T-61.246 Digital Signal Processing and Filtering

Round R1 (v 1.0, 13.9.2004), Matlab I, autumn 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 27th Sep at 12.00**. Return your solutions **on paper** in the returning box of CIS Laboratory in CS building.

Exercises

1. You can use Matlab as a calculator.

   ```
   x = 5
   z = 3 + 4*j
   exp(-j*pi), log10(100), log2(8), atan(4/3), sqrt(25)
   abs(z), angle(z), real(z), imag(z)
   ```

   Task: What can you see in the tab “Workspace” and in the window “Command history”. Type `l` (letter “l” without enter) in the command window and use up/down arrow keys - how does Matlab respond? What does the semicolon “;” in the end of line do?

2. Matlab processes matrices (sizes: matrix MxN, column vector Mx1, row vector 1xN, scalar 1x1). Some notations:

   ```
   y = [3 2 -1 6]
   x = [2 : pi : 12]’    % [start : interval : end]
   y*x
   x*y, size(x*y)        % [#columns #rows]
   y*y
   y.*y, length(y.*y)    % max{#columns, #rows}
   y./x’
   plot(x, y)            % adds a line between the points
   ```

   Task: Solve the following problem using Matlab. Write down all the code in a file with a Matlab editor, which can be opened with `edit`. Save the script file with the file name `t2.m` in your working directory (`Z:\DSP\`). You can run the script from Command window in Matlab with `t2` (without the extension `.m`).

   [**T2**] Examine a complex-valued function

   $$H(\omega) = 2 - e^{-j\omega}$$

   where $\omega \in [0 \ldots \pi] \in \mathbf{R}$.

   a) Compute values of Table 1 with a calculator. Euler: $e^{j\omega} = \cos(\omega) + j\,\sin(\omega)$.

   b) Draw the values at $\omega = \{0,\ \pi/4,\ \ldots,\ \pi\}$ into complex plane $(x,\ y)$. Interpolate smoothly between the points.

   c) Sketch $|H(\omega)|$ as a function of ω. Interpolate.

   d) Skecth $\angle H(\omega)$ as a function of ω. Interpolate.

ω	$x = \mathrm{Real}(H(\omega))$	$y = \mathrm{Imag}(H(\omega))$	$r = \lvert H(\omega)\rvert$	$\theta = \angle H(\omega)$
0				
$\pi/4$				
$\pi/2$				
$3\pi/4$				
π				

Table 1: Problem 2: values of a complex-valued function in rectangular $(x,\ y)$ and polar $(r,\ \theta)$ coordinates.

3. Command for plotting figures is `plot`. It is normally used with two inputs `plot(x,y)`, where `x` is a vector containing values in x-axis and `y` in y-axis. A new figure is opened or activated with `figure` and it can be cleared with `clf`. Several axis in one window can be plotted with `subplot`, and several curves in one axis with `hold on`. It is useful to write down some information in the figure. You can use `title, xlabel, ylabel, grid on` for this purpose.

   ```
   t = [0 : 1/1000 : 0.2]; f = 20;
   y = 2.5 * cos(2*pi*f*t + pi/6);
   plot(t,y);
   grid on; title(’My cosine’); xlabel(’time (s)’); ylabel(’amplitude’);
   ```

   Task: Create a figure No. 42 with the cosine of the previous example and a discrete-time cosine ($A = 2.5,\ f = 1020,\ \theta = \pi/6$) plotted with black circles in the same axis (`help plot`). The interesting result is that discrete-time signal values with $f = 1020$ Hz seem to lie on the curve of a continuous-time cosine with $f = 20$ Hz. How do you explain the result?

   You can print the figures directly from Matlab or save them into a file. The command `print -dmeta myfig.emf` prints the current figure in an appropriate Windows format, and `print -deps myfig.eps` is a format “Encapsulated PostScript” for a LaTeX document.

4. You can add control sequences with commands `if, else, elseif, for, while`, etc.

   Let us examine now an imaginary time series (from stock market, temperatures in Kaisaniemi, energy consumption in Finland, etc). The data points may have a trend but there are also components that can be considered as noise. A normal procedure to get “a big picture” is to take averages over a certain time window. The simpliest average is to add two adjacent values and divide the result by two (Moving Average 2):

   $$y[n] = 0.5 \cdot (x[n] + x[n-1])$$

   It can be shown that the corresponding impulse response and frequency response are

   $$\begin{aligned} h[n] &= 0.5 \cdot (\delta[n] + \delta[n-1]) \\ H(e^{j\omega}) &= \sum_k h[k] e^{-jk\omega} = 0.5 \cdot (1 + e^{-j\omega}) \end{aligned}$$

Task: Draw a flow/block diagram of the filter. Write down the code which implements MA-2 for the input. How does the output look like compared to the input? How can you get a smoother output? What can be said about frequency-domain properties of MA-2, in other words, **draw** $|H(e^{j\omega})|$ in range $[0 \ldots \pi]$. Is the filter lowpass / highpass / bandpass / bandstop / allpass?

```
function prob4
% prob4 -- Problem 4, tehtävä 4
% generate the data
t = [1999 : 1/365 : 2002];
x = 5*cos(2*pi*1*t) + 0.2*randn(1,length(t)) + 40;

figure(1); clf