

DSP Notes: Low Pass Filter Design by Windowing

Dr. Fred DePiero, CalPoly State University

FIR Low Pass Filter Design By Windowing Using SciLab

In the windowing method, the unit sample response, $hd(n)$ of a low pass filter is found via an inverse DTFT,

$$hd(n) = \frac{\sin(2\pi F_c * (n - (M-1)/2))}{\pi * (n - (M-1)/2)},$$

for $n=0, M-1$ and $n \neq (M-1)/2$

$$hd((M-1)/2) = 2 * F_c$$

where F_c is the cutoff frequency (cycles/sample) and M is the filter length. In SciLab, it is easy to setup the truncated unit sample response, $hd(n)$, using a for loop. In the following example, parameters associated with $F_c = 1/4$ and $M = 15$ are hard coded into the SciLab statements. The first step initializes hd with zeros and establishes it as a column vector. (Note PI is redefined for convenience).

```
--> PI = %pi;
--> hd = zeros(15,1);

--> for k=0:6, hd(k+1) = sin((k-7) * PI/2) / ((k-7) * PI); end
--> for k=8:14, hd(k+1) = sin((k-7) * PI/2) / ((k-7) * PI); end

--> k = 7;
--> hd(k+1) = 2 * 0.25;
```

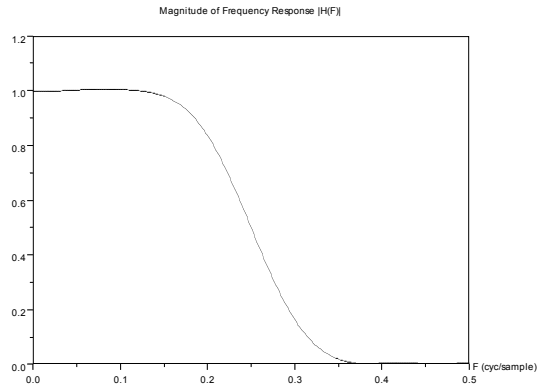
Two 'for' loops were used above to avoid reckless evaluation of $\sin n / n$ at $n=0$. The middle value of the filter ($n=7$, in this case) is then evaluated. Also note that array indices start with 1 in SciLab.

After $hd(n)$ is ready it is multiplied by a window function. Many window functions are available, use 're' for rectangular. (Try 'help window' in SciLab for more info). Continuing the above example, multiplying by a Hamming window for $M = 15$

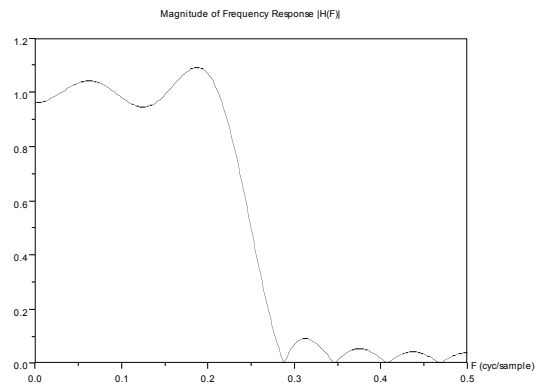
```
--> hd = hd .* window('hm',15,0);
```

To find the magnitude of the frequency response $H(F)$, for the filter designed above use:

```
--> [Hm,F]=frmag(hd,256);
--> plot(F, Hm)
--> xtitle('Frequency Response |H(F)|','F (cyc/sample)')
```



Note the increased ripples, due to Gibbs Phenomona, in the version of the filter designed with a rectangular window, below:



To compare the response of the ideal filter to the designed filter, the mean absolute difference in magnitude may be compared. First the magnitude of the ideal spectrum must be setup (here for an ideal low pass with $F_c = 0.25$):

```
--> Hd = zeros(1,length(Hm));
--> for k=1:length(Hd)/2, Hd(k) = 1; end
```

The mean square of the difference may then be found using:

```
--> ee = abs(Hd-Hm);
--> err = sum(ee) / length(ee)
```

This yields a mean absolute error of 8.1% in this case.