Design of 2-Dimensional Recursive Filters by using Neural Networks

Valeri M. Mladenov
Department of Theoretical Electrotechnics
Faculty of Automation
Technical University of Sofia
1756, Sofia
BULGARIA
TEL-FAX:+339 2 636 2388,

Nikos E. Mastorakis
Military Institutions of University Education
Hellenic Naval Academy
Chair of Computer Science
Terma Hatzikyriakou, 18539, Piraeus,
GREECE
TEL-FAX:+301 777 5660,

Abstract: A new design method for two-dimensional (2-D) recursive digital filters is investigated. The design of the 2-D filter is reduced to a constrained minimization problem the solution of which is achieved by the convergence of an appropriate Neural Network. An illustrative example is given and a comparison with the results of previous methods is attempted. Many advantages of the present method against previous methods of the literature can be ascertained.

Key-Words: Two-Dimensional Recursive Filters, Constrained Optimization, Neural Networks

1. Introduction

During the last two decades many authors have proposed various methods for the design of 2-D (recursive or non-recursive) discrete signal, linear and shift invariant filters. An excellent overview is given in [3].

This growing interest for the design of 2-D filters is due to a variety of applications in fields as digital image processing, medical data processing, artificial vision, radar and sonar data processing, remote sensing, pattern recognition, numerical stereoscopy, astronomy and applied physics, biomedical engineering, biochemistry, robotics and mechanical engineering [1],[2].

Design approaches for 2-D filters can be broadly classified into two categories:

i) based on appropriate transformation of 1-D filters [2], [3]

ii) based on appropriate optimization techniques [3÷10]

The stability of the designed filters is essential for their practical implementation. However, most of the existing algorithms [3÷10] may result in an unstable filter. Various receipts have been proposed in order to overcome these instability problems, but the outcome is likely to be a system that has a very small stability margin and therefore no of essential practical importance.

In this paper, an optimization procedure is adopted by using continuous-time Artificial Neural Network (NN). The desired stability of 2-D filter yields our appropriate constraints for the minimization problem. Furthermore, an extension of the method is given in which we pre-determine the stability margin of the filter and therefore we know if the designed filter is stable and how stable is.

Artificial Neural Networks or simply Neural Networks (NN) have already been used to obtain solution of constrained optimization problems [11]. In 1984 Chua and Lin [12] developed the canonical non-linear programming circuit, using the Kuhn-Tucker conditions from the mathematical programming theory. Later, Tank and Hopfield [13] developed an optimization network for solving linear programming problems. Some practical design problems of their network along with its stability properties are discussed in [14].

An extension of the results of Tank and Hopfield to more general non-linear programming problems is presented in [15]. The authors noted that the network introduced by Tank and Hopfield could be considered to added to account for the dynamic behavior of the circuit. The above discussed approach implicitly utilizes the penalty function method [11], [16] where a constrained optimization problem is approximate by an unconstrained optimization problem. In [17], the authors use the penalty function method approach and synthesize a new neural optimization network capable of solving a general class of constrained optimization problems. The proposed architecture can be viewed as a continuous Neural Network model and in [18], the authors use SIMULINK for modeling and simulations of its behavior.

The advantage of using Neural Networks for solving a 2-D recursive filter design optimization problem is in the rapid action of the network (VLSI scheme), where the solution is obtained in real time. We present a Neural Network approach for solving 2-D recursive filters design problem.

The paper is outlined as follows. In the next section, the problem for design of 2-D recursive filters is described. In section 3, a Neural Network for solving the design problem is utilized. In section 4, numerical example for testing the Neural Network is illustrated. The example is the example of [3] and [4] and can show the advantages of the method against the previous ones. Finally, conclusion remarks are given.

2. Problem Formulation

Consider M_d, the desirable amplitude response of a 2-D filter as a function of the frequencies ω_1, ω_2 (ω_1, $\omega_2 \in [0, \pi]$). The design task at hand amounts to finding a transfer function $H(z_1,z_2)$ such that the function $H(e^{j\omega_1}, e^{j\omega_2})$ approximates the desired amplitude response $M_d(\omega_1, \omega_2)$.

For the design purposes we consider that

$$H(z_1,z_2) = H_0 \frac{\sum_{i=0}^{K} \sum_{j=0}^{K} a_{ij} z_1^i z_2^j}{\prod_{k=1}^{K} (1 + b_k z_1 + c_k z_2 + d_k z_1 z_2)}, \quad a_{00} = 1 \tag{1}$$

Our aim is to minimize

$$J = J(a_{ij}, b_k, c_k, d_k, H_0) = \sum_{n_1=0}^{N_1} \sum_{n_2=0}^{N_2} \left[|M(\omega_1,\omega_2)| - M_d(\omega_1,\omega_2) \right]^p \tag{2}$$

where

$$M(\omega_1,\omega_2) = H(z_1,z_2) \Big|_{\substack{z_1 = e^{-j\omega_1} \\ z_2 = e^{-j\omega_2}}} \tag{3}$$

$\omega_1 = \frac{\pi}{N_1} n_1$, $\omega_2 = \frac{\pi}{N_2} n_2$ and p is an even positive integer (usually p=2 or p=4) or equivalently to minimize

$$J = \sum_{n_1=0}^{N_1} \sum_{n_2=0}^{N_2} \left[\left| M\left(\frac{\pi n_1}{N_1}, \frac{\pi n_2}{N_2}\right) \right| - M_d\left(\frac{\pi n_1}{N_1}, \frac{\pi n_2}{N_2}\right) \right]^p \tag{4}$$

e.g. the aim is to minimize the difference between actual and desired amplitude response of the filter in $N_1 N_2$ points. It is known that a linear shift-invariant causal single-input single-output 2-D system described by the following transfer function: $G(z_1, z_2) = \frac{A(z_1, z_2)}{B(z_1, z_2)}$ - where $A(z_1, z_2)$ and $B(z_1, z_2)$ are coprime polynomials in the independent complex variables z_1 and z_2 - is stable if and only if the following two conditions are fulfilled:

a) $B(0, z_2) \neq 0$, for $|z_2| \leq 1$ and

b) $B(z_1, z_2) \neq 0$, for $|z_1| \leq 1$, $|z_2| = 1$

Here by the term Stability, we mean Bounded Input Bounded Output (BIBO) Stability. One should note that the first condition is relatively easy to check using any 1-D stability test. However, the second condition is more difficult since it includes two variables. We also assume that there are no nonessential singularities of the second kind on the closed unit bi-disk, i.e. there are no points (z_1, z_2) with $|z_1| \leq 1$ and $|z_2| \leq$ such that $A(z_1, z_2) = B(z_1, z_2) = 0$.

For our design purposes, since we deal with first-degree factors in denominator, it had been proved, [1÷3], that the stability conditions are given by

$$|b_k + c_k| - 1 < d_k < 1 - |b_k - c_k| \quad, k = 1,2,\ldots K$$

or

$$\left| \begin{array}{l} |b_k + c_k| - 1 < d_k \\ d_k < 1 - |b_k - c_k| \end{array} \right. \quad, k = 1,2,\ldots K \tag{5}$$

Thus the design of 2-D recursive filters is equivalent of the following constrained minimization problem

$$\min J = \sum_{n_1=0}^{N_1} \sum_{n_2=0}^{N_2} \left[\left| M\left(\frac{\pi n_1}{N_1}, \frac{\pi n_2}{N_2}\right) \right| - M_d\left(\frac{\pi n_1}{N_1}, \frac{\pi n_2}{N_2}\right) \right]^p$$

subject to: (6)

$$\left| \begin{array}{l} |b_k + c_k| - 1 < d_k \\ d_k < 1 - |b_k - c_k| \end{array} \right. \quad, k = 1,2,\ldots K$$

where p is an even positive integer (usually p=2 or p=4) and N_1, N_2 and K are given positive integers.

3. Neural Network for Solving the Design Problem

To simplify the notations and without loss of generality we will consider the case K=2 and $H(z_1, z_2)$ from (1) is

$$H(z_1, z_2) = H_0 \frac{a_{00} + a_{01} z_2 + a_{02} z_2^2 + a_{10} z_1 + a_{20} z_1^2 + a_{11} z_1 z_2 + a_{12} z_1 z_2^2 + a_{21} z_1^2 z_2 + a_{22} z_1^2 z_2^2}{(1 + b_1 z_1 + c_1 z_2 + d_1 z_1 z_2)(1 + b_2 z_1 + c_2 z_2 + d_2 z_1 z_2)}$$

The 2-D recursive filter design problem (6) consists inequality constraints only and this is the reason we utilize the Neural Network [1] shown on Fig.1 for solving the constrained optimization problem given in a form:

$$\begin{array}{l} \min f(x) \\ \text{subject to} \\ g_i(x) > 0 \;, i = 1,2,\ldots,m; \;\; x = (x_1, x_2, \ldots, x_n)^T, \end{array} \tag{7}$$

where $S_i = \begin{cases} 0, \text{ if } g_i > 0 \\ 1, \text{ if } g_i \le 0 \end{cases}$ $k_i>0$, $i=1,2,\ldots,m$ and $\mu_j>0$, $j=1,2,\ldots,n$.

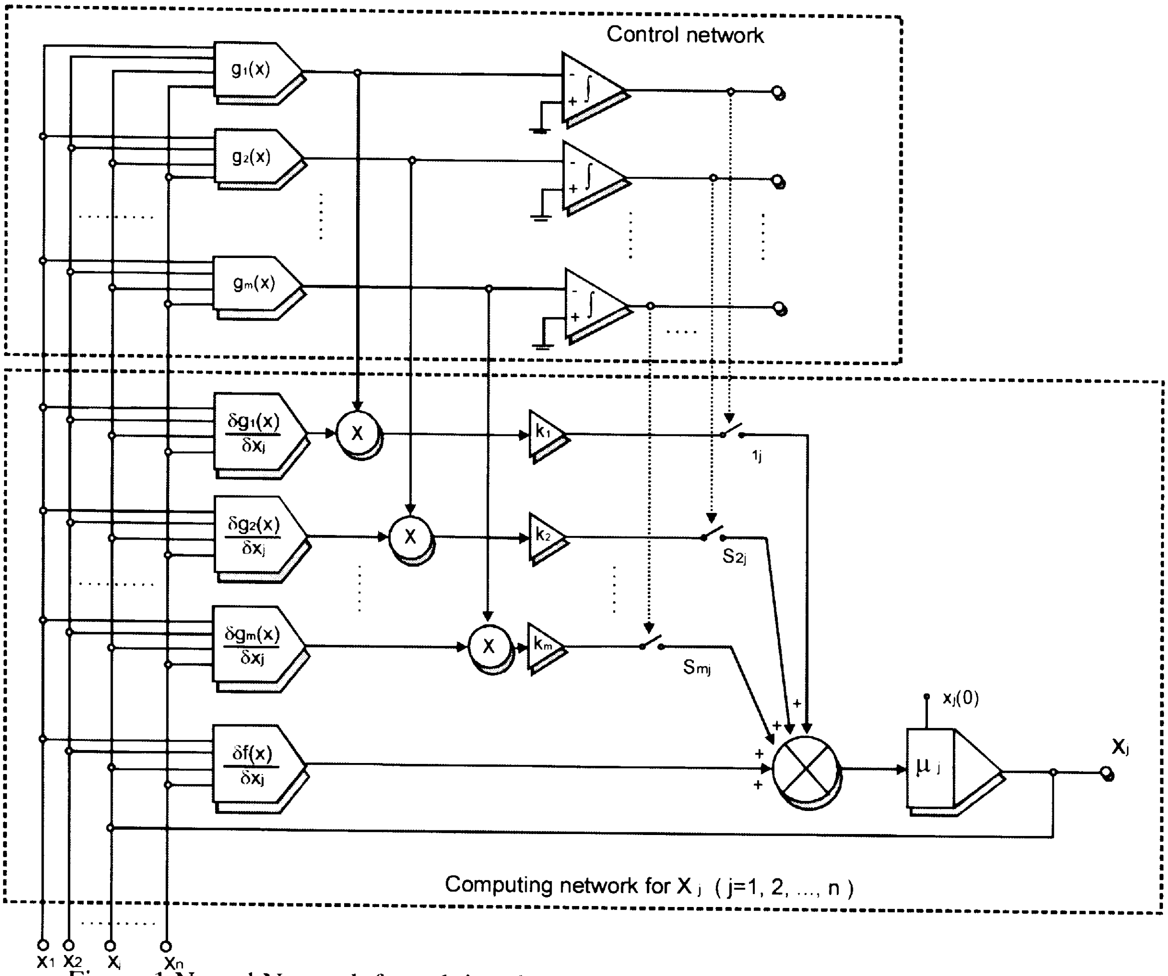

Figure 1 Neural Network for solving the constrained optimization problem

For successful implementation of the network the goal function's component $|M(\omega_1,\omega_2)|$ and the constraints shown in formula (5) must be presented in an appropriate form.

From (4) and (3)

$$M(\omega_1,\omega_2) = H_0 \frac{[a_{00}+a_{01}c_{01}+a_{02}c_{02}+a_{10}c_{10}+a_{20}c_{20}+a_{11}c_{11}+a_{12}c_{12}+a_{21}c_{21}+a_{22}c_{22}]-}{[(1+b_1c_{10}+c_1c_{01}+d_1c_{11})-j(b_1s_{10}+c_1s_{01}+d_1s_{11})]*}$$

$$\ldots \frac{-j[a_{01}s_{01}+a_{02}s_{02}+a_{10}s_{10}+a_{20}s_{20}+a_{11}s_{11}+a_{12}s_{12}+a_{21}s_{21}+a_{22}s_{22}]}{*[(1+b_2c_{10}+c_2c_{01}+d_2c_{11})-j(b_2s_{10}+c_2s_{01}+d_2s_{11})]}$$

where

$$c_{ij} = c_{ij}(\omega_1,\omega_2) = \cos(i\omega_1 + j\omega_2)$$
$$s_{ij} = s_{ij}(\omega_1,\omega_2) = \sin(i\omega_1 + j\omega_2) \qquad i,j = 0,1,2 \tag{8}$$

In a compact form $M(\omega_1,\omega_2)$ is

$$M(\omega_1,\omega_2)=H_0\frac{A_R-jA_I}{(B_{1R}-jB_{1I})(B_{2R}-jB_{2I})} \tag{9}$$

where

$$\begin{aligned}
A_R &= a_{00}+a_{01}c_{01}+a_{02}c_{02}+a_{10}c_{10}+a_{20}c_{20}+a_{11}c_{11}+a_{12}c_{12}+a_{21}c_{21}+a_{22}c_{22}\\
A_I &= a_{01}s_{01}+a_{02}s_{02}+a_{10}s_{10}+a_{20}s_{20}+a_{11}s_{11}+a_{12}s_{12}+a_{21}s_{21}+a_{22}s_{22}\\
B_{1R} &= 1+b_1c_{10}+c_1c_{01}+d_1c_{11}\\
B_{1I} &= b_1s_{10}+c_1s_{01}+d_1s_{11}\\
B_{2R} &= 1+b_2c_{10}+c_2c_{01}+d_2c_{11}\\
B_{2I} &= b_2s_{10}+c_2s_{01}+d_2s_{11}
\end{aligned} \tag{10}$$

Therefore we obtain for $|M(\omega_1,\omega_2)|$

$$|M(\omega_1,\omega_2)|=H_0\frac{\sqrt{A_R^2+A_I^2}}{(B_{1R}^2+B_{1I}^2)(B_{2R}^2+B_{2I}^2)} \tag{11}$$

A continuous differentiable form of the constraints has been obtained after considering (5)

$-(1+d_k)<(b_k+c_k)<1+d_k$
$-(1-d_k)<(b_k-c_k)<1-d_k$
$1+d_k>0$
$1-d_k>0$

or

$$\begin{aligned}
&g_{6(k-1)+1}(x)=b_k+c_k+d_k+1>0\\
&g_{6(k-1)+2}(x)=-b_k-c_k+d_k+1>0\\
&g_{6(k-1)+3}(x)=b_k-c_k-d_k+1>0\\
&g_{6(k-1)+4}(x)=-b_k+c_k-d_k+1>0\\
&g_{6(k-1)+5}(x)=1+d_k\\
&g_{6(k-1)+6}(x)=1-d_k,\ k=1,2,\dots,K
\end{aligned} \tag{12}$$

where

$$x=(a_{01},a_{02},a_{10},a_{11},a_{12},a_{20},a_{21},a_{22},b_1,b_2,c_1,c_2,d_1,d_2,H_0)^T \tag{13}$$

is a vector of unknown parameters and as a result we use the goal function $f(x)=J$ from (3), subject to 6K constraints (12).

The convergence of the network in Fig. 1 has been proved in [1] and the solution of the 2-D recursive filters design problem could be found after transient of the network settles down.

