ANALYTICAL METHODS FOR THE DESIGN OF 2-D ELLIPTICALLY
SYMMETRIC DIGITAL FILTERS OF ARBITRARY ORIENTATION
USING GENERALIZED McCLELLAN TRANSFORMATION

N. Nagamuthu and M.N.S. Swamy
Concordia University

I. INTRODUCTION

The current research interests in the area of two-dimensional (2-D) digital filters are
towards the development of efficient methods for the design and implementation of the filters.
In applications like automatic quality control, computer vision, and robotics there is need for
real-time image processing due to large volume of data. The 2-D image signal, to be processed,
may have a spatial variation along a certain direction and it may not coincide with the spatial
coordinates along which the sampling is performed. To process such signals, 2-D elliptically
symmetric digital filters of arbitrary orientation are employed.

A simple and fast method of designing 2-D zero-phase finite/infinite impulse response
(FIR/IIR) filters is to transform one-dimensional (1-D) zero-phase finite/infinite impulse
response filter by using McClellan transformation [1]. This method is capable for designing 2-D
zero-phase filters of even higher order in a modest amount of computer time, and there are
efficient implementations for the filters [2]. Since the 2-D zero-phase IIR filter obtained by using
the transformation is unstable, it is spectrally factorized [3] into 4 quarter-plane [4, 5] or 2
half-plane [6] filters which are stable.

The original McClellan transformation [1] has only cosine terms; therefore, it is only
suitable for designing 2-D quadrantally symmetric filters [7]. When a horizontal or vertical
elliptical cutoff contour is rotated, it no longer possesses quadrantal symmetry, but possesses
only centro-symmetry [8]. So, in order to have a better approximation for this type of contours,
we should use the generalized McClellan transformation which has sine terms also [7, 9]. It
is given by

\[
\cos(\omega) = t_{00} + t_{01}\cos(\omega_1) + t_{11}\cos(\omega_1)\cos(\omega_2) + s_{11}\sin(\omega_1)\sin(\omega_2)
\]

\[
= F(\omega_1, \omega_2)
\]

For \(0 \leq \omega \leq \pi\), the left hand side of (1) i.e. \(\cos(\omega)\) varies from 1 to -1. So, the right hand side
of (1) should satisfy the constraint

\[
| F(\omega_1, \omega_2) | \leq 1, \quad 0 \leq \omega_1 \leq \pi, \quad 0 \leq \omega_2 \leq \pi.
\]

The McClellan transformation coefficients \(t_{ij}\)'s, \((i, j) \in (0,1)\) and \(s_{11}\) in (1) are generally
found using an optimization method [10] so that the 1-D pass-band cutoff frequency \(\omega_0\) is
mapped onto the desired 2-D pass-band cutoff contour. If both 1-D and 2-D filters are low-pass,
one of the desirable requirement is that the 1-D origin be mapped onto the 2-D origin i.e.
\(0 \rightarrow (0,0)\). This gives the constraint equation

\[
t_{00} + t_{01} + t_{10} + t_{11} = 1.
\]

Using (3), any one of the \(t_{ij}\)'s can be expressed as a function of the other three. Thus, the total
number of independent coefficients in (1) to be optimized reduces from 5 to 4.

Even though the optimization method [10] gives the best result, it is computationally very expensive. The amount of computation exceeds the arithmetic and/or speed capability of low cost, stand-alone 2-D signal processors and therefore, it is not suitable for real-time applications. Also, the nonlinear optimization method may be unreliable because of the local minima. To overcome these problems, recently, Nguyen and Swamy [11], and Reddy and Hazra [12] have proposed analytical methods to find the coefficients. The method [11] uses second-order sine terms and gives good approximation for lower values of frequency specifications. It is much simpler than the method [12]. A disadvantage of the method is that its scaled coefficients produce two extra pass-bands around (π, π) and (−π, π), in addition to the normal pass-band around (0, 0). The method [12] uses first-order sine terms and gives reasonably good approximation for higher values of frequency specifications as well. It is more complex than the method [11], and it does not give good results for lower values of the ratio of minor-axis (2ω_a) to major-axis (2ω_b). The analytical methods presented here, do not have the disadvantages of [10]-[12], and have the advantages of [11]. First, we consider rotated versions of vertical [i.e. type-(a)] elliptical contour. In Sections II and III, formulas are presented for unscaled and scaling-free coefficients. Then, we consider rotated versions of horizontal [i.e. type-(b)] elliptical contour. The corresponding formulas are given in Sections IV and V. In Section VI, we compare the results of several analytical methods and optimization method. Section VII is the summary.

II. APPROXIMATION FOR COEFFICIENTS FOR TYPE-(A) CONTOURS

Consider a type-(a) elliptical cutoff contour rotated about the 2-D origin by an angle θ in the anticlockwise direction. It is depicted in Fig. 1. The equation of the rotated elliptical contour is

\[ \frac{ω_0^2}{ω_a^2} + \frac{ω_0^2}{ω_b^2} + \frac{ω_0^2}{ω_c} = 1 \]  

where

\[ ω_A^2 = ω_a^2 \left[ 1 - \left( 1 - \frac{ω_0^2}{ω_b^2} \right) \sin^2(θ) \right]^{-1} \]  

\[ ω_B^2 = ω_b^2 \left[ 1 - \left( 1 - \frac{ω_0^2}{ω_a^2} \right) \sin^2(θ) \right]^{-1} \]  

and

\[ ω_C = ω_a ω_b \left[ \frac{ω_b}{ω_a} - \frac{ω_a}{ω_b} \right] \sin(2θ) \]  

To have a better fit for the nonquadrantal contour, and to have a 2-D filter with minimum filter order, we use the first-order generalized McClellan transformation i.e. (1). It should be noted that in (1), only first-order cosine terms and sine term are used to provide the quadrantal symmetric and centro-symmetric component respectively. Fig. 2 shows the 2-D unit impulse response of the transformation in (1).

\[ \cos(x) \] and \[ \sin(x) \] can be approximated by their truncated power series as

\[ \cos(x) \approx 1 - \frac{x^2}{2} + \frac{x^4}{24} \text{ and } \sin(x) \approx x - \frac{x^3}{6} + \frac{x^5}{120}. \]  

Using these in (1), and neglecting sixth- and higher-order powers, we get,

\[ 1 - \frac{ω_0^2}{2} + \frac{ω_0^4}{24} = (t_{00} + t_{01} + t_{10} + t_{11}) - \frac{1}{2} (p_1 ω_1^2 + p_2 ω_2^2) \]  

