On the General Design Problem of 2-Dimensional Recursive Filters by using Neural Networks

N. E. MASTORAKIS*, V. M. MLADENOV**

(*)Military Institutions of University Education, Hellenic Naval Academy, Terma Hatzikyriakou, 18539, Piraeus, GREECE

(**) MSM group, Department of Electrical Engineering, Eindhoven University of Technology, the Netherlands

Abstract: - In this paper, a new design method for two-dimensional (2-D) recursive digital filters using Neural Networks is proposed. 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. Here, the method given in the previous authors work [29] has been substantially improved by introducing a better approximation function. Numerical example is given as well.

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

1 Introduction

Recently many methods for the design of 2-D (recursive or non-recursive) discrete signal, linear and shift invariant filters have been proposed [3] due to a variety of applications in fields as digital image 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. Unfortunately, most of the existing algorithms [3]-[10] may result in an unstable filter. Regarding the stability, the previous methods, are based on optimization techniques and they are more or less trial-and-error approaches and cannot always guarantee the stability of the 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 [3, page 234].

In a recent paper [29], an attempt to adopt an optimization procedure by using a continuous-time Neural Network (NN) has been made. The desired stability of 2-D filter yields our appropriate constraints for the minimization problem. In the present paper, the same problem is solved using however, a more general approximation function.

In [29], the authors have used as approximation function a transfer function with a denominator that is product of first-order 2-D polynomials. The stability of the factors of this products guarantees the stability of the designed filter. Fortunately, for the stability of first-order 2-D polynomials, simple stability conditions exist yielding simple inequality constraints. These inequality constraints can easily be implemented for our minimization problem by NN.

In this paper, we use as approximation function a transfer function having as a denominator a product of second-order 2-D polynomials. The stability of the factors of this product ensures the stability of the designed filter. So, we have more general transfer function for our approximation.

In this case, stability conditions also exist, being however more complicated than the corresponding stability conditions of the first-order 2-D polynomials in [29].

As a result the utilized NN will be more complex than the corresponding in [29], giving, however, better approximation results. The design can not be extended for higher (than second) order 2-D polynomials, since the stability analysis for example of a n-th order 2-D polynomial (n-th degree in both
variables) is reduced, after algebraic manipulation to the study of roots of a $2n$-th degree one-variable polynomial. It is well known that Algebra provides formulas for the roots till 4th degree one-variable polynomial. As a result, the maximum value of $n$ is 2.

Artificial Neural Networks or simply Neural Networks (NN) have already been used to obtain solution of constrained optimization problems [11-16] (Non-linear programming circuits, using the Kuhn-Tucker conditions from the mathematical programming theory, optimization network for solving linear programming problems).

In [17], the authors have used 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 used the MATLAB with SIMULINK software package for modeling and simulations of its behavior.

Several NN architectures are used for the hardware implementation of the proposed NN [19-25]. Surveys can be found in recent papers. Many recent papers dedicated on the implementation of NN using VLSI technology or optoelectronic devices have also been published [25-28].

The paper is organized as follows. In the next section, the problem for design of 2-D recursive filters using approximation function with second-order 2-D polynomials in the denominator is considered. In section 3, a Neural Network for solving the design problem is presented. In section 4, an example for testing the utilized Neural Network is given. Concluding remarks are finally given in section 5.

## 2 Problem Formulation

In the design of a two-dimensional filter, the two-dimensional $z$-transformation with positive powers, i.e. for a two-dimensional sequence $f(n_1,n_2)$ ($f(n_1,n_2) = 0$ for $n_1 < 0$ or $n_1 < 0$) is given by

$$F(z_1,z_2) = \sum_{n_1=0}^{\infty} \sum_{n_2=0}^{\infty} f(n_1,n_2) z_1^{n_1} z_2^{n_2}$$

This is the standard definition in literature of the stability and stability margin of 2-D systems. In this paper, we consider $M_d$, the desirable amplitude response of a 2-D filter as a function of the frequencies $\omega_1$, $\omega_2$, ($\omega_1$, $\omega_2$ $\in [0, \pi]$). The design task at hand amounts to finding a transfer function $H(z_1,z_2)$ such that the function $H(\omega_1,\omega_2)$ approximates the desired amplitude response $M_d$ ($\omega_1$, $\omega_2$). For the design purposes we consider that

$$M_1(\omega_1,\omega_2) = H(z_1,z_2)|_{z_1=e^{j\omega_1}, z_2=e^{j\omega_2}}$$

$$\omega_1 = \frac{\pi}{N_1} n_1, \quad \omega_2 = \frac{\pi}{N_2} n_2$$

and $p$ is an even positive integer (usually $p=2$ or $p=4$). This is equivalent to minimize

$$J = \sum_{n_1=0}^{N_1} \sum_{n_2=0}^{N_2} \left[ M_1(\pi_1,\pi_2) - M_d(\pi_1,\pi_2) \right]^p$$

where
Consider now the quantities [31]
\[ G = p_1^3 - 4p_1p_2 + 8p_3 \]
\[ H = 8p_2^3 - 3p_2^2 \]
\[ I = p_2^3 + 12p_2 - 3p_2p_3 \]
\[ F = 16p_2^3 + 3p_1^3 - 16p_2^2p_1^2 - 64p_4 + 16p_1p_3 \]
\[ J = 72p_2p_4 + 9p_1p_2p_3 - 2p_3^2 - 27p_3^2 - 27p_4p_1^2 \]
\[ \Delta = 4I^3 - J^2 \]

It has been proved [31] that (5) holds if and only if
\( (0 \geq H \text{ and } 0 > \Delta) \) or \( (0 < H \text{ and } 0 < F \text{ and } 0 > \Delta) \) or \( (0 > H \text{ and } 0 = G \text{ and } 0 = F) \)

or
\[ p_1 > 0 \text{ and } p_2 > 0 \text{ and } p_3 > 0 \text{ and } p_4 > 0 \text{ and } (H < 0 \text{ and } F > 0 \text{ and } \Delta > 0) \text{ or } (H \geq 0 \text{ and } G = 0 \text{ and } F = 0) \]

\[ H(z_1, z_2) = H_0 \frac{a_{00} + a_{01}z_2 + a_{02}z_2^2 + a_{10}z_1 + a_{20}z_1z_2 + a_{21}z_1^2 + a_{11}z_1z_2 + a_{22}z_1z_2^2 + a_{21}z_1^2 + a_{22}z_1z_2^2}{(b_{00} + b_{01}z_2 + b_{02}z_2^2)(b_{10} + b_{11}z_2 + b_{12}z_2^2) + z_1 + (b_{20} + b_{21}z_2 + b_{22}z_2^2)z_1^2} \]

Instead of using the conditions proved in [31] and described in the previous section we introduce an extra variable \( y \). In our case we have inequality constraint only and because of this we utilize the Neural Network from [11] (page 369), [29] depicted in Figure 1. The vector of unknown variables is
\[ x = (a_{01}, a_{02}, a_{10}, a_{11}, a_{12}, a_{20}, a_{21}, a_{22}, b_{00}, b_{01}, b_{02}, b_{10}, b_{11}, b_{12}, b_{20}, b_{21}, b_{22}, H_0, y)^T \]

Thus the design of 2-D recursive filters is equivalent of the following constrained minimization problem
\[ \min J \text{ where } J \text{ is given by equation (5) subject to equation (9) for every second order polynomial factor } \sum_{i=1}^{2} \sum_{j=1}^{2} b_{ij}z_1^iz_2^j \text{ of the denominator of (1)} \]

The convergence for continuous differentiable functions of the proposed net has been proved in [11] and the transient has been in the frame of the time constants i.e. several ms. This NN is completely stable in the sense that it will never oscillate or display other exotic modes of operation [11].

\section{Simulation Example for testing the Neural Networks}

We consider the design of 2-D recursive filter (1) and use the desired amplitude response
\[ M_d(\omega_1, \omega_2) = \begin{cases} 
1 & \text{if } \sqrt{\omega_1^2 + \omega_2^2} < 0.08\pi \\
0.5 & \text{if } 0.08\pi < \sqrt{\omega_1^2 + \omega_2^2} < 0.12\pi \\
0 & \text{otherwise} 
\end{cases} \]

\[ p_4 > 0 \text{ and } \{ (p_1 = 0 \text{ and } p_3 < 0) \text{ or } (p_1 > 0 \text{ and } p_3 > 0) \text{ or } (p_1, p_3 < 0 \text{ and } G > 0) \} \text{ and } \{ \Delta < 0 \text{ or } (H \leq 0 \text{ and } F < 0 \text{ and } \Delta = 0) \text{ or } (H > 0 \text{ and } (G = 0 \text{ and } F > 0) \text{ or } G \neq 0 \text{ and } \Delta = 0) \} \]

\[ \text{We chose } k=1, \ p=2, \ N_1=50 \text{ and } N_2=50 \text{ and the corresponding constrained optimization problem becomes:} \]
\[ \min J = \sum_{n_1=0}^{50} \sum_{n_2=0}^{50} \left[ M \left( \frac{\pi}{50} n_1, \frac{\pi}{50} n_2 \right) - M_d \left( \frac{\pi}{50} n_1, \frac{\pi}{50} n_2 \right) \right]^2 \]

subject to
\[ g(y) > 0 \]

where \( x = (a_{01}, a_{02}, a_{10}, a_{11}, a_{12}, a_{20}, a_{21}, a_{22}, b_{00}, b_{01}, b_{02}, b_{10}, b_{11}, b_{12}, b_{20}, b_{21}, b_{22}, H_0, y)^T \) is the vector of unknown variables.

The corresponding Neural Network from Figure 1 has been realized and simulated in MATLAB using SIMULINK environment. It has been tested for different initial conditions. It should be noted that because of the high computational speed of the network the solution has been obtained in the frame of time constants i.e. several ms and a lot of examples have been performed. The satisfactory one, obtained after the transient of the Neural Network is:

Therefore:
The corresponding amplitude response $|M(\omega_1, \omega_2)|$ of the designed 2-D filter is given in Fig. 2, while in Fig. 3 one can obtain the amplitude response of the desired ideal 2-D filter $|M_d(\omega_1, \omega_2)|$. 

$$H(z_1, z_2) = -0.0037 + 2.3928z_2 + 8.6029z_2^2 + 1.3793z_1 + 6.5954z_1^2 + 7.3471z_1z_2 + 2.5799z_1z_2^2 + 1.1099z_1^2z_2^2 + 4.7493z_1^2z_2^4 - 3.7124 + 0.5453z_1^2 + 4.2065z_2^2 + 4.9519 + 4.1532z_1 + 11.2284z_1z_2 + 3.2818 + 0.3162z_1 + 4.6308z_2^2$$
5 Conclusions
In this paper, a new design method for two-dimensional (2-D) recursive digital filters using Neural Networks is proposed. The design of the 2-D filter is reduced to a constrained minimization problem the solution of which is achieved by the convergence of the utilized Neural Network. Here, we substantially improve, the method appeared in [29] by introducing a better approximation function. Numerical example is given as well.

References: