T-61.246 Digital Signal Processing and Filtering

Round R5 (v 1.0, 7.10.2004), Matlab II, 12.-15.10.2004. **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 1-4. Write down the code into separate files in your working directory (`Z:\DSP\`). 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 18th Oct at 12.00**. Return your solutions (code / output, not hand-written) **on paper** in the returning box of CIS Laboratory in CS building.

Exercises

1. Analyze the audio sample `kiisseli.wav` in time and frequency domain. If not found in WWW, you can use `load train; fs = Fs;` and omit the `wavread` line.

```
% file r5prob1.m
[y, fs, nbits] = wavread(’Z:\DSP\kiisseli.wav’);
M   = length(y);
t   = [ 0 : M-1 ] / fs;      % time-axis
figure(1); clf;
plot(t, y); grid on; xlabel(’time (sec)’); title(’/kiisseli/’);

yF  = fft(y);                % DFT of signal x, 0..2pi
mag = 20*log10(abs(yF));     % in decibels; OR linearly = abs(yF)
w   = fs * [0 : (M-1)]/M;    % frequency axis
figure(2); clf;
plot(w, 20*log10(abs(yF)));  % spectrum
grid on; xlabel(’frequency (Hz)’); title(’DFT of /kiisseli/’);
% axis([xmin xmax ymin ymax]); % zooming from the code

figure(3); clf;
specgram(y, [], fs);         % spectrogram
colorbar;                    % adds a colorbar: color <=> value
```

Task: What is the sampling frequency of the audio file? The spectrum in Figure No. 2 seems to be symmetric, why so? What can you see in the spectrogram? (You can also run `specgramdemo(y, fs)` from command line.)

2. In Round 1 it was shown that a complex-valued function $H(\omega) = 2-e^{-j\omega}$ can be visualized in many ways. Digital LTI systems have transfer functions of type

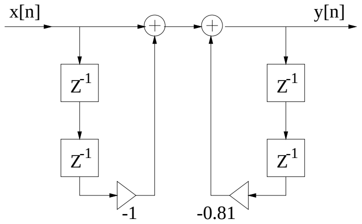
$$\begin{aligned} H(z) &= b_0 + b_1 z^{-1} + \ldots + b_M z^{-M} \\ H(z) &= \frac{b_0 + b_1 z^{-1} + \ldots + b_M z^{-M}}{a_0 + a_1 z^{-1} + \ldots + a_N z^{-N}} \end{aligned}$$

where the first is a FIR and the latter IIR. The frequency response is received by substitution $z = e^{j\omega}$. It can be written in Matlab using the coefficients of the numerator and denominator polynomials:

$$H(e^{j\omega}) = \frac{b_0 + b_1 e^{-j\omega} + \ldots + b_M e^{-jM\omega}}{a_0 + a_1 e^{-j\omega} + \ldots + a_N e^{-jN\omega}}$$

```
B = [B(1) B(2) ... B(M+1)];   % or ‘‘num’’ = numerator polynomial
A = [A(1) A(2) ... A(N+1)];   % or ‘‘den’’ = denominator polynomial
```

For example, in [**T28**] we have a second-order LTI system with feedback as shown below.



The frequency response is received from the difference equation:

$$\begin{aligned} y[n] &= x[n] - x[n-2] - 0.81y[n-2] \\ y[n] + 0.81y[n-2] &= x[n] - x[n-2] && \mid \text{z-transform} \\ Y(z) + 0.81z^{-2}Y(z) &= X(z) - z^{-2}X(z) \\ Y(z)(1+0.81z^{-2}) &= X(z)(1-z^{-2}) && \mid : X(z) \\ Y(z)/X(z)(1+0.81z^{-2}) &= (1-z^{-2}) && \mid : (1+0.81z^{-2}) \\ H(z) = Y(z)/X(z) &= \frac{1-z^{-2}}{1+0.81z^{-2}} && \mid z \leftarrow e^{j\omega} \\ H(e^{j\omega}) &= \frac{1-e^{-2j\omega}}{1+0.81e^{-2j\omega}} \end{aligned}$$

The numerator polynomial is $B(e^{j\omega}) = 1 + 0 \cdot e^{-j\omega} - e^{-2j\omega}$. The coefficients of the polynomial are $B_k = \{\underline{1},\ 0,\ -1\}$ with respect to powers of $e^{-j\omega}$. In the same way, the denominator polynomial is $A(e^{j\omega}) = 1 + 0 \cdot e^{-j\omega} + 0.81e^{-2j\omega}$, and the coefficients are $A_k = \{\underline{1},\ 0,\ 0.81\}$.

Shortly, we have now a complex-valued function $H(\omega)$, which could be analyzed just like in Round 1. However, the filter can be analyzed using commands from Signal Processing Toolbox:

```
% file r5prob2.m
B  = [1  0    -1];
A  = [1  0  0.81];  % H(z) from [T28]

figure(1); clf;
freqz(B, A);          % default amplitude (log) and phase response
figure(2); clf;
[H, w] = freqz(B, A);
plot(w, abs(H));     % linear y-axis; OR in decibels (w, 20*log10(abs(H)))
grid on; xlabel(’angular frequency \omega (rad)’); title(’Amplitude resp.’);
figure(3); clf;
[H, w] = freqz(B, A, 1024, 24000); % fs = 24000 Hz, DFT-1024 (512 default)
```

```
plot(w, angle(H));                 % now x-axis 0 ... 12000 Hz = fs/2
grid on; xlabel(’frequency (Hz)’); title(’Phase response’);

figure(4); clf;
zplane(B, A);       % pole-zero plot

figure(5); clf;
[h, n] = impz(B, A); % impulse response
stem(n, h);
grid on; xlabel(’index n’); title(’Impulse response h[n]’);
```

If you want see the block diagram, you can use a program by Iikka Väkiparta. It can be found in WWW `http://www.cis.hut.fi/Opinnot/T-61.246/Suodin/`, and you should give the vectors `B` (numerator polynomial) and `A` (denominator polynomial) just like in Matlab.

Task: Consider the following LTI systems and classify if they are lowpass / highpass / bandpass / bandstop / allpass / comb filter.

a) $H(e^{j\omega}) = \frac{1}{1+0.2e^{-j\omega}+1.2e^{-2j\omega}}$

b) $H(e^{j\omega}) = \frac{0.07+0.1e^{-j\omega}+0.1e^{-2j\omega}+0.07e^{-3j\omega}}{1-1.4e^{-j\omega}+1.1e^{-2j\omega}-0.3e^{-3j\omega}}$

c) $H(e^{j\omega}) = 1 + e^{-4j\omega}$

3. The filtering can be executed using the command `filter`: `y = filter(B, A, x)`. In the time-domain the filtering is convolution (`conv`) of the input and the impulse response - in the frequency-domain it is product of the input spectrum and frequency response.

$$y[n] = h[n] \circledast x[n] \qquad \leftrightarrow \qquad Y(e^{j\omega}) = H(e^{j\omega}) \cdot X(e^{j\omega})$$

Task: Analyze the amplitude response $|H(e^{j\omega})|$ of the elliptic 3rd order filter

$$H(e^{j\omega}) = \frac{0.34 - 0.77e^{-j\omega} + 0.77e^{-2j\omega} - 0.34e^{-3j\omega}}{1 - 0.6e^{-j\omega} + 0.72e^{-2j\omega} + 0.1e^{-3j\omega}}$$

using the tools in Problem 2. Apply the filter for the signal `music.wav`. Listen to the original signal and the filtered signal.

4. The connection between pole-zero plot and the amplitude response can be visualized using SPTool. For a transfer function $H(z) = B(z)/A(z)$ the poles are the roots of $A(z)$, i.e. $A(z) = 0$, and the zeros are the roots of $B(z)$, i.e. $B(z) = 0$. The thumb of rules are explained in detail in [**T27**]. Instructions in Finnish: `http://www.cis.hut.fi/Opinnot/T-61.246/Demo/esim4.shtml`, and in English in B24 next page.

