DSB Portfolio 2

Nis Sarup

November 26, 2010

1 Fourier Transform method

Mathematica functions:

$$\begin{aligned} t &= 67; \\ m &= \frac{t-1}{2}; \\ cH &:= \frac{2\pi 1800}{8000}; \\ cL &:= \frac{2\pi 800}{8000}; \\ h[0] &:= \frac{cH - cL}{\pi}; \\ h[n\_] &:= \frac{Sin[cH \cdot n]}{n \cdot \pi} - \frac{Sin[cL \cdot n]}{n \cdot \pi}; \end{aligned} \tag{1}$$

I get the results h[n] with n ranging from $-m$ to m:

$$Table[h[i], i, -m, m, 1]$$

This gives a large table of values which I copy into the MathLab program (Program 7.1 for example 7.3 in the book) and get the following graph:

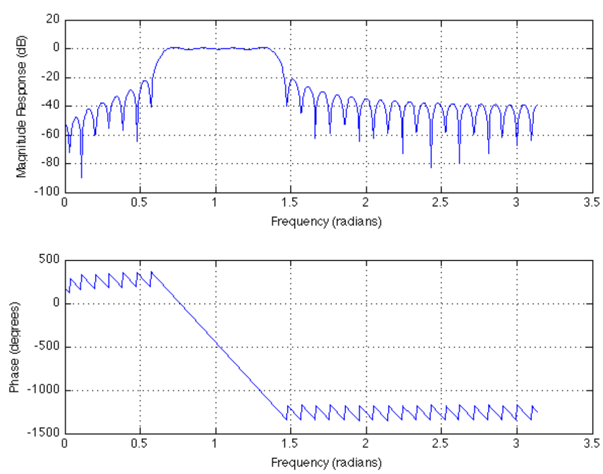

Figure 1: Fourier Transform method: Magnitude response and Phase.

2 Window Method

2.1 Rectangular Window

Mathematica function for the rectangular window:

$$rectangularWindow[n\_] := 1;$$

Not overly exciting. For each value in the set from Section 1 I multiply it with the corresponding value from the window function. Here is a plot of the "new" graphs. Not surprisingly both Figure 1 and Figure 2 looks quite similar.

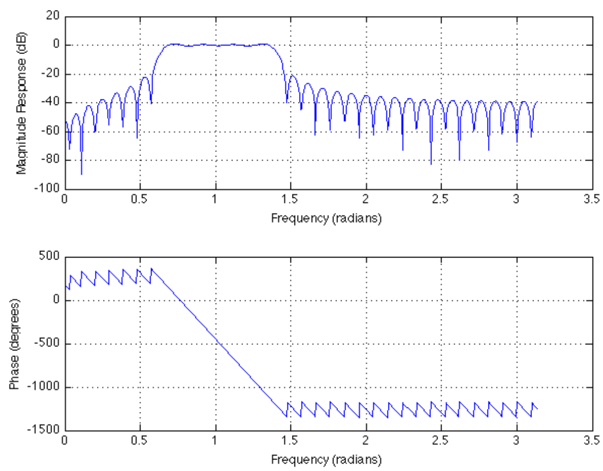

Figure 2: Rectangular window: Magnitude response and Phase.

2.2 Hamming Window

Mathematica function for the Hamming Window:

$$hammingWindow[n\_] := 0.54 + 0.46Cos[\frac{n \cdot \pi}{m}];$$

Same procedure as with the Rectangular window. As can be seen there are some changes on this graph.

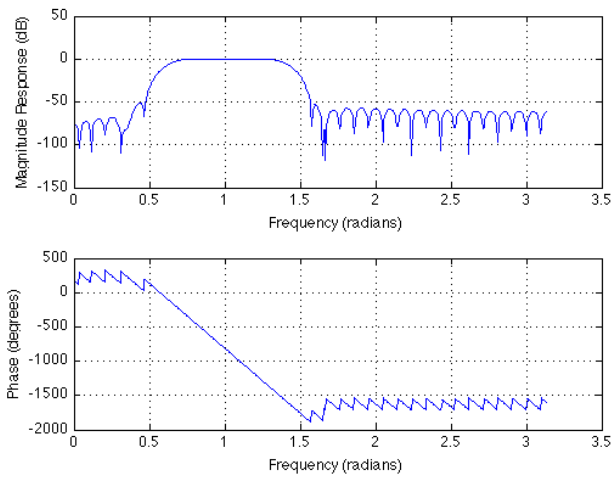

Figure 3: Hamming window: Magnitude response and Phase.

2.3 Kaiser Window

Mathlab has a build-in Kaiser-function, so I use that to generate the Kaiser window-values:

$$\begin{aligned} beta &= 0.5842 * (50-21)^{0.4} + 0.07886 * (50-21); \\ wkaiser &= kaiser(67, beta); \end{aligned} \tag{2}$$

THat particular equation for the Kaiser window is chosen because the specification calls for a stopband attenuation at 50dB. Again the values are multiplied and graphed.

