EE123 Spring 2014 Miki Lustig
Digital Signal Processing

Laboratory 4: Optimal Filter Design

Miki Lustig

In this lab we will design optimal linear-phase FIR filters. Recall, that for a causal linear phase FIR filter with length $M+1$ we have, $H(e^{j\omega}) = \tilde{H}(e^{j\omega})e^{-j\omega M/2}$, where $\tilde{H}(e^{j\omega})$ is real. Therefore we get,

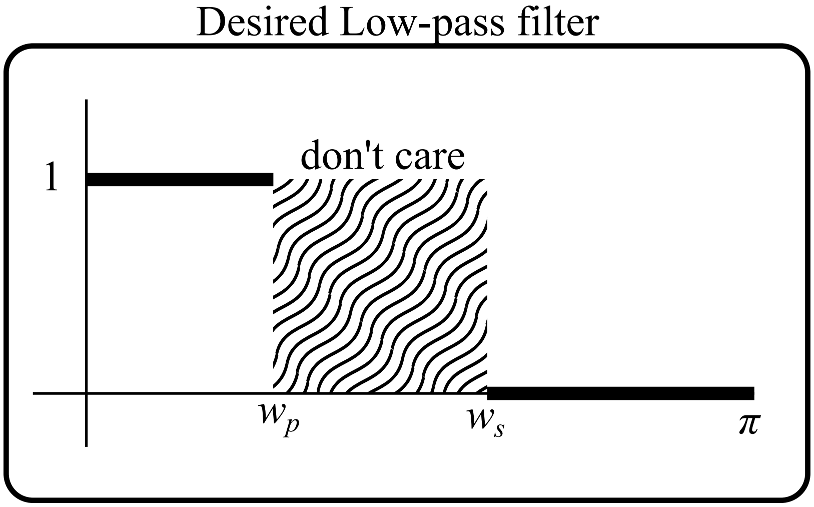
$$\tilde{H}(e^{j\omega}) = \tilde{h}[0] + 2\sum_{n=1}^{M/2} \tilde{h}[n]cos(\omega n)$$

We would like to find the optimal $\tilde{h}[n]$, and hence a causal $h[n]$ that will satisfy some design constraints.

The result of designing optimal filters depends of course on the optimality criteria that is used. Often in textbooks, a filter design criteria is described by a maximum deviation of the pass/stop band ripples, δ_p/δ_s and its transition width $\Delta\omega = \omega_s - \omega_p$. However, we can also define other criteria such as the mean-square pass/stop band deviation. These are not the same and will produce different results.

Task I: Optimal minimum mean-square deviation filter (Least-squares)

A least-squares design optimizes for the minimum mean-square deviation from a targeted frequency response profile, $H_d(e^{j\omega})$. For example, for the following specifications:



we would like to find $\tilde{H}(e^{j\omega})$ that satisfies:

$$\text{argmin}_{\tilde{h}} \quad \int_0^{\pi} |\tilde{H}(e^{j\omega}) - H_d(e^{j\omega})|^2 d\omega.$$

for the low-pass filter we get:

$$\text{argmin}_{\tilde{h}} \quad \int_0^{\omega_p} |\tilde{H}(e^{j\omega}) - 1|^2 d\omega + \int_{\omega_2}^{\pi} |\tilde{H}(e^{j\omega})|^2 d\omega$$

To solve this using numerical optimization, we are going to discretize the frequency by dividing the frequency space to P parts. A good approximation is obtained when $P \geq 15M$. In this case we solve for (written in "free form"):

$$\text{argmin}_{\tilde{h}} \quad \sum_{k \in \{0 \leq \omega_k \leq \omega_p\}} |\tilde{H}(e^{j\omega_k}) - 1|^2 \quad + \quad \sum_{k \in \{\omega_s \leq \omega_k \leq \pi\}} |\tilde{H}(e^{j\omega_k})|^2$$

This can be written in matrix form as,

$$\text{argmin}_{\tilde{h}} \quad ||A\tilde{h} - b||^2$$

where A is a matrix that computes the DTFT at the appropriate ω_ks and b is the corresponding desired frequency response at these locations in the spectrum.

The solution is given by the pseudo inverse, which is:

$$\tilde{h} = (A^T A)^{-1} A^T b$$

The matrix A can be easily constructed in Matlab. A simple way is to first generate a matrix A_p that computes the DTFT in the pass band, then A_s that computes it in the stop band and concatenating the two.

For example, A_p can be generated this way:

```
wp=0.4*pi; ws=0.6*pi; M=32;
P = M*15;
nn = 1:M/2;
ww = [0:P]/P*pi;
idxp = find(ww <= wp);
Pp = length(idxp);
[n1,w1] = meshgrid(nn,ww(idxp));
Ap = [ones(Pp,1) , 2*cos(w1.*n1)];
```

1) Write a function dzls.m that accepts an even M, pass-band and stop-band cutoffs wp, ws and returns an $M + 1$ linear-phase filter. The function call should be `h = dzls(M, wp, ws);`

2) Using your function design filters with $M = 32$, $\omega_p = 0.5\pi - \Delta\omega/2$, $\omega_s = 0.5\pi + \Delta\omega/2$ for $\Delta\omega = 0.1\pi$. Look at the filter and its frequency response. Compare to a filter generated by mat lab's `firls` function. Use: `h = firls(32,[0,0.45,0.55,1],[1,1,0,0]);`

3) Repeat your design for $\Delta\omega = 0.2\pi, 0.1\pi, 0.05\pi, 0.01\pi$. Take a look at how the filters look like. Plot the frequency response by overlaying then on the same graph. What is the mean-square deviation in each case? Comment on the tradeoffs.

4) Using your function design filters with $\omega_p = 0.45\pi$, $\omega_s = 0.55\pi$ for $M = 8, 16, 32, 64$ Take a look at how the filters look like. Plot the frequency response by overlaying then on the same graph. What is the mean-square deviation in each case? Comment on the tradeoffs.

Task II: Optimal linear-phase filter design with convex optimization

Optimization is a very broad and interesting topic and I strongly recommend taking a class like EE127A on optimization. It will be useful sometime in your life. I promise.

Here we will use a tool called CVX that was developed by Michael Grant and Steaphen Boyd that makes optimization much more accessible to the general public. CVX is a Matlab-based modeling system for convex optimization. CVX turns Matlab into a modeling language, allowing constraints and objectives to be specified using standard Matlab expression syntax. For example, consider the least-squares design problem you solved in Task I

$$\text{minimize} \quad ||A\tilde{h} - b||_2^2$$

Assuming you have the code to generate A and b, the following code segment will solve the problem:

```
cvx_begin
       variable h_tilde(M/2+1)
       minimize(norm(A*h_tilde - b,2))
cvx_end
```

Download and install CVX from `http://cvxr.com/cvx/download/` The install should work for windows, Mac and linux. Follow the instructions. If you are having difficulties, consult either the TA or myself. If you use a mac, download the unix version, it has binaries for macs.

Using CVX it is also possible to solve constrained optimization problems. Here, we will learn how to use it to design filter in which the maximum deviation is minimized. Specifically, we would like to solve for the filter $\tilde{h}[n]$ which satisfies:

$$\text{minimize} \quad \max\left\{\tilde{H}(e^{j\omega_k}) - H_d(e^{j\omega_k})\right\}, \qquad 0 \leq k < P$$

Assuming you have the code to generate A_p and A_s which compute the DTFT for the pass-band and stop band respectively the following code will solve the problem:

```
cvx_begin
       variable ht(M/2+1)
       variable delta(1)

       minimize(delta)
       subject to
             Ap*ht <= (1 + delta)
             Ap*ht>= (1 - delta)
             As*ht <= delta
             As*ht >= -delta
             delta >= 0
cvx_end
h = [ht(end:-1:2),ht];
```

1) Write a function dzopd.m that accepts an even M, pass-band and stop-band cutoffs wp, ws and returns an $M+1$ linear-phase filter that minimizes the maximum deviation δ. The function call should be `[h,delta] = dzopd(M, wp, ws);`

2) Using your function design filters with $M = 32$, $\omega_p = 0.5\pi - \Delta\omega/2$, $\omega_s = 0.5\pi + \Delta\omega/2$ for $\Delta\omega = 0.1\pi$. Look at the filter and its frequency response. Compare the frequency response you got to the least-squares design from Task I. What are the main differences?

3) Repeat your design for $\Delta\omega = 0.2\pi, 0.1\pi, 0.05\pi, 0.01\pi$. Take a look at how the filters look like. Plot the frequency response by overlaying then on the same graph. What is the ripple in each case? Comment on the tradeoffs.

4) Using your function design filters with $\omega_p = 0.45\pi$, $\omega_s = 0.55\pi$ for $M = 8, 16, 32, 64$ Take a look at how the filters look like. Plot the frequency response by overlaying then on the same graph. What is the ripple in each case? Comment on the tradeoffs.

Task III: Optimizing for minimum transition width or minimum filter length

Sometimes, we know the specifications for the pass-band and stop band ripple and are interested in optimizing for the minimum transition width. This may seem difficult, since the transition width appears within the matrix A. The problem is indeed non-convex and can not be optimized directly using CVX. However, it does belong to a class of problems which are called quazi-convex which we can find a solution for using bisection.

Recall the results you got in Task II problem 3. The ripple is a non-decreasing function of $1/\Delta\omega$. That is, a small $\Delta\omega$ will result in the ripple, δ, to be large and a large $\Delta\omega$ will result in the ripple being small. We can therefore use bisection and calls to `dzopd` to find the optimal $\Delta\omega$ that satisfies our ripple constraints!

Bisection works in the following way; We start with $\Delta\omega_{\text{min}} = 0$ and $\Delta\omega_{\text{max}} = \pi$. The solution for $\Delta\omega_{min}$ of course violates our constraints and the one for $\Delta\omega_{\text{max}}$ is guaranteed to satisfy it. We then initialize $\Delta\omega = \Delta\omega_{\text{min}}/2 + \Delta\omega_{\text{max}}/2$. It the result satisfies our ripple constraint then we set $\Delta\omega_{\text{max}} = \Delta\omega$. If it doesn't then we set $\Delta\omega_{\text{min}} = \Delta\omega$. We repeat till $\Delta\omega_{\text{max}} - \Delta\omega_{\text{min}}$ is less than some tolerance (for example 1e-3).

1) Write a function `dzopw.m` that accepts an even M, The cutoff frequency wp, and the desired ripple δ and returns an $M + 1$ linear-phase filter with the smallest transition width $\Delta\omega$ ($ws = wp + \Delta\omega$). The function call should be `h = dzopw(M, wp, delta);` Use the function `dzopd` and bisection.

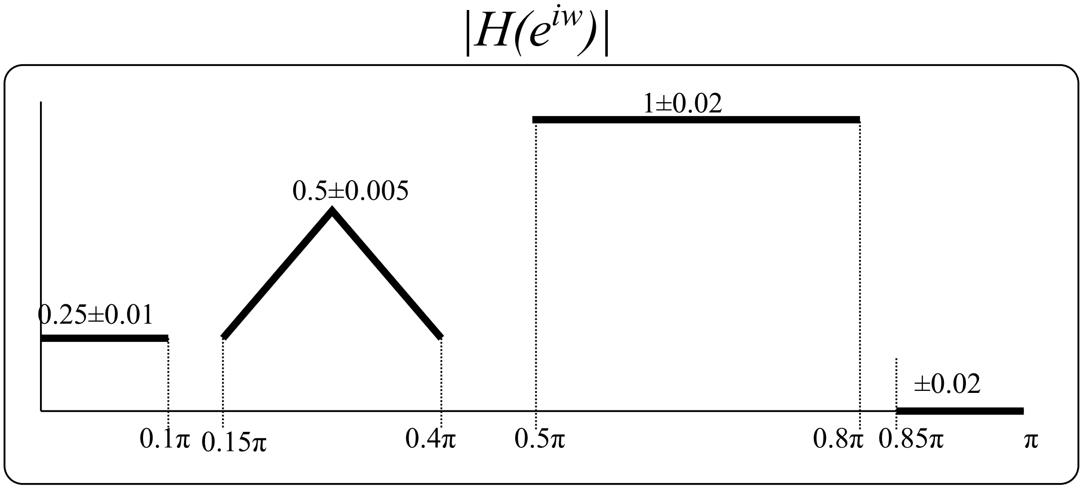
2) Design a filter with $M = 32$, $wp = 0.45\pi$ and $\delta = 0.05$. What is ws? Plot the filter and its frequency response.

3) Optimizing the filter length is also quasi-convex and can be solved by applying bisection on M. Write a function `dzopM.m` that accepts the cutoff frequencies wp, ws and the desired ripple δ. The function returns the minimum $M + 1$ linear-phase filter with the smallest even M (M must be even here!). The function call should be `h = dzopM(wp,ws delta);` Use the function `dzopd` and bisection. (HINT: Mmax should be large enough to make the problem feasible. If it is not, double it till it does.)

4) Design a filter with $wp = 0.25\pi$, $ws = 0.3\pi$ and $\delta = 0.001$. What is M? Plot the filter and its frequency response.

Task IV: Design challenge

You would like to design a linear phase filter with the following Magnitude response:



1) Using CVX, construct and solve an optimization that satisfies these constraints. Make sure that the response in the "Don't care" regions is less than 1.02 and greater than -1.02. Using bisection, find the one with the lowest order. (HINT: should be less than $M = 70$) What is it? Plot the filter and its frequency response

I hope you had fun.