Blekinge Institute of Technology
Department of Telecommunications and Signal Processing
Sven Nordebo[1]

November 2003
V.2003.1

Multidimensional Signal Processing ETD003

Laboratory Experiment 2: 2–D FIR Filter Design

1 Filter Design in Matlab

In this experiment two-dimensional FIR filters are going to be designed by using the window method and the frequency transformation method. Initially, some examples of the design procedure for a 2–D low pass filter, having the following desired circular symmetric frequency response function, will be shown

$$H_d(\omega_1, \omega_2) = \begin{cases} 1, & \sqrt{\omega_1^2 + \omega_2^2} \leq 0.4\pi \\ 0, & \text{elsewhere} \end{cases}$$

where $-\pi \leq \omega_1 \leq \pi$ and $-\pi \leq \omega_2 \leq \pi$.

When designing 2–D filters in Matlab, the Toolbox-functions **fwind1** and **fwind2** are suitable. As input parameter to the functions, a matrix $\mathbf{H}_d$, which corresponds to a sampled version of the desired frequency response function $H_d(\omega_1, \omega_2)$, sampled on a grid in the (ω_1, ω_2)–plane. This could be carried out by the following commands which calculates and plots the matrix $\mathbf{H}_d$

```
>> [f1,f2]=freqspace(64);
>> [x,y]=meshgrid(f1,f2);
>> Hd=zeros(size(x));
>> r=sqrt(x.^2 + y.^2);
>> d=find(r<0.4);
>> Hd(d)=ones(size(d));
>> mesh(f1,f2,Hd)
>> xlabel('f1')
>> ylabel('f2')
```

Note that the filter design methods in Matlab uses the normalized frequency (f_1, f_2) where $\omega_1 = \pi f_1$ and $\omega_2 = \pi f_2$, i.e. $-1 \leq f_1 \leq 1$ and $-1 \leq f_2 \leq 1$. To fully comprehend what the commands above produce, exchange the number 64 in the first row to a lower number, 8 for an example, and repeat each matrix-calculation in the Matlab command window.

A two-dimensional FIR filter could now be calculated, according to the windowing method, by using the command

```
>> h=fwind1(Hd,boxcar(11),boxcar(11));
```

where the second and third argument defines a separable 2–D window. The function **boxcar** creates a 1–D rectangular window. Other 1–D windows could of course be used as well, such as the Hamming window

[1] This laboratory experiment tutorial has been modified by Jan-Olof Gustavsson 1999, Benny Lövström 2001 and Benny Sällberg 2003

```
>> h=fwind1(Hd,hamming(11),hamming(11));
```

A filter, designed using circular symmetric windows according to Huang's method, will be returned when only two arguments are used with the method **fwind1**, for example

```
>> h=fwind1(Hd,hamming(11));
```

To plot the two–dimensional frequency response function one could use the Toolbox–function

```
>> freqz2(h)
```

This function plots the frequency response function without applying logarithm functions. If examination of the frequency response function in dB–scale is wanted (which is quite common), a home-made plot function, can be written for example

```
function freqz2d(h,dB)
colormap('default')
db=10^(-dB/20);
h=rot90(fliplr(flipud(h)),-1); % rotates the filter mask to matrix coordinates
H=fft2(h,64,64);
axel=-pi:2*pi/64:pi-2*pi/64;
mesh(axel,axel,20*log10(abs(fftshift(H'))+db))
xlabel('omega1')
ylabel('omega2')
zlabel('magnitude [dB]')
end
```

This function could then be called by using the syntax

```
>> freqz2d(h,40)
```

where 40 renders a 'floor' of -40 dB.

To carry out a filter design by using the transformation method, one could use the function **ftrans2** according to

```
>> h=ftrans2(h1);
```

where **h1** is the one–dimensional FIR filter. As a default, **ftrans2** uses the so called McClellan transformation $T(\omega_1, \omega_2) = -0.5 + 0.5\cos(\omega_1) + 0.5\cos(\omega_2) + 0.5\cos(\omega_1)\cos(\omega_2)$.

To use another transformation sequence **t**, use the command

```
>> h=ftrans2(h1,t);
```

The one–dimensional FIR filter **h1** could be designed by using the function **fir1** as well. With the two–dimensional design example above

```
>> N=11;
>> h1=fir1(N-1,0.4,boxcar(N));
>> h=ftrans2(h1);
```

Note that the function **fir1** operates by using f, which is a normalized frequency, having $\omega = \pi f$, i.e. $-1 \leq f \leq 1$. If one wish to plot the frequency response function of the filter **h1**

```
>> H1=fft(h1,512);
>> plot(0:1/256:1,20*log10(abs(H1(1:257))))
```

2 Tasks

a) Design a 11×11 2–D low pass FIR filter according to the example above. Use the three methods; separable windows, circular symmetric windows and the transformation method. A contour–plot of the McClellan transformation is shown in figure 2a.

Use rectangular windows and hamming windows as 1–D windows for all three methods. The resulting frequency response function will have different properties. Try to explain theses differences. Which method is to prefer?

b) Design a 11×11 2–D high pass FIR filter which approximates the desired frequency response function

$$H_d(\omega_1, \omega_2) = \begin{cases} 1, & \sqrt{\omega_1^2 + \omega_2^2} \geq 0.2\pi \\ 0, & \text{elsewhere} \end{cases}$$

where $-\pi \leq \omega_1 \leq \pi$ and $-\pi \leq \omega_2 \leq \pi$. Use the window method and the frequency transformation method.

c) Design a 11×11 2–D band pass FIR filter which approximates the desired frequency response function

$$H_d(\omega_1, \omega_2) = \begin{cases} 1, & 0.3\pi \leq \sqrt{\omega_1^2 + \omega_2^2} \leq 0.6\pi \\ 0, & \text{elsewhere} \end{cases}$$

where $-\pi \leq \omega_1 \leq \pi$ and $-\pi \leq \omega_2 \leq \pi$. Use the window method and the frequency transformation method.

d) Design a 21×21 2–D FIR filter which approximates the desired frequency response function according to the figure below.

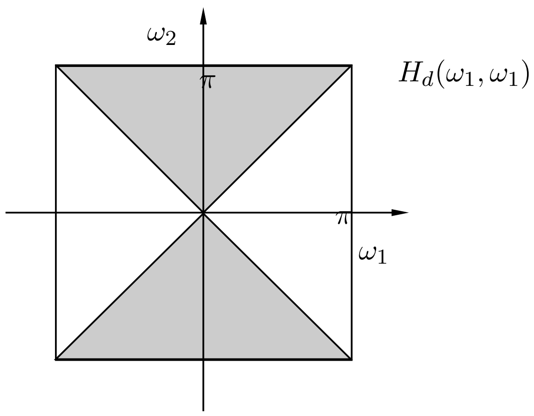


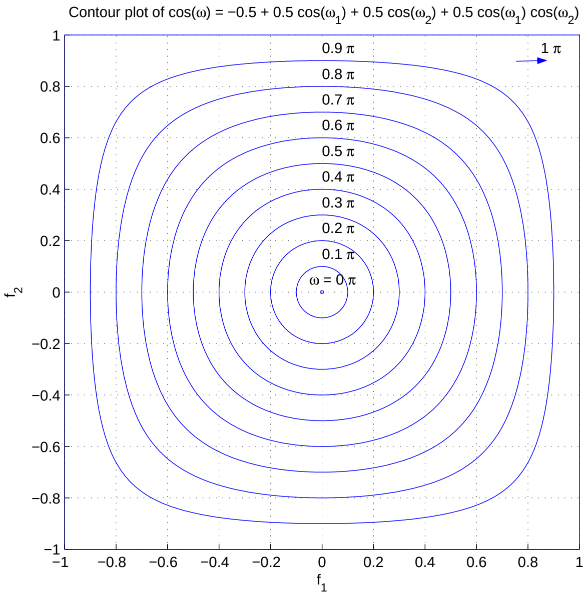
Figure 1: Filter specification.

$$H_d(\omega_1, \omega_2) = \begin{cases} 1, & \text{shadowed area} \\ 0, & \text{clear area} \end{cases}$$

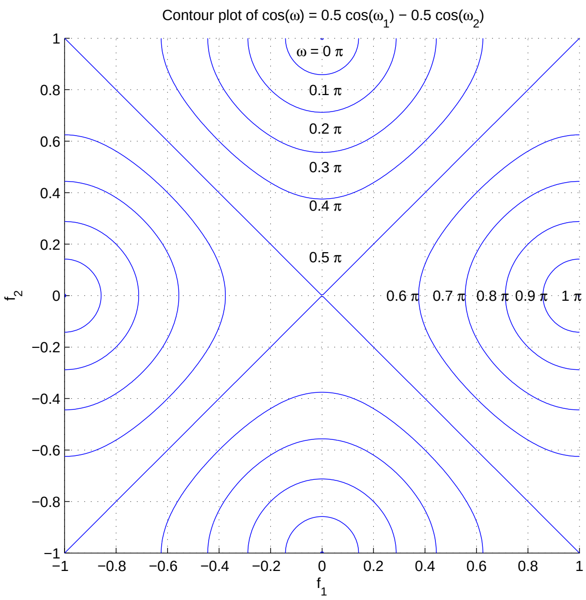
Use the frequency transformation method. A possible transformation sequence has the Fourier Transform.

$$T(\omega_1, \omega_2) = 0.5\cos(\omega_1) - 0.5\cos(\omega_2)$$

A contour–plot of this function is shown in figure 2b.



a)



b)

Figure 2: a) Contour plot of the McClellan transformation for different values of ω. The McClellan transformation is defined as: $\cos(\omega) = \frac{1}{2}(-1 + \cos(\omega_1) + \cos(\omega_2) + \cos(\omega_1) \cdot \cos(\omega_2))$. b) Contour plot of the transform $\cos(\omega) = \frac{1}{2}(\cos(\omega_1) - \cos(\omega_2))$ for different values of ω.