Task: Create a highpass filter with the sampling frequency of 8000 Hz. Let the cut-off frequency (stopband/passband) be 3000 Hz. Implement both FIR (only zeros) and IIR (both zeros and poles) filter.

Bonus problems

Print some pictures and relevant code (not hand-written) for each problem in your returning.

B19) * Continue from Problem 1. Pick up the first long vowel /ii/ from the vector into a vector `y2`, and plot the figures 1 and 2 again. If considered as a quasi-periodic (almost periodic) signal, what is the fundamental period T_0 (Figure 1) and the fundamental frequency f_0 (Figure 2)? Hint: You can plot the signal with `plot(y)`, and read the index numbers from x-axis, and then call `y2 = y(startindex : stopindex)`. If you don't remember T_0 or f_0, read `http://www.cis.hut.fi/Opinnot/T-61.246/Demo/esim2.shtml`.

B20) ** What does it mean if the signal is (not) stationary? When doing Fourier-analysis for speech signal, why is it important to segment the signal? Run the demo `bonus20.m` in `http://www.cis.hut.fi/Opinnot/T-61.246/Laskarit/`, and see the signals and spectra in Figure No. 20.

In addition, there is a trade-off between time windows and frequency resolution as shown in Figure No. 21. The longer the sequence, the better the spectral resolution. However, in order to be stationary, the signal has to be short.

B21) *** (2p) Continue from B19. Divide the original vector into adjacent pieces of N samples, $(1 \dots N,\ N+1 \dots 2N,\ 2N+1 \dots 3N, \dots)$, and compute spectrum of each subvector. You may want to listen to the subvector, and keep breaks after each subvector (`pause`). Change the value of N and give your suggestion when the signal segments are stationary enough.

B22) * Analyze the following $H(e^{j\omega})$. Plot the amplitude response, phase response, pole-zero plot and impulse response. Is your filter stable? Let A be your date of birth (range 1-31), B the month of your birth (range 1-12), and then $r = 1.4-(A/100)$ and $s = 0.9-(B/10)$.

$$H(e^{j\omega}) = \frac{1}{1 + r \cdot e^{-j\omega} + s \cdot e^{-2j\omega}}$$

B23) ** Just like in B22, but **also** write down the impulse response $h[n]$ in **closed form** (you need partial fraction expansion, `residuez`) like in [**T28(f)**]. What is the order of the filter? Is it IIR or FIR? Scale the factor K so that the maximum is 1.

$$H(z) = K \cdot \frac{1.2 + 2.3z^{-1} + 3.1z^{-2} + 2.3z^{-3} + 1.2z^{-4}}{1 + 3.13z^{-1} + 3.75z^{-2} + 2.03z^{-3} + 0.42z^{-4}}$$

B24) * A rough estimate of the amplitude response can be sketched by looking at the pole-zero plot. The rules of thumb can be read in [**T27**].

Open a Matlab GUI SPTool by typing `sptool`. Press the buttons “New” and “View” in the middle column “Filters”. You will get two different windows, “Filter Designer” and “Filter Viewer”, respectively. Place them at the screen so that you see the both. Click the pop-up menu “Algorithm” in the design window, and choose “Pole/Zero Editor”.

You can add and remove any number of poles (crosses) and zeros (circles) in the Z-plane. You can press the button “Delete All” in the bottom of the figure in order to remove old zeros and poles. The amplitude response will be shown interactively in the viewer window. The sampling frequency is by default 8192 Hz, but can be changed in the box above the pole-zero plot.

Task: Create a bandpass filter using the pole-zero editor. The filter should be of low order, passband at 2000 . . . 2500 Hz, while the sampling frequency is 10000 Hz. Design a (a) FIR (only zeros) and (b) IIR (zeros and poles) filter.