T-61.3010 Digital Signal Processing and Filtering

(v 1.1, 15.3.2006), Matlab 3. **Registration** in WWWTopi. Bring your own **headphones** if you have. The assistant will guide you through the exercises, but you may go on your own speed. Feel free to ask the assistant, if you have troubles. This exercise is designed for **Matlab 7** - most programs and features work also with earlier versions. The problems marked with [**Bx**] are from the course exercise material (Spring 2006), where **Bx** refers to the problem.

In **Windows** just click `Programs - Matlab`.
In **Unix** type `use matlab`, `matlab-keyfix` (see instructions) and then `matlab &`. Adjust audio volume from GNOME - Applications - Multimedia - Audio Volume.

Write down the code into separate files in your working directory (e.g. `Z:\DSP\` or `~/DSP/`) for future use. Set the "Current Directory" in Matlab to point to the working directory (or type `cd <workdir>`).

In the end of this session **you should know**:
(a) some basics of filter design using SPTool, (b) characteristics of linear-phase FIR filter, (c) how to record voice and read it to Matlab.

1. Read an audio file into Matlab using `[x, fs, nbits] = wavread('Matlab3.wav');` It can be seen that `x` is a matrix with two columns, i.e. audio file is stereo with left and right channel.

   Open a graphical user interface (GUI) SPTool from command line by typing `sptool`.

   You can import an audio signal into GUI from `File - Import`. Assign `x` to data and `fs` to sampling frequency. You can see and hear the signal by clicking `View` in the left-most column "Signals" (probably not hear in Unix-Matlab).

   You can design, for example, a Butterworth lowpass filter by clicking `New` in the middle column "Filters" and choosing `Butterworth IIR` in the algorithm in the "Filter Designer" window. Your filter name is now (probably) `filt1` in the pop-up menu top-left. You can set the sampling frequency of the signal as the sampling frequency of the filter. Exact numbers can be given in the edit boxes left, or the specifications can be dragged with mouse. Set the passband edge `Fp` to be 3000 Hz, maximum passband attenuation (ripple) `Rp` to 1 dB, stopband edge `Fs` to 4000 Hz, and minimum stopband attenuation `Rs` to 40 dB. These values are called **filter specifications**. See more [**B46**].

   GUI has automatically computed the coefficients of $H(z) = B(z)/A(z)$, that is, executed filter design. Now you can read some actual values in "Measurements" right: the order is probably 18 and passband ripple 0.7966 dB.

   You can filter the signal by clicking the signal in left column and just created filter in middle column of main window SPTool. After that press `Apply` button.

   You can see a power spectrum estimate of a signal in right-most column by choosing a signal in left-most column and then clicking `Create`. There are plenty of options with different estimate methods.

   You can analyze your filter by clicking `View` in middle column. You can also export the filter coefficients `File - Export - Export to Workspace` and use command line functions.

   Verify that everything which is done with GUI, can be also done from the command line:

```
B = filt1.tf.num;           % Matlab struct/record with several fields...
A = filt1.tf.den;           % ... filt1, filt1.tf, filt1.tf.num = [<numeric>]
figure(1); freqz(B, A, 512, fs)
figure(2); zplane(B, A);    % all zeros should be at z = -1
y = filter(B, A, x);        % filtering
xleft = x(:,1); yleft = y(:,1);      % vectors for specgram and pwelch
figure(3); specgram(xleft, [], fs);  % [] means choosing default values
figure(4); specgram(yleft, [], fs);
figure(5); pwelch(xleft, [], [], [], fs);
soundsc(y, fs);             % filtered signal. soundsc for Windows and ...
%mysoundsc(y, fs);          % mysoundsc.m for UNIX from course www
```

   Note that all zeros should lie at $z = -1$ in case of Butterworth lowpass filter. Filter design functions are the topic of the next Matlab session.

   Task: Analyze the signal again. Why the signal seems to be bandlimited to 16 kHz even if the sampling frequency is 44 kHz. Design an elliptic IIR bandpass filter using SPTool GUI. The passband should be from 3 to 4 kHz. Choose other values. Filter the signal.

2. There are four types of linear-phase filters. They can be recognized by looking at the impulse response, the transfer function, or pole-zero plot. In $h[n]$ (or $H(z)$) there must be a certain type of symmetry in coefficients. In pole-zero plot the zeros must lie in a mirror-symmetric way with respect to the unit circle. See more [**B35**], or book/slides.

   Load in `linearF.mat` using `load linearF.mat`, which contains filter coefficients of four different linear-phase FIR filters $H_1(z)$ (`B1`), $H_2(z)$, $H_3(z)$, and $H_4(z)$, and a demo sequence $x[n]$, which consists of three cosines $x_1[n] = \cos(\omega_1 n)$ (`x1`), $x_2[n]$, and $x_3[n]$, with normalized angular frequencies $\omega_1 = 0.1\pi$ (rad/sample), $\omega_2 = 0.2\pi$, $\omega_3 = 0.6\pi$, respectively. Because $\omega = 2\pi/N$, there are 20 samples in one period of $x_1[n]$.

   Let us analyze the filter $H_1(z)$ by plotting magnitude response, phase response, group delay, impulse response, and pole-zero plot. Filter each signal component with $H_1(z)$, and plot the components and the sum of them before and after filtering. Remember that in frequency domain $Y = H \cdot X$, which in polar coordinates gives $|Y| = |H| \cdot |X|$, and $\angle Y = \angle H + \angle X$. For instance, $|H(0.1\pi)| \approx -3$ dB (signal power to half!) and $|H(0.2\pi)| \approx -19$ dB (much more attenuated!), and $\angle_\Delta H(0.1\pi) \approx -72$ degrees $\approx 0.4\pi = 4 \cdot 0.1\pi$ and $\angle_\Delta H(0.2\pi) \approx -144$ degrees $\approx 0.8\pi = 4 \cdot 0.2\pi$.

```
close all                   % close all windows
load linearF                % contains: B1, B2, B3, B4, x1, x2, x3
tf2latex(B1,1)              % from course web page
figure(1); freqz(B1,1);     % magnitude and phase response
figure(2); grpdelay(B1,1); % group delay
figure(3); impz(B1,1);      % impulse response (SYMMETRY HERE!)
figure(4); zplane(B1,1);    % pole-zero plot (SYMMETRIC ZEROS!)
y1 = filter(B1, 1, x1);     % filtering of each component x1, x2, x3
figure(5); subplot(4,1,1); plot(x1); hold on; plot(y1,'k');
% ... the same for x2 and x3
subplot(4,1,4); plot(x1+x2+x3); hold on; plot(y1+y2+y3,'k');
```

   Task: Show mathematically that group delay of $H_1(e^{j\omega}) = [1 + e^{-j\omega} + \ldots + e^{-j8\omega}]/9$ is constant 4. Show that $H_1(z) = [1 - z^{-9}]/[1 - z^{-1}]$, where $1 - z^{-9} = 0$ gives 9 zeros at equal intervals on unit circle.

Task: Analyze pole-zero plot of filter $H_2(z)$ and verify that zeros have mirror-symmetry with respect to unit circle. If you want to compute exact locations of zeros, use `tf2zp` or `roots`.

Task: See the phase response of filter $H_3(z)$ at $\omega = \{0.1\pi,\ 0.2\pi,\ 0.6\pi\}$. Analyze the delay or phase shift of each $x_i[n]$. How much more the angle of $x_2[n]$ is shifted than that of $x_1[n]$? How much is that in time indices?

3. Matlab assignment for this course has been released. First topic is to record a word with a certain vowel and analyze that signal.

   Consider the recording procedure. Human speech is longitudinal (pitkittäinen) vibration in the air. Microphone coil vibrates according to the changes of air pressure. This is turned to electricity and sampled in the sound card with a certain sampling frequency and number of bits for each sample. The sequence $x[n]$ is then applied by software.

   Consider now a task where a word $w(t)$ is uttered with some background noise $x(t) = w(t) + s(t)$. The gain G can be adjusted in the sound card.

   If your mouth is far from mic, then the proposition of $s(t)$ is large (SNR is small). If your gain G was small, too, then increasing $x[n]$ by factor K increases $s[n]$ as well by factor K.

   If you gain G is small, then values of $x[n]$ do not vary much, e.g. they are around $(-0.05 \ldots 0.05)$ even the dynamic range of $x[n]$ is from -1 to 1, see middle column in Figure 1. If your bits/sample is low, e.g. 8, you have only $2^8 = 256$ quantization levels, that is, $\Delta = 2/256$. Now when later amplifying $K \cdot x[n]$ the quantization becomes visible and audible.

   If you shout into microphone, then some values are saturated and the signal $x[n]$ is clipped ("särö"), see right column in Figure 1.

   All in all, you should adjust your recording and avoid both too low and too high level of mic volume, and you should have your microphone close. In addition, be aware of 50 Hz interference and other types of distortion. They are hard to suppress afterwards.

   Task: Record a word from the table below into a WAV file (number *no* is the last digit in your student ID). Read it into Matlab and check that your recording sounds and looks nice.

no	fonem	word	no	fonem	word
0	/a/	l**aa**tikko	5	/o/	r**oo**malainen
1	/a/	lounast**aa**	6	/u/	s**uu**llinen
2	/e/	s**ee**steinen	7	/y/	s**yy**llinen
3	/e/	am**ee**ba	8	/ä/	p**ää**rynä
4	/i/	v**ii**si	9	/ö/	t**öö**ttäys

Matlab Extras

1. (Bonus **E40** *3 bp*) Read more about DTMF (dual-tone multifrequency) tones. Choose any "DTMF signal" from www-library `http://www.cis.hut.fi/Opinnot/T-61.3010/Harjoitustyo/Q4_S2005` and analyze it with a spectogram and extract the numbers.

2. (Bonus **E41** *3 bp*) Linear-phase FIR filters are divided to four categories (*Mitra 2Ed Sec. 4.4.3, 4.4.4 / 3Ed Sec. 7.3*)

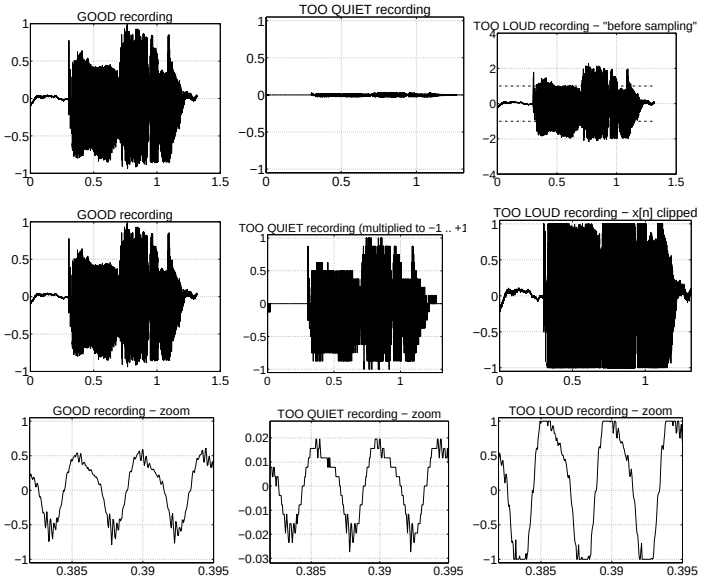

Figure 1: Examples on recording. Left column: mic volume set up correctly. Middle column: mic volume too small or word uttered too quietly. Right column: mic volume too big or word shouted. Dynamic scale for values of sequence $x[n]$ is between -1 and 1. It is useful try to use all the scale.

- Type I, symmetric impulse response with odd length. Order N is even. Either an even number of no zeros at $z = 1$ and $z = -1$.
- Type II, symmetric impulse response with even length. Order N is odd. Either an even number or no zeros at $z = 1$, and an odd number of zeros at $z = -1$.
- Type III, antisymmetric impulse response with odd length. Order N is even. An odd number of zeros at $z = 1$ and $z = -1$.
- Type IV, antisymmetric impulse response with even length. Order N is odd. An odd number of zeros at $z = 1$, and either an even number or no zeros at $z = -1$.

Create an example of each type type of linear-phase filters (Type I, II, III, IV). Start from setting at least 4 zeros in a pole-zero plot, and then form a transfer function from them using `zp2tf`. Check the magnitude and phase response.