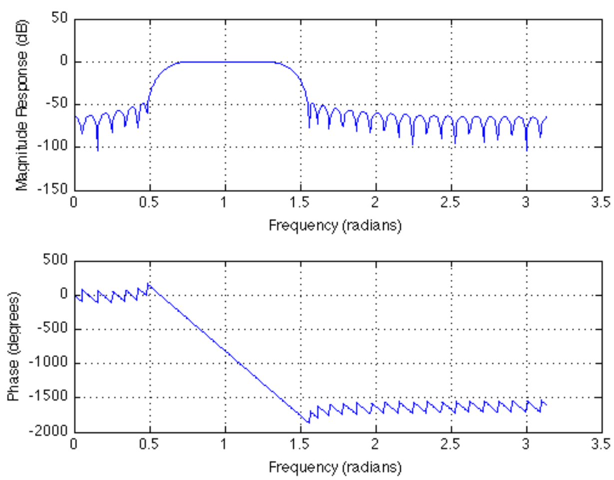

Figure 4: Kaiser window: Magnitude response and Phase.

3 Frequency Sampling Method

3.1 Setup

I change the low cut frequency to 1000Hz and the upper cut frequency to 3000Hz. Here is the relevant mathematica functions (cL and cH is calculated the same way as in Section 1.):

$$\begin{aligned} M &= 12; \\ H[k\_] &:= If[cL < \frac{2\pi k}{2M+1} < cH, 1, 0]; \\ h[n\_] &:= \frac{1}{2M+1}(H[0] + 2\sum_{k=1}^{M}(H[k]Cos[\frac{2\pi k(n-M)}{2M+1}])); \end{aligned} \tag{3}$$

The bandpass frequency response vector is then:

$$\{0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 0, 0, 0\}$$

And with a smooth transition:

$$\{0, 0, 0, 0.5, 1, 1, 1, 1, 1, 1, 0.5, 0, 0\}$$

Here is the plot for a 25 tap frequency sampling filter:

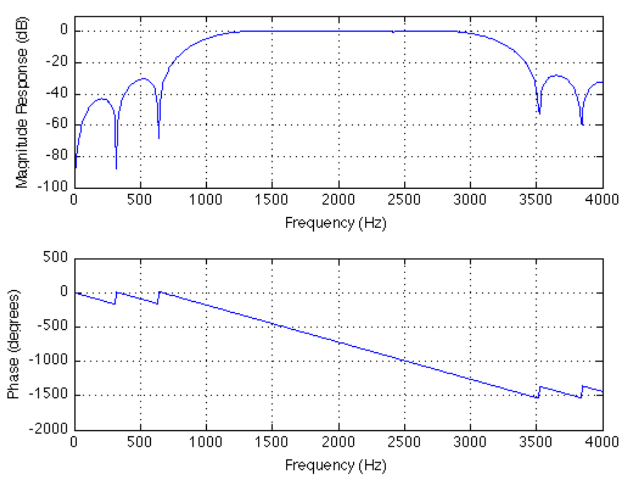

Figure 5: Frequency Sampling methos, with smooth transition.

3.2 Increasing and decreasing the taps

Increasing and decreasing the taps on a filter will have an effect. Here is plots of different tap sizes:

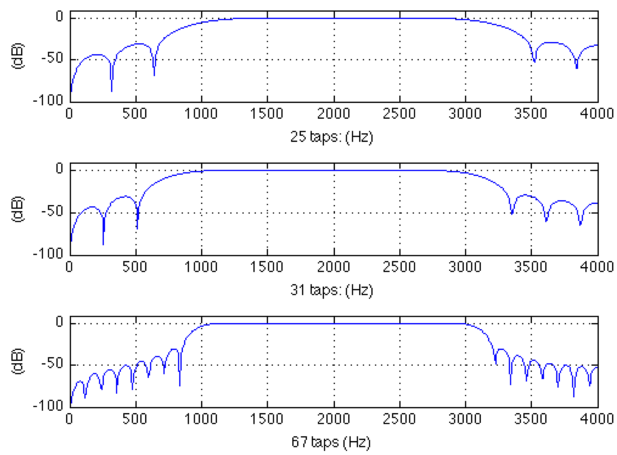

Figure 6: 25, 31 and 67 tap Frequence Sampling filter plots.

3.3 Discussion

The higher tap-number chosen when designing the Frequency Sampling filter, the higher the precision of the filter. The main-lobe follows a straight line better and dips faster at the cut-frequencies.

4 Parks-McClellan algorithm

Variables used:

$$
\begin{aligned}
dP &= 10^{\frac{0.05}{20}} - 1 = 0,00577306 \\
dS &= 10^{\frac{-50}{20}} = 0,00316228 \\
W &= \frac{dP}{dS} = \frac{0,00577306}{0,00316228} \approx \frac{18}{10} \\
WS &= 18 \\
WP &= 10
\end{aligned}
\tag{4}
$$

The only thing that needs changing from the MathLab program 7.16 in the book is error weight factors. The rest is the same. Here is the plot of the filter:

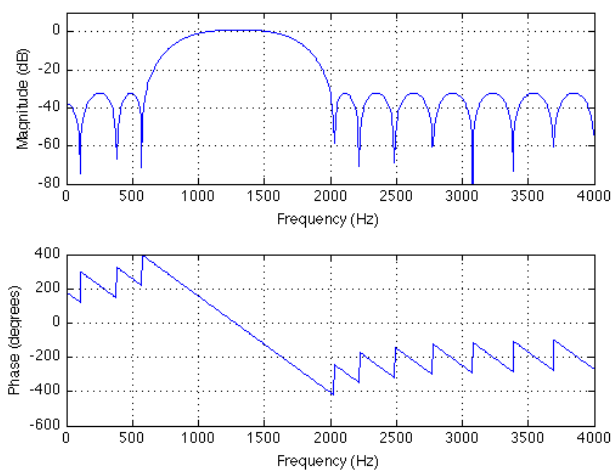

Figure 7: Parks-McClellan filter