4. Numerical Example for testing the Neural Network

We consider the design of 2-D recursive filter (1) for the case K=2 or

$$H(z_1, z_2) = H_0 \frac{\sum_{i=0}^{2} \sum_{j=0}^{2} a_{ij} z_1^i z_2^j}{\prod_{k=1}^{2} \left(1 + b_k z_1 + c_k z_2 + d_k z_1 z_2\right)} \tag{14}$$

and consider the desired amplitude response

$$M_d(\omega_1, \omega_2) = \begin{cases} 1 & \text{if} \quad \sqrt{\omega_1^2 + \omega_2^2} \le 0.08\pi \\ 0.5 & \text{if} \quad 0.08\pi \le \sqrt{\omega_1^2 + \omega_2^2} \le 0.12\pi \\ 0 & \text{otherwise} \end{cases} \tag{15}$$

We chose p=2, N_1=50 and N_2=50 and the corresponding constrained optimization problem become:

$$\min J = \sum_{n_1=0}^{50} \sum_{n_2=0}^{50} \left[\left| M\left(\frac{\pi}{50} n_1, \frac{\pi}{50} n_2\right) \right| - M_d\left(\frac{\pi}{50} n_1, \frac{\pi}{50} n_2\right) \right]^2 \tag{16}$$

subject to

$$\begin{aligned}
&g_1(x)=b_1+c_1+d_1+1>0\\
&g_2(x)=-b_1-c_1+d_1+1>0\\
&g_3(x)=b_1-c_1-d_1+1>0\\
&g_4(x)=-b_1+c_1-d_1+1>0\\
&g_5(x)=d_1+1>0\\
&g_6(x)=-d_1+1>0\\
&g_7(x)=b_2+c_2+d_2+1>0\\
&g_8(x)=-b_2-c_2+d_2+1>0\\
&g_9(x)=b_2-c_2-d_2+1>0\\
&g_{10}(x)=-b_2+c_2-d_2+1>0\\
&g_{11}(x)=d_2+1>0\\
&g_{12}(x)=-d_2+1>0
\end{aligned} \tag{17}$$

where $x=(a_{01}, a_{02}, a_{10}, a_{11}, a_{12}, a_{20}, a_{21}, a_{22}, b_1, b_2, c_1, c_2, d_1, d_2, H_0)^T$

For better quality of the desired result we use a weight 1000 for $n_1=0$, $n_2=0$ e.g. for the point $\omega_1=0$, $\omega_2=0$, while the weights for other 2600 points are 1. After the transient of the Neural Network from Fig. 1, we find the following vector with optimal parameters:

x=[1.8922 -1.2154 0.0387 -2.5298 0.3879 0.6115 -1.4619 2.5206 -0.8707 -0.8729 -0.8705 -0.8732 0.7756 0.7799 -0.0010]^T.

Therefore

$$H(z_1, z_2) = -.001 \frac{1 + 0.0387 z_1 + 0.6115 z_1^2 - 2.5298 z_1 z_2 + 0.3879 z_1 z_2^2 + 1.8922 z_2 - 1.4619 z_1^2 z_2 + 2.5206 z_1^2 z_1^2 - 1.2154 z_2^2}{\left(1 - .8707 z_1 - 0.8705 z_2 + .7756 z_1 z_2\right)\left(1 - .8729 z_1 - 0.8732 z_2 + .7799 z_1 z_2\right)}$$

The corresponding amplitude response $|M(\omega_1,\omega_2)|$ is given in Fig. 2, while in Fig.3 one can obtain the amplitude response of the desired ideal $|M_d(\omega_1, \omega_2)|$. Furthermore in Fig.4, we present, for comparison, the result of the method of [3],[4].

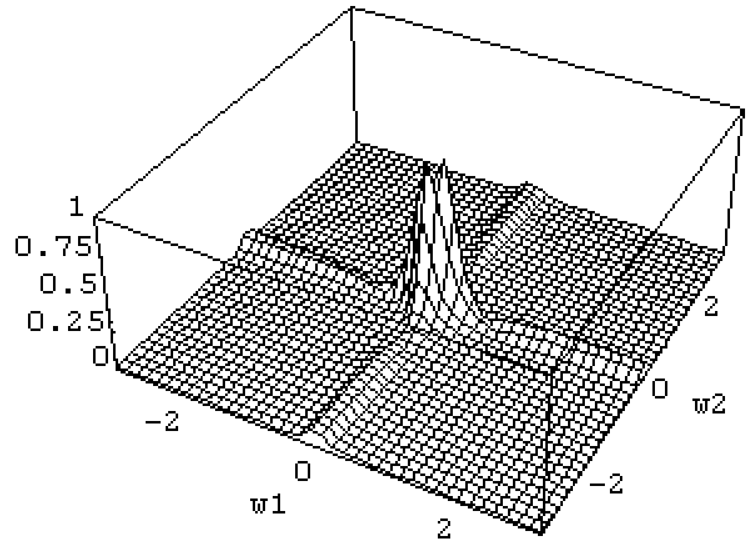

Figure 2 Obtained amplitude response $|M(\omega_1,\omega_2)|$ of the considered 2-D filter

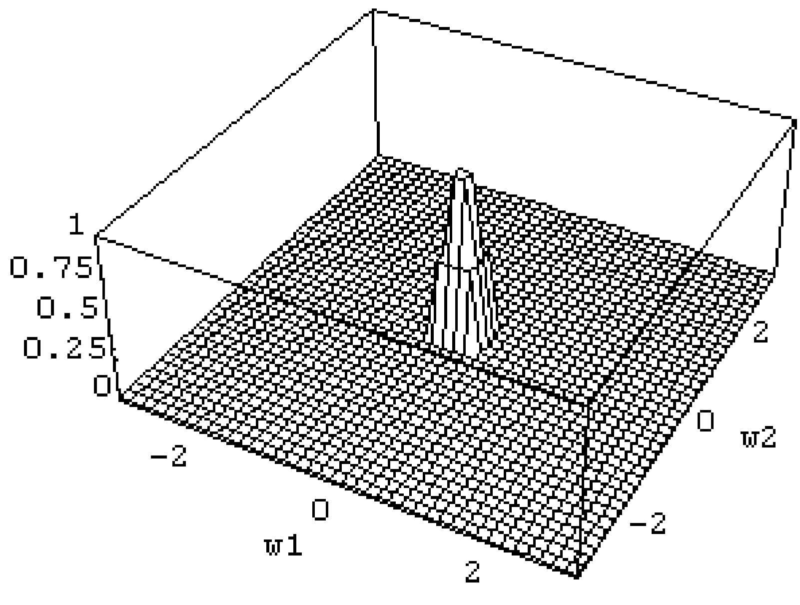

Figure 3 Obtained amplitude response $|M_d(\omega_1, \omega_2).|$ of the desirable (ideal) 2-D filter

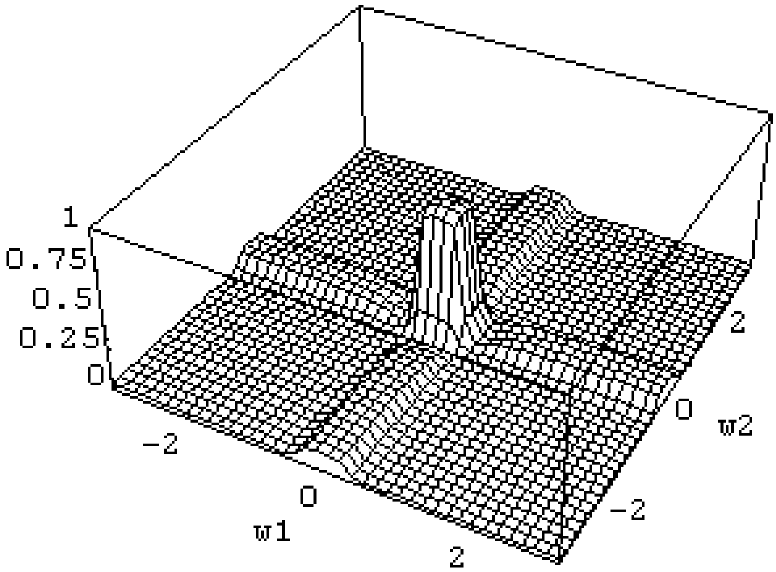

Figure 4 Obtained amplitude response $|M(\omega_1,\omega_2)|$ of the considered 2-D filter by using the method of [3],[4]

The advantages of the present method against the method of [3] and [4] are: *a)* We can check the stability of the designed filter from the beginning of the procedure, since we introduce the desired stability as appropriate constraints. On the contrary, the previous methods which are based more or less on a trial-and-error approach can not always guarantee the stability of the filter. *b)* We implement a simpler filter since, in practice, we have to realize a factorable numerator and in particular of first-order filters which obviously are simpler than those of [3],[4].

5. Conclusions

In this study, a Neural Network approach in the design of 2-D recursive filters is adopted. The design problem is reduced to a constrained optimization problem and a continuous Neural Network is used in order to find the optimal solution. We give the general form of the network and a specific numerical example that show the applicability, the efficiency and the elegance of the method in a real design. Further, an extension of the method is presented.

In Section 4, the advantages of the method against previous ones of the 2-D systems bibliography have been discussed in details. More specifically, the method appears to: a) give guarantee for the stability of the designed filter b) yield simpler filter implementation.

References:

[1] T.Kaczorek, *Two-Dimensional Linear Systems*, Springer-Verlag, Lecture Notes in Control and Information Sciences, Berlin-Heidelberg, 1985.

[2] S.G.Tzafestas (Editor), *Multidimensional Systems, Techniques and Applications*, Marcel Dekker, New York, 1986.

[3] W.-S. Lu and A. Antoniou, *Two-Dimensional Digital Filters*, Marcel Dekker, New York, 1992.

[4] G. A. Maria and M. M. Fahmy "An lp Design Technique for Two-Dimensional Digital Recursive Filters", *IEEE Trans. on Acoust., Speech, Signal Process.*, Vol.22, pp. 15-21, Feb. 1974.

[5] C. Charalambous, "Design of 2-Dimensional Circularly-Symmetric Digital Filters" *IEE Proceedings,* Vol.129, Part G, pp.47-54, April 1982.

[6] P. Rajan and M. N. S. Swamy, "Quadrantal Symmetry Associated with two-dimensional Digital Transfer Functions", *IEEE Trans. on Circuits Syst*., Vol. 29, pp.340-343, June 1983.

[7] T. Laasko and S. Ovaska, "Design and Implementation of Efficient IIR Notch Filters with Quantization Error Feedback", *IEEE Trans. on Instrumentation and Measurement,* Vol43, No3, pp449-456, June !994

[8] C.-H. Hsieh, C.-M. Kuo, Y.-D. Jou and Y.-L. Han, "Design of two-dimensional FIR Digital Filters by a two-dimensional WLS Technique*", IEEE Trans. on Circuits and Systems - Part II* Vol.44. No5, pp.348-412, May 1997.

[9] M. Daniel and A. Willsky, "Efficient Implementations of 2-D non-Causal IIR Filters", *IEEE Trans. on Circuits and Systems - Part II,* Vol.44, No7, pp549-563, July 1997.

[10] W.-P. Zhu, M. Alhmad and M. N. S. Swamy, "A Closed-Form Solution to the Least-Square Design Problem of 2-D Linear- Phase FIR Filters" *IEEE Trans. on Circuits and Systems - Part II,* Vol44, No12, pp.1032-1039,December 1997.

[11] A. Cichocki, R. Unbehauen, *Neural Networks for Optimization and Signal Processing*, John Wiley & Sons, Chichester-New York-Brisbane-Toronto-Singapore, 1993.

[12] L. O. Chua and G. N. Lin. “Non-linear programming without computation”, *IEEE Trans. Circuits and Systems*, CAS-31, pp.182-186, 1984.

[13] D. W. Tank and J. J. Hopfield. “Simple “neural” optimization networks: an A/D converter, signal decision circuit, and a linear programming circuit”, *IEEE Trans. Circuits and Systems*, CAS-33, pp.533-541, 1986.

[14] M. J. Smith *and* C. L. Portmann. “Practical design and analysis of a simple “neural” optimization circuit”, *IEEE Trans. Circuit and Systems*, Vol.36, pp.42-50, 1989.

[15] M. P. Kennedy and L. O. Chua. “Neural networks for non-linear programming”, *IEEE Trans. Circuit and* Systems, Vol. 35, pp.554-562, 1988.

[16] W. E. Lillo, M. H. Loh, S. Hui and S. H. Ž ak. “On solving constrained optimization problems with neural networks : a penalty method approach”, *Technical Report TR EE 91-43*, School of EE, Purdue University, West Lafayette, IN, 1991.

[17] W. E. Lillo, S. Hui, S. Hui and S. H. Žak. “Neural network for constrained optimization problems”, *International Journal of Circuit Theory and Applications*, vol. 21, pp.385-399, 1993.

[18] V.M. Mladenov, P.N. Proshkov, "Modelling and Simulation of Continuous Neural Networks for Constrained Optimization Problems", *2nd IMACS International Conference on: Circuits, Systems and* Computers, *(CSC'98)*, Greece, published in "Recent Advances in Information Science and Technology", (Editor Nikos E. Mastorakis), World Scientific, ISBN 981-02-3657-3, 386-393, 1998.