EE 141 DIGITAL SIGNAL PROCESSING

Lab 7: FIR Filter Design Using DFT and Projection Onto Convex Sets

In this lab, we will design discrete-time linear phase FIR filters using the iterative design technique discussed in class.

We have the following specifications for the design of our discrete-time filter $H(e^{j\omega})$:

- $0.98 \leq |H(e^{j\omega})| \leq 1$ for $0 \leq \omega \leq 0.25\pi$ and $1.75\pi \leq \omega \leq 2\pi$
- $|H(e^{j\omega})| \leq 0.02$ for $0.5\pi \leq \omega \leq 1.5\pi$

Instead of taking the DTFT, though, we will rely on 512-point DFTs. Since 512-point DFT is nothing but a sampling of the DTFT at frequencies $\frac{2\pi k}{512}$ for $k = 0, 1, \ldots, 511$, the above can be translated into

- $0.98 \leq |H[k]| \leq 1$ for $0 \leq k \leq 64$ and $448 \leq k \leq 511$
- $|H[k]| \leq 0.02$ for $128 \leq k \leq 384$

We desire to design a Type I linear phase FIR filter $h[n]$ of some relatively low order N matching these requirements. Therefore $H[k]$ will always be of the form

$$H[k] = e^{-j\frac{2\pi k}{512}\cdot\frac{N}{2}}A[k] = e^{-j\frac{\pi k N}{512}}A[k]$$

and it suffices to apply the design requirements to $A[k]$ instead of $H[k]$, i.e.,

- $0.98 \leq A[k] \leq 1$ for $0 \leq k \leq 64$ and $448 \leq k \leq 511$
- $-0.02 \leq A[k] \leq 0.02$ for $128 \leq k \leq 384$

Design algorithm in pseudo-code:

1) Set $N = 10$.
2) Initialize with $A[k] = 1$ for all $0 \leq k \leq 64$ and $448 \leq k \leq 511$, and $A[k] = 0$ for all other k.
3) Compute the 512-point inverse DFT of $H[k] = e^{-j\frac{\pi k N}{512}}A[k]$ using the `ifft` command, and set it to $g[n]$.
4) Set
$$h[n] = \begin{cases} g[n] & 0 \leq n \leq N \\ 0 & \text{otherwise} \end{cases}$$
5) Set $H[k] = \text{DFT}\{h[n]\}$ using the `fft` command.
6) Set $B[k] = H[k]e^{j\frac{\pi k N}{512}}$.
7) Clip $B[k]$ wherever it violates the specifications above and set the clipped version to $A[k]$. That is, for $0 \leq k \leq 64$ or $448 \leq k \leq 511$,
$$A[k] = \begin{cases} B[k] & 0.98 \leq B[k] \leq 1 \\ 1 & B[k] > 1 \\ 0.98 & B[k] < 0.98 \end{cases}.$$
The interval $128 \leq k \leq 384$ should be treated similarly. For all remaining k values, $A[k] = B[k]$.
8) Go to Step 3.

Clearly, this code will run forever, and it needs to be modified such that there is a stopping criterion. One way to set that criterion is to measure the "distance" between $h[n]$ obtained at the current and previous runs of Step 4:

$$\Delta h = \sum_{n=0}^{N}\left(h_{old}[n] - h_{new}[n]\right)^2.$$

If $\Delta h = 0$, $h[n]$ must be satisfying the specifications and therefore a solution to the problem has been found. If, on the other hand, Δh is very small, the infinite loop needs to be stopped, N needs to be increased by 2, and the whole code needs to be run again. So, an additional step must be

7.5) Compute Δh. If zero, quit. If very small, set $N = N + 2$, and go to Step 3.

Problem: Implement the algorithm described above, plotting both $h[n]$ and $|H[k]|$ after every 20th time Step 5 is executed. Indicate on the $|H[k]|$ plot the given filter specifications (as in the notes) and specify the current N each time.