2 EGECIOGLU, Ö., and KOÇ, K.: 'Exponentiation using canonical recoding', *Theoretical Computer Science*, 1994, **129**, (2), pp. 407–417
3 KAI-HWANG: 'Computer arithmetic, principles, architecture, and design' (John Wiley & Sons, New York, 1989)
4 KOREN, U.: 'Computer arithmetic algorithms' (Prentice-Hall, Englewood Cliffs, NJ, 1993)
5 LADNER, R., and FISCHER, M.: 'Parallel prefix computation', *J. ACM*, 1980, **27**, (4), pp. 831–838
6 REITWIESNER, G.W.: 'Binary arithmetic', *Advances in Computers*, 1960, **1**, pp. 231–308

2D multiple notch filter design

Chien-Cheng Tseng and Soo-Chang Pei

Indexing terms: Notch filters, Two-dimensional digital filters

The authors propose a decomposition method to reduce the 2D multiple notch filter design problem to two pairs of 1D filter design problems. They develop a simple algebraic method for the design of two pairs of 1D IIR filter design. This approach not only has closed form transfer function but also satisfies the bounded input/output (BIBO) stability condition.

Introduction: Notch filters are an effective means for eliminating narrowband or sinusoidal interferences in certain signal processing applications ranging from power line interference cancelation for electrocardiograms to multiple sinusoidal interference removal for corrupted images. For the 1D case, several methods for the design and performance analysis of IIR and FIR notch filters have been developed, see [1 – 3] among others. For the 2D case, [4] proposed a method that reduces the design of a 2D single notch filter to the design of a 2D parallel line filter and a 2D straight line filter. However, a technique for designing a 2D multiple notch filter has not yet been developed. In this Letter, we address this design problem.

Design technique: The frequency response of an ideal 2D multiple notch filter is given by

$$H_d(e^{j\omega_1}, e^{j\omega_2}) = \begin{cases} 0 & (\omega_1, \omega_2) = (\omega_{1k}^*, \omega_{2k}^*) \text{ and } \\ & (-\omega_{1k}^*, -\omega_{2k}^*) \quad k = 1, ..., N \\ 1 & \text{otherwise} \end{cases} \tag{1}$$

where $(\omega_{1k}^*, \omega_{2k}^*)$ are the notch frequencies. The aim of this work is to find a stable 2D transfer function which satisfies this specification. Our design technique is mainly based on the following decomposition of the frequency response of an ideal notch filter:

Fact 1: Given that two 1D filters $H_{ci}^k(z_i)$ and $H_{si}^k(z_i)$ have the following frequency responses ($i = 1, 2$):

$$H_{ci}^k(e^{j\omega_i}) = \begin{cases} 1 & \omega_i = \omega_{ik}^* \text{ and } -\omega_{1k}^* \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

$$H_{si}^k(e^{j\omega_i}) = \begin{cases} -j & \omega_i = \omega_{ik}^* \\ j & \omega_i = -\omega_{ik}^* \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

then the frequency response of an ideal 2D multiple notch filter can be written as

$$H_d(e^{j\omega_1}, e^{j\omega_2}) = 1 - \sum_{k=1}^{N} \left[\frac{1}{2} H_{c1}^k(e^{j\omega_1}) H_{c2}^k(e^{j\omega_2}) - \frac{1}{2} H_{s1}^k(e^{j\omega_1}) H_{s2}^k(e^{j\omega_2}) \right] \tag{4}$$

From Fact 1, we see that the design of a 2D multiple notch filter can be decomposed into two types of 1D filter design. The first is the design of filter $H_{ci}^k(e^{j\omega_i})$ defined in eqn. 2, the other is the design of filter $H_{si}^k(e^{j\omega_i})$ defined in eqn. 3 (i = 1, 2). We shall address the design of these two types of filters.

(i) *Design of filter* $H_{ci}^k(z_i)$: The frequency response of $H_{ci}^k(e^{j\omega_i})$ can be approximated by the second-order IIR bandpass filter whose transfer function is given by

$$H_{bi}^k(z_i) = \frac{1}{2}\left(1 - \frac{a_{ik2} - a_{ik1}z_i^{-1} + z_i^{-2}}{1 - a_{ik1}z_i^{-1} + a_{ik2}z_i^{-2}}\right) \quad i = 1, 2 \tag{5}$$

where, from the results in [4], the coefficients a_{ik1} and a_{ik2} are given by

$$a_{ik1} = \frac{2\cos(\omega_{ik}^*)}{1 + \tan(\frac{BW}{2})}$$
$$a_{ik2} = \frac{1 - \tan(\frac{BW}{2})}{1 + \tan(\frac{BW}{2})} \tag{6}$$

with ω_{ik}^* is the centre frequency of $H_{bi}^k(z_i)$ and BW is the 3 dB bandwidth of $H_{bi}^k(z_i)$. Note that $H_{bi}^k(e^{j\omega^*_{ik}})$ is exactly equal to unity, i.e. $H_{bi}^k(e^{j\omega_{ik}})$ has unit gain and zero phase at $\omega_i = \omega_{ik}^*$. Thus, $H_{bi}^k(e^{j\omega_i})$ will be an excellent approximation of $H_{ci}^k(e^{j\omega_i})$ provided that BW is sufficiently small.

(ii) *Design of filter* $H_{si}^k(z_i)$: It can readily be verified that the filter $H_{si}^k(z_i)$ can be obtained as $H_{si}^k(z_i) = H_{ci}^k(z_i)H_{ai}^k(z_i)$ where the frequency response of $H_{ai}^k(z_i)$ is given by

$$H_{ai}^k(e^{j\omega_i}) = \begin{cases} -j & \omega_i = \omega_{ik}^* \\ j & \omega_i = -\omega_{ik}^* \\ \text{don't care} & \text{otherwise} \end{cases} \tag{7}$$

Since $H_{ci}^k(z_i)$ has been designed in the preceding subsection, we now only need to design filter $H_{ai}^k(z_i)$ For simplicity, we choose $H_{ai}^k(z_i)$ to be the following first order allpass filter:

$$H_{ai}^k(z_i) = \frac{b_{ik} + z_i^{-1}}{1 + b_{ik}z_i^{-1}} \qquad i = 1, 2 \tag{8}$$

Since $|H_{ai}^k(e^{j\omega_i})|$ is equal to unity for all frequencies, we have $H_{ai}^k(e^{j\omega_i}) = e^{j\theta_{ik}(\omega_i)}$ where the phase response $\theta_{ik}(\omega_i)$ is given by

$$\theta_{ik}(\omega_i) = -\omega_i + 2\arctan\left(\frac{b_{ik}\sin(\omega_i)}{1 + b_{ik}\cos(\omega_i)}\right) \tag{9}$$

Also, the specification in eqn. 7 implies that

$$\theta_{ik}(\omega_{ik}^*) = -\frac{\pi}{2} \tag{10}$$

Substituting eqn. 10 into eqn. 9, we obtain

$$b_{ik} = \frac{\sin(\frac{\omega_{ik}^*}{2} - \frac{\pi}{4})}{\sin(\frac{\omega_{ik}^*}{2} + \frac{\pi}{4})} \tag{11}$$

Based on the above discussion, a complete procedure for the design of a 2D multiple notch filter can be summarised as follows:
(i) Step 1: Specify notch frequencies $(\omega_{1k}^*, \omega_{2k}^*)$ and bandwidth BW for $k = 1 ... N$.
(ii) Step 2: Use eqn. 6 to compute filter coefficients a_{ik1}, a_{ik2}, and construct transfer function $H_{bi}^k(z_i)$.
(iii) Step 3: Use eqn. 11 to calculate coefficients b_{ik}, and construct transfer function $H_{ai}^k(z_i)$
(iv) Step 4: Form the transfer function of the 2D multiple notch filter as

$$H(z_1, z_2) = 1 - \sum_{k=1}^{N} \left[\frac{1}{2} H_{b1}^k(z_1) H_{b2}^k(z_2) \left(1 - H_{a1}^k(z_1) H_{a2}^k(z_2)\right)\right] \tag{12}$$

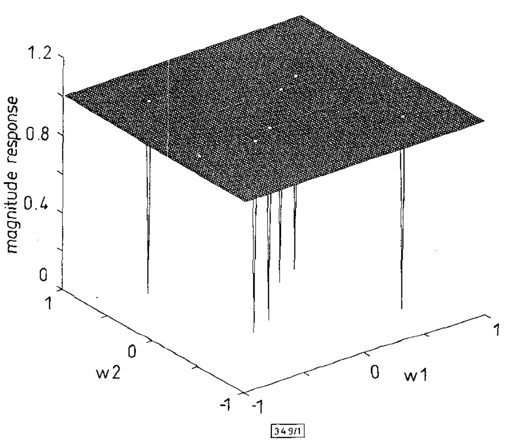

Fig. 1 *Magnitude response of designed 2D multiple notch filter*

Design example: In this example, three notch frequencies and bandwidth BW are chosen as $(\omega_{11}^*, \omega_{21}^*) = (0.2\pi, 0.2\pi)$, $(\omega_{12}^*, \omega_{22}^*) = (0.3\pi, 0.4\pi)$, $(\omega_{13}^*, \omega_{23}^*) = (-0.6\pi, 0.6\pi)$, $BW = 0.001\pi$. Fig. 1 shows the magnitude response of the designed multiple notch filter in linear scale. It is clear that the specification is well satisfied. In fact, when the bandwidth BW approaches zero, the designed multiple notch filter will become an ideal one.

Conclusions: In this Letter, a 2D multiple notch filter design problem has been investigated. First, we reduce the 2D notch filter design problem to two pairs of 1D filter design problems. Secondly, we provide the closed-form solutions for the design of two pairs of 1D IIR filters.

8 August 1996
Electronics Letters Online No: 19961385

Chien-Cheng Tseng (*Applied Research Laboratory, Telecommunication Laboratories, Chunghua Telecom Co., Ltd., Yang Mei, Taiwan, Republic of China*)

Soo-Chang Pei (*Department of Electrical Engineering, National Taiwan University, Taipei, Taiwan, Republic of China*)

References

1 ER, M.H.: 'Designing notch filter with controlled null width', *Signal Process.*, 1991, **24**, (3), pp. 319–329

2 MEDLIN, G.W.: 'A novel design technique for tuneable notch filters'. Proc. Int. Symp. on Circuits and Systems, New Orleans, Louisiana, USA, 1990, **1**, pp. 471–474

3 DUTTA ROY, S.C., JAIN, S.B., and KUMAR, B.: 'Design of digital FIR notch filters', *IEEE Proc.-Vis. Image Signal Process.*, 1994, **141**, (5), pp. 334–338

4 SOO-CHANG PEI, and CHIEN-CHENG TSENG: 'Two dimensional IIR digital notch filter design', *IEEE Trans. Circuits and Syst.-II*, 1994, **CAS-41**, (5), pp. 227–231

Accurate parallel form filter synthesis

M. Price, S. Holden and M. Sandler

Indexing terms: Digital filters, Network synthesis

Standard algorithms for synthesising parallel form digital filters are adequate for basic applications. However, recursive filters required in acoustics synthesis applications often have very high orders, and errors from filter synthesis can become intolerable. The authors describe a new method for synthesising parallel form recursive filters. In the experiment described, the new algorithm yielded smaller errors than the standard technique for almost every filter tested.

Introduction: In acoustics synthesis applications, the recursive synthesis filter transfer function is usually specified in its direct form as the rational function, defined in eqn. 1:

$$H(z) = \frac{U}{Q}(z) \quad (1)$$

where $Q(z)$ and $U(z)$ are assumed to be known, with degree[$Q(z)$] = N and degree [$U(z)$] = M.

However, it is useful to implement the synthesis filter with a parallel form, so that each resonance can be independently controlled [2]. The filter is converted into a parallel form by decomposing the transfer function into a sum of rational functions, as defined in eqn. 2:

$$H(z) = S_0(z) + \sum_{i=1}^{\beta} \frac{P_i(z)}{(1 - x_i z^{-1})(1 - x_i^* z^{-1})} = \sum_{i=0}^{\beta} S_i'(z) \quad (2)$$

where β is the sum of the number of real poles and the number of complex pairs of poles, and $x_i^* = 0$ when x_i is real. This is usually achieved by computing the partial fraction decomposition, and recombining the terms that involve complex conjugate pairs of poles.

The standard numerical algorithm for computing a partial fraction decomposition is known as 'direct determination of principal-parts' (DDPP) [1]. Despite its relatively low computational complexity, we have found that the DDPP method yields large errors when synthesising high order filters ($N \geq 8$), such as those required in acoustics synthesis applications.

In this Letter, we define a new parallel form synthesis algorithm, and compare it with the DDPP method. The comparison is made using Monte-Carlo techniques.

New parallel form synthesis algorithm: This algorithm is a generalisation of the 'undetermined coefficients' partial fraction expansion algorithm [1]. The generalisation enables us to obtain the parallel form representation directly, avoiding the use of complex arithmetic.

We begin with $H(z)$, which is to be transformed into the parallel form representation of eqn. 2. We assume that $M < N$ and hence $S_0(z) = 0$. From eqn. 2, we obtain the identity

$$U(z) = \sum_{j=1}^{\beta} P_j \Phi_j(z) = \sum_{i=0}^{M} u_i z^{-i} \quad (3)$$

where $\Phi_j(z)$ is defined as

$$\Phi_j(z) = \frac{Q}{Q_j}(z) = \sum_{i=0}^{N-\rho_j} \phi_{j,i} z^{-i} \quad (4)$$

with $Q_j(z) = (1 - x_j z^{-1})(1 - x_j^* z^{-1})$, for $j = 1, 2, ..., \beta$, and where ρ_j = degree[$Q_j(z)$]. The pole locations x_j and x_j^* are assumed to be known.

Equating coefficients of eqn. 3 for powers in z, yields a 2β order, non-singular system of simultaneous equations. In matrix form, it is

$$\mathbf{\Phi} \cdot \underline{\mathbf{p}} = \underline{\mathbf{u}} \quad (5)$$

where

$$\underline{\mathbf{u}}^{\mathbf{T}} = [u_0 \quad u_1 \quad \cdots \quad u_{(2\beta)-1}] \quad (6)$$

with $u_i = 0$ for $i > M$,

$$\underline{\mathbf{p}}^{\mathbf{T}} = [p_{1,0} \quad p_{2,0} \cdots \quad p_{\beta,0} \quad p_{1,1} \quad p_{2,1} \cdots \quad p_{\beta,1}] \quad (7)$$

with $P_{j,1} = 0$ when $\rho_j = 1$ and for $j = 1, 2, ..., \beta$, and

$$\mathbf{\Phi} = \begin{bmatrix} \phi_{1,0} & \phi_{2,0} \cdots \phi_{\beta,0} & 0 & 0 & \cdots & & \cdots & 0 \\ \phi_{1,1} & \phi_{2,1} \cdots \phi_{\beta,1} & \phi_{1,0} & \cdots & \phi_{\beta,0} & 0 \; 0 & \cdots & 0 \\ \vdots & & & & & & & \vdots \\ \phi_{1,(2\beta)-2} & \cdots & & & & & \cdots & \phi_{\beta,0} \\ \phi_{1,(2\beta)-1} & \cdots & & & & & \cdots & \phi_{\beta,1} \end{bmatrix} \quad (8)$$

with $\phi_{j,i} = 0$ for $i > N - \rho_j - 1$ and for $j = 1, 2, ..., \beta$.

The coefficients of the subfilter numerator polynomials $P_j(z)$ are therefore obtained by solving eqn. 5 for $\underline{\mathbf{p}}$.

Monte-Carlo tests: Monte-Carlo methods were applied to the problem of quantifying the average error incurred from a synthesis process as follows. 12 sets of 100 all-pole test filters, each set for a specific filter order ranging from 8th to 96th order, were generated. The filters were generated by randomly placing the required number of poles within the unit circle, with a uniform distribution. The test proceeded by computing the parallel form realisation of each test filter, using each of the synthesis methods. The errors in the resultant realisations were then measured as follows.

Fig. 1 *Filter simulation for Monte-Carlo tests*

Each synthesised test filter was simulated as illustrated in Fig. 1, using double precision floating-point arithmetic. The errors in the frequency responses of the test filters were then measured, by comparing with the frequency responses of the cascade form equivalents, in terms of average spectral deviation, e_j. This is defined in eqn. 9: