

T-61.246 Digital Signal Processing and Filtering

Round R7 (v 1.0, 13.10.2004), Matlab III, 1.-5.11.2004. Registration in WWWTopi. Bring your own **earphones** if you have. Make sure that you have a valid **Windows password**.

The assistant will guide you through the exercises 1-3. Write down the code into separate files in your working directory (Z:\DSP2004\). Feel free to ask the assistant, if you have troubles. The problems marked with [Tx] are from the course exercise material, where **x** refers to the problem.

If you want to have **bonus points**, there are a couple of exercises in the end of this paper. The deadline for the bonus point returnings is **Monday 8th Nov at 12.00**.

Exercises

- Look at the **filter specifications** for each filter in Figure 1, read the values for the following quantities and convert them to values that Matlab requires (e.g. `help buttord`)¹

- filter type: lowpass / highpass / bandpass / bandstop
- sampling frequency $f_T =$ Hz, in Matlab it corresponds to 2.
- passband cut-off freq. $f_p =$ Hz, in Matlab $w_p = [\quad]$, $(w_p/2) = (f_p/f_T)$
- stopband cut-off freq. $f_s =$ Hz, in Matlab $w_s = [\quad]$, $(w_s/2) = (f_s/f_T)$
- maximum passband attenuation (maximum ripple) in decibels $\alpha_{max} =$ dB, in Matlab $R_p = [\quad]$.
- minimum stopband attenuation in decibels $\alpha_s =$ dB, in Matlab $R_s = [\quad]$.

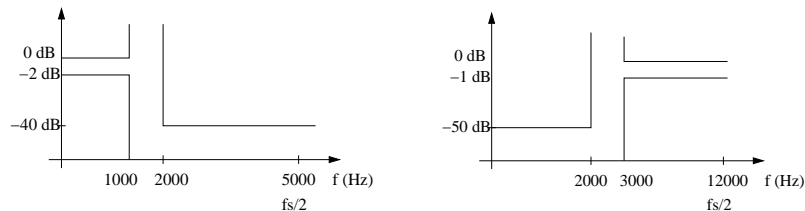


Figure 1: Filter specifications of two filters for Problem 1. The notation “fs/2” in the figure means the half of the sampling frequency $f_T/2$.

Sampling frequency f_T (f_s !?)	0	$f_T/4$	$f_T/2$	$3f_T/4$	f_T	Hz
Frequency f , $f_T = 20$ kHz	0	5000	10000	15000	20000	Hz
Angular frequency Ω	0	10000π	20000π	30000π	40000π	rad/s
Normalized angular frequency ω	0	$\pi/2$	π	$3\pi/2$	2π	rad/sample
Normalized frequency f_{norm}	0	0.5	1	1.5	2	1

Table 1: Examples on frequency normalization, with the sampling frequency $f_T = 20$ kHz. Matlab uses f_{norm} . Angular frequency Ω is used with analog signals, ω with digital sequences.

¹There are also several ways to express the filter specifications, see Section 7.1.1 in Mitra. For example, equiripple FIR filter with the function `remezord` requires ripples as deviations in linear scale.

- Basics of filter design are introduced in Section 7.1 in Mitra. Here is a short list:

- Determine the specifications of the filter. You may need to use it, for example, as a lowpass filter. Draw the specifications on paper. You may also require to analyze the input signal in order to know what are the cut-off frequencies.
- Choose the type of the filter. Is it IIR or FIR? Is it linear-phase (FIR) or not? Choose from IIR filters: Butterworth, Chebyshev I, Chebyshev II, Elliptic. Choose from FIR filters: window method, frequency method, least-square.
- Estimate the order of the filter. When working with Matlab, write the code into a file, because probably you have to modify filter. Filter order should be big enough to fulfill the specifications and it should be low in order to be avoid computation. Some Matlab functions: `buttord`, `cheblord`, `cheb2ord`, `ellipord`, `remezord`. Type `help functionname` to read more. You may also have to look at some table (Table 7.2, page 454 in Mitra) in order to get the correct order for a fixed window FIR filter.
- Compute the coefficients of the transfer function. This means in Matlab finding the vectors **B** and **A** for $H(z) = B(z)/A(z)$. Some Matlab functions: `butter`, `cheby1`, `cheby2`, `ellip`, `fir1`, `fir2`, `firls`, `remez`. Type `help functionname` to read more. There are also lots of window functions: `boxcar`, `hamming`, `hann`, `blackman`, `kaiser`.
- Check that the specifications are fulfilled! For instance, use `freqz`. You can draw the specifications in the same figure with the function `speksit.m`, which can be found from the course web page.

When plotting the spectra of signals or the frequency responses of filters, zoom the x-axis to be from 0 to $f_T/2$. You can use `axis([0 fT/2 minx maxy]);` or set the axis properties directly `set(gca, 'XLim', [0 fT/2]);`

- Apply your filter to the input signal. For instance, use `filter`.

Task: Design an elliptic IIR lowpass filter with f_p (end of passband frequency) 3000 Hz, f_s (start of stopband frequency) 3300 Hz, and f_T (sampling frequency) 20000 Hz. Passband ripple may be 3 dB, and minimum stopband attenuation 40 dB.

Go on all phases of the list above. Read `help ellipord` | `ellip`.

Write down the order of the filter and the coefficients of the transfer function.

Use `freqz`, `zplane`, `impz` in order to get "all possible" plots (amplitude, phase response, pole-zero plot, impulse response) for the filter.

In the end, create a signal consisting of two cosines with $f_1 = 1000$ Hz and $f_2 = 4000$ Hz. Filter the signal and listen to the original and filtered signal.

```
% Matlab III, Round 7, Problem 2
% Filter design (skeleton)
```

```
% (i) specifications
fT = 20000; % sampling frequency 20 kHz
Wp = ; Ws = ; Rp = ; Rs = ; % give correct values!
```

```
% (ii), (iii) filter order, help ellipord
[N, Wn] = ellipord( ); % add correct variables
```

```

% (iv) filter coefficients, help ellip
[B, A] = ellip( );           % add correct variables

% (v) check if specifications are fulfilled
figure(1); clf;
[H, w] = freqz(B, A, 512, fT);
magdB = 20*log10(abs(H)/max(abs(H))); % squared magnitude in dB
plot(w, magdB);           % amplitude response in decibels
speksit( );              % add correct variables

% (vi) generate signal
M = 10000; t = [1 : M]/fT; % 0.5 seconds, if fT=20 kHz
A = [1 1]; f = [1000 4000]; th = [0 0]; % ampl., freq., phase
x = sum(repmat(A',1,M) .* cos(2 * pi * f' * t + repmat(th',1,M)));

% (vi) use the filter and test it
y = filter(B, A, x);       % apply X -> H(z) -> Y
soundsc(x, fT); pause;    % listen to the original, and wait for pressing
soundsc(y, fT);          % listen to the filtered

```

3. This demo is closely related to [T45], Section 7.6 in Mitra. Next, a FIR filter is designed using the window method. The impulse response $h_{ideal}[n]$ of an ideal lowpass filter $H(e^{j\omega})$ with a cut-off angular frequency ω_c is of infinite length (IIR). Therefore, in order to get a N th order ($N = 2M$) FIR filter, we use a window function $w[n]$ of length $2M + 1$

$$\begin{aligned}
 h_{ideal}[n] &= \sin(\omega_c n) / (\pi n), & n = -\infty \dots +\infty \\
 h_{FIR}[n] &= h_{ideal}[n] \cdot w[n], & \Rightarrow h_{FIR}[n] = 0, \text{ when } |n| > M
 \end{aligned}$$

Task: [T45(a)] Use the rectangular window function (`help boxcar`) to create a 4th order FIR filter with the cut-off frequency $\omega_c = 0.75\pi$. The filter is noncausal - how do you get it causal?

```

M      = 2;                % order N=2M, length 2M+1;
K      = 100;              % dummy variable, K >> M
n      = [-K+1 : K-1];    % time axis
wc     = 0.75*pi;         % cut-off frequency w_c
h_ideal = sin(wc*n)./(pi*n); % h_ideal[n]
h_ideal(K) = wc/pi;       % h_ideal[0]
winf   = boxcar(2*M+1)';  % rectangular window function w_R[n]
h_FIR  = h_ideal(K-M:K+M) .* winf; % FIR filter

subplot(3,2,1); stem(n, h_ideal); title('Time-domain'); ylabel('h_i[n]');
subplot(3,2,3); stem([-M:M], winf); ylabel('w[n]');
subplot(3,2,5); stem([-M:M], h_FIR); xlabel('Time index n'); ylabel('h_F[n]');

subplot(3,2,2); [H_I,w] = freqz(h_ideal,1); plot(w/pi, abs(H_I));
title('Frequency-domain');
subplot(3,2,4); [H_R,w] = freqz(winf,1); plot(w/pi, abs(H_R));
subplot(3,2,6); [H_F,w] = freqz(h_FIR,1); plot(w/pi, abs(H_F));
xlabel('Angular frequency (\times \pi)');

```

Bonus Problems

Return your solutions (code / output, not hand-written) **on paper** in the returning box of CIS Laboratory in CS building. Replace symbols **A**, **B** and **C** by their numeric values: **A** is the third digit in your student ID, **B** fourth, and **C** fifth digit. **NOTICE THE NEW INSTRUCTIONS! - HUOMAA UDEET PALAUTUSOHJEET!**

- B30) * (1p) Implement a lowpass Butterworth IIR filter, which works with the sampling frequency of $f_T = 24000$ Hz, passband ends at 2000 Hz, stopband starts from 3000 Hz. The maximum ripple in the passband is allowed to be 2 decibels, and the minimum attenuation in the stopband is required to be 50 decibels. What is the order of the filter? Make sure that your specifications are fulfilled! IMPORTANT: Check that you have scaled the x-axis according to the same scale as in your specifications.
- B31) ** (1p) Continue from Problem 2. Design filters with the same specifications but using functions `buttord`, `cheb1ord`, `cheb2ord` and `butter`, `cheby1`, `cheby2`. Plot also the pole-zero diagrams. Compare the results.
- B32) ** (1p) Bandpass and bandstop filters require $W_p = [W_{p1} \ W_{p2}]$ and $W_s = [W_{s1} \ W_{s2}]$ in Matlab. The value N returned by the order function is only half of the order of the filter. Draw specifications for a bandpass filter with $W_p = [0.04 \ 0.05] * (B + 5)$, $W_s = [0.02 \ 0.06] * (B + 5)$, and other specifications from Exercise 2. Implement the filter.
- B33) ** (1p) Continue from Problem 3. Create a lowpass FIR filter with the same specifications but using Hamming and Blackman window functions.

After that, increase the filter order N to 50, and run the code for each three window functions.

Comment the results based on the theory.

- B34) *** (2p) Examine the following window functions: rectangular (`boxcar`), Hamming and Blackman. What are the connections between main lobes and side lobes against spectral resolution?

Plot the sequence $w[n]$, the amplitude response $|W(e^{j\omega})|$, and the pole-zero plot of each window of length 128 or 256.

Load the variable `B33signal.mat` with `load B33signal`, there are two variables: `signal` `y` and sampling frequency `Fs`. Divide the signal into small segments (128 or 256) and compute short-time spectra using different windows: `xF = fft(x .* w)`; over the whole signal. Non-stationary signal contains two strong components near to each other and one weak component a bit further.

Compare the window functions according your results.