perform spectrum analysis. Spectrum analysis can be treated as a transform from input signal space into a feature space. Relying on RLSC approach, we propose to define here a skew-symmetric kernel $K_{\mathbf{u}i}(\mathbf{v})$:

$$K_{\mathbf{u}i}(\mathbf{v}) := \Theta(\mathbf{u}^T_i\, \mathbf{v}) \qquad (30)$$

where: $\mathbf{u}_i = (\mathbf{W}\text{-}\mathbf{1})\, \mathbf{x}_i$

$\mathbf{v} = -(\mathbf{W}+\mathbf{1})\, \mathbf{x}$

$\Theta(\,\bullet\,)$ is an odd function(e.g. sigmoid)

$\mathbf{W}^2 = -\mathbf{1}$, $\mathbf{W}^T\mathbf{W} = \mathbf{1}$

$w_0 > 0$

$\mathbf{u}_i$, $\mathbf{v}$ – Haar spectrum of input $\mathbf{x}_i$ and $\mathbf{x}$, respectively

Thus:

$$K_{\mathbf{u}i}(\mathbf{v}_j) = \Theta(\mathbf{u}^T_i\, \mathbf{v}_j) = \Theta(2\mathbf{x}^T_i \mathbf{W}\, \mathbf{x}_j)$$

and

$$K_{\mathbf{v}j}(\mathbf{u}_j) = -\, K_{\mathbf{u}i}(\mathbf{v}_j)$$

Hence, matrix

$\mathbf{K}_a = \{\mathbf{K}_{i,j}\} = \{\mathbf{K}_{\mathbf{u}i}(\mathbf{v}_j)\}$ is skew-symmetric

As mentioned above, the key problem in design of the approximation is the solvability of linear equation:

$$\mathbf{K}_a\, \mathbf{c} = \mathbf{y} \qquad (31)$$

Since $\mathbf{K}_a$ is skew-symmetric, it needs to be regularized, making equation (31) well-posed.
Hence, we propose the following regularization of matrix $\mathbf{K}_a$:

$$\mathbf{K}_r = (\gamma\, \mathbf{1} + \mathbf{K}_a) \qquad (32)$$

where: $\gamma \neq 0$

Then, the following design equation is well-posed for any $m < \infty$ (number of training vectors):

$$\mathbf{K}_r\, \mathbf{c} = \mathbf{y} \qquad (33)$$

One of possible architectures, implementing equation (27) with skew-symmetric kernels $K_{i,j}$ is shown in Fig.7.

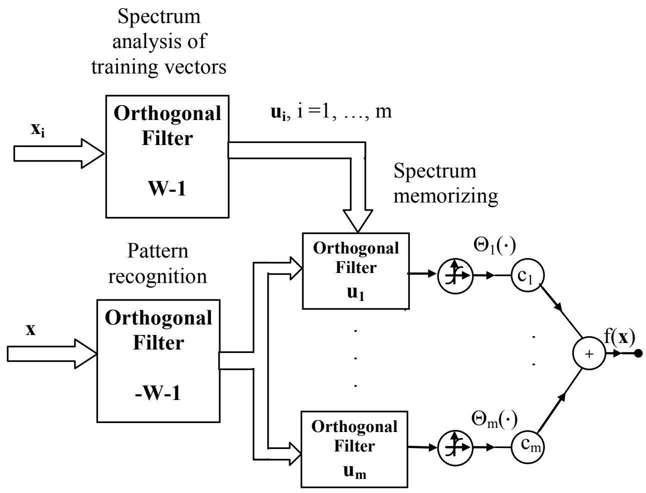

Figure 7. Structure of function f($\mathbf{x}$) approximator.

Thus, the unknown function f(($\mathbf{x}$) can be approximated by the structure from Fig.7. consisting of two HNN based spectrum analyzers and a set of m orthogonal filters memorizing the spectrum of m training points. It is easy to see that the activation functions $\Theta(\bullet)$ of neurons should be endowed with a "superconducting impulse" γ, as shown in Fig. 8.

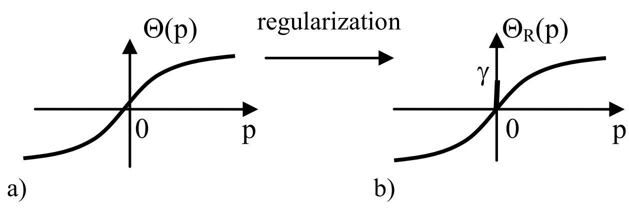

Figure 8. Activation function of neurons.

Due to the properties of matrix $\mathbf{K}_r$, a solution of key equation (33) exists for any number m of training points. It is clear that structure of the function approximator in Fig. 7, is based on oscillatory neural networks, as presented in Fig. 6.

V. Design of Orthogonal Filters

The main issue with the structure of the function approximator shown in Fig.7. is the design of the orthogonal filters. Such a design can be based on using the family of Hurwitz-Radon matrices. Indeed, a set of orthogonal, skew-symmetric matrices $\mathbf{A}_k$ with the following properties: $\mathbf{A}_j \mathbf{A}_k + \mathbf{A}_k\mathbf{A}_j = \mathbf{0}$, $\mathbf{A}_j^2 = \mathbf{-1}$ for $j \neq k$, $k = 1, \dots, s$ are known as a family of Hurwitz-Radon matrices. Any family of Hurwitz-Radon matrices (n x n) consists of s_{max} matrices, where $s_{max}=\rho(n)-1$ and Radon number $\rho(n) \leq n$. $\rho(n) = n$ for n = 2, 4, 8, only. For our purposes the following statements on Hurwitz-Radon matrices could be interesting [15]:

a. The maximum number of continuous orthogonal tangent vector fields on sphere $S^{n-1} \subset R^n$ is $\rho(n) - 1$.
b. Let $\mathbf{W}_1, \dots, \mathbf{W}_s$ be a set of orthogonal Hurwitz-Radon matrices and $w_1, \dots, w_s$ be real numbers:
Then:

$$\mathbf{W} = \sum_{i=1}^{s} w_i \mathbf{W}_i \tag{34}$$

is orthogonal and skew-symmetric.

Matrix **W** from Eq.(34) can be used for creating the weight matrices of HNN.
Thus, for example, matrix **W** for n = 8, 16, 32, … is given by:

$$\mathbf{W}_8 = \begin{bmatrix} 0 & w_1 & w_2 & w_3 & w_4 & w_5 & w_6 & w_7 \\ -w_1 & 0 & w_3 & -w_2 & w_5 & -w_4 & -w_7 & w_6 \\ -w_2 & -w_3 & 0 & w_1 & w_6 & w_7 & -w_4 & -w_5 \\ -w_3 & w_2 & -w_1 & 0 & w_7 & -w_6 & w_5 & -w_4 \\ -w_4 & -w_5 & -w_6 & -w_7 & 0 & w_1 & w_2 & w_3 \\ -w_5 & w_4 & -w_7 & w_6 & -w_1 & 0 & -w_3 & w_2 \\ -w_6 & w_7 & w_4 & -w_5 & -w_2 & w_3 & 0 & -w_1 \\ -w_7 & -w_6 & w_5 & w_4 & -w_3 & -w_2 & w_1 & 0 \end{bmatrix} \tag{35}$$

$$\mathbf{W}_{16} = \left[\begin{array}{c|c} \mathbf{W}_8 & \begin{matrix} w_8 & & \mathbf{0} \\ & \ddots & \\ \mathbf{0} & & w_8 \end{matrix} \\ \hline \begin{matrix} -w_8 & & \mathbf{0} \\ & \ddots & \\ \mathbf{0} & & -w_8 \end{matrix} & \mathbf{W}_8^T \end{array}\right]$$

$$\mathbf{W}_{32} = \left[\begin{array}{c|c} \mathbf{W}_{16} & \begin{matrix} w_9 & & \mathbf{0} \\ & \ddots & \\ \mathbf{0} & & w_9 \end{matrix} \\ \hline \begin{matrix} -w_9 & & \mathbf{0} \\ & \ddots & \\ \mathbf{0} & & -w_9 \end{matrix} & \mathbf{W}_{16}^T \end{array}\right]$$

where: $w_8, w_9 \in R$

Hurwitz-Radon matrices for other n can be found elsewhere. It can be seen, that a basic component of matrix family for n=8, 16, 32, … is an eight dimensional matrix. Hence, one obtains the following statement: Structures of orthogonal filters, used for realization of functions and mappings models (Fig.7), can be based on compatible composition of 8-dim. building blocks (octonionic modules). Octonionic module performs the following transformation:

$$\mathbf{y} = \mathbf{H}_8\mathbf{x} \tag{36}$$

where: **x** and **y** are 8-dim, input and output vectors, respectively.
Transformation matrix $\mathbf{H}_8$ has a form:

$$\mathbf{H}_8 = \begin{bmatrix} -h_0 & h_1 & h_2 & h_3 & h_4 & h_5 & h_6 & h_7 \\ -h_1 & -h_0 & h_3 & -h_2 & h_5 & -h_4 & -h_7 & h_6 \\ -h_2 & -h_3 & -h_0 & h_1 & h_6 & h_7 & -h_4 & -h_5 \\ -h_3 & h_2 & -h_1 & -h_0 & h_7 & -h_6 & h_5 & -h_4 \\ -h_4 & -h_5 & -h_6 & -h_7 & -h_0 & h_1 & h_2 & h_3 \\ -h_5 & h_4 & -h_7 & h_6 & -h_1 & -h_0 & -h_3 & h_2 \\ -h_6 & h_7 & h_4 & -h_5 & -h_2 & h_3 & -h_0 & -h_1 \\ -h_7 & -h_6 & h_5 & h_4 & -h_3 & -h_2 & h_1 & -h_0 \end{bmatrix} \tag{37}$$

where: columns (and rows) constitute the orthogonal basis, i.e., the output vector **y** gives the Haar spectrum of **x**. Moreover, for given $\mathbf{x}_0 = [x_1, \dots, x_8]^T$ and $\mathbf{y}_0 = [y_1, \dots, y_8]^T$ one sets up so called best adapted basis:

$$\begin{bmatrix} h_0 \\ h_1 \\ h_2 \\ h_3 \\ h_4 \\ h_5 \\ h_6 \\ h_7 \end{bmatrix} = \frac{1}{\sum_{i=1}^{8} x_i^2} \begin{bmatrix} y_1 & y_2 & y_3 & y_4 & y_5 & y_6 & y_7 & y_8 \\ -y_2 & y_1 & -y_4 & y_3 & -y_6 & y_5 & y_8 & -y_7 \\ -y_3 & y_4 & y_1 & -y_2 & -y_7 & -y_8 & y_5 & y_6 \\ -y_4 & -y_3 & y_2 & y_1 & -y_8 & y_7 & -y_6 & y_5 \\ -y_5 & y_6 & y_7 & y_8 & y_1 & -y_2 & -y_3 & -y_4 \\ -y_6 & -y_5 & y_8 & -y_7 & y_2 & y_1 & y_4 & -y_3 \\ -y_7 & -y_8 & -y_5 & y_6 & y_3 & -y_4 & y_1 & y_2 \\ -y_8 & y_7 & -y_6 & -y_5 & y_4 & y_3 & -y_2 & y_1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \\ x_8 \end{bmatrix} \tag{38}$$

It means that a given $\mathbf{x}_0$ is transformed into a given $\mathbf{y}_0$ ($\mathbf{x}_0 \rightarrow \mathbf{y}_0$) by the orthogonal filter with weight matrix given by Eq.(38). Thus, a classical perceptron performing a scalar product can be implemented by an orthogonal filter with best adapted basis ($\mathbf{x}_0 \rightarrow \mathbf{y}_0$), as shown in Fig. 9.:

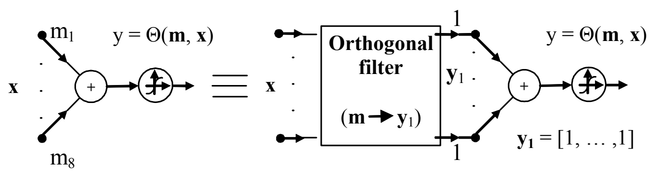

Figure 9. Implementation of perceptron by orthogonal filter.

It is worth noting that the implementation in Fig.9. relies on a linear summing of the output spectrum of the orthogonal filter. Orthogonal filters, used in Fig.7. for spectrum memorizing, have structure as shown in Fig.9., and they are implementable as oscillatory (PLL) octonionic modules.

Hence, one obtains:

the overall structure from Fig.7. can be implemented by connection of oscillatory (PLL) octonionic modules.

VI. SIMULATIONS

As mentioned above, Eq.(24) and (25) describe the mean equation of a PLL network. But, in real implementations, the loop filters have to be taken into consideration. In other words, one has to pose a question: Is it possible to find such loop filters where full phase equation and Adler equation for PLL network are approximately the same? From an analytical point of view such a possibility exists, since a PLL network has a stable integral manifold [8]. This possibility has been experimentally proven by simulations using the makro-models of PLL, offered by Matlab-Simulink.

Some simulations of octonionic modules have been performed by using a general PLL model from Fig.6.. Full analysis using Matlab-Simulink macro-models of phase-locked loops, endowed with different loop filters, showed that algebraic transformation given by Eq. 36. can be, under synchronization, exactly performed by oscillatory structure. Moreover, this structure sets up an oscillatory memory cell, according to solution presented in Fig. 9.

VII.CONCLUSION

The main issue considered in this paper is the design of mappings. Mappings designed here rely on multivariate function approximations with skew-symmetric kernels, giving rise to very large scale classifiers and associative memories. Due to regularization, such classifiers and memories can be implemented for any even n (dimension of input vector space) and any $m < \infty$ (number of training patterns). Accuracy of classification depends on phase-space geometry of mappings. It can be changed by appropriate covering of the neighborhood of the approximation points. Kernels utilized in function and mapping approximation are implemented by using HNN based orthogonal filters. Thus, classifiers and memories, here designed, can exist as numerically stable algorithms or physical devices, performing their functions in real-time. Moreover, we have proposed oscillatory (PLL) implementation of mappings. Presented in this paper PLL neural networks can be seen as a special problem in the theory of coupled oscillators. To our knowledge, orthogonal filters based information processing can be considered as inspired by biological systems.

REFERENCES

[1] K. H. Pribram, "Free Will: The Brain as an Anticipatory System", *Proceedings of Third International Conference on Computing Anticipatory Systems,* AIP Conference Proceedings No 517, Liege, Belgium, 1999.

[2] J. A. Anderson, P. Allopenna, "The Ersatz Brain Project: Neural Inspiration, Cognitive Application*", Proceedings of IEEE International Join Conference on Neural Networks*, Vancouver, Canada, 2006.

[3] B. Widrow, J. C. Aragon, "Cognitive Memory", *Computational Intelligence: Principles and Practice, Edited by G.Y. Yen and D.B. Fogel, IEEE World Congress on Computational Intelligence,* Vancouver, Canada, 2006.

[4] T. Poggio, S. Smale, "The Mathematics of Learning: Dealing with Data", *Notices of the AMS*, vol. 50, no. 5, pp. 537-544, 2003.

[5] W. Sieńko, D. Zamojski, "Hamiltonian Neural Networks Based Classifiers and Mappings", *Proceedings of IEEE International Join Conference on Neural Networks*, Vancouver, Canada, 2006.

[6] W. Sieńko, D. Zamojski, "On Geometry of Hamiltonian Neural Networks Based Classifiers", *Proceedings of IEEE International Conference on Signals and Electronic Systems*, Łódź, Poland, pp. 471-474, 2006.

[7] W. Sieńko, W. Citko, "Orthogonal Filter-Based Classifiers and Associative Memories", *Proceedings of International Joint Conference on Neural Networks (IJCNN*), Orlando, USA, 2007.

[8] J. Kudrewicz, "Dynamics of Phase Loop" WNT, Warszawa 1991(in polish)

[9] F. C. Hoppenstead, E. M. Izhikevich, "Weakly Connected Neural Network", *Springer*, New York 1997.

[10] E. M. Izhikevich, "Weakly connected quasiperiodic oscillators, FM interactions, and multiplexing in the brain", *SIAM Journal on Applied Mathematics 59*, pp.. 2193-2223, 1999.

[11] E. M. Izhikevich, "Dynamical Systems in Neuroscience: The Geometry of Excitability and Bursting*", Cambridge, MA: The MIT Press*, 2006.

[12] F. C. Hoppensteadt, E. M. Izhikevich, "Pattern Recognition Via Synchronization in Phase-Locked Loop Neural Networks", *IEEE Transactions on Neural Networks*, vol. 11, No.2, May 2000.

[13] S. H. Strogatz, „From Kuramoto to Crawford exploring the onset of synchronization in populations of coupled oscillators", *Physica D* 143: p. 1-20, 2000.

[14] W. Sieńko, "Quantum Aspect of Passive Neural Networks", *Proc. of Third International Conference on Computing Anticipatory Systems*, Liege, Belgium, Sym. 2, p.18, 1999.

[15] B. Eckmann, "Topology, Algebra, Analysis-Relations and Missing Links", *Notices of the AMS*, vol. 46, no. 5, pp. 520-527, 1999.

Wieslaw Citko received his Ph.D. degree in electrical engineering from Technical University of Lodz in 2000. Currently he is an Associate Professor at the Telecommunications Department of the Electrical Engineering Faculty of Gdynia Maritime University. His research interests include nonlinear circuits, neural networks and signal processing.

Wieslaw Sienko received his Ph.D. and D.Sc. degrees in electrical engineering from the Technical University of Gdansk. Currently he is a Professor at the Telecommunications Department of the Electrical Engineering Faculty of Gdynia Maritime University. His research interests include circuits theory, neural networks, quantum algorithms and signal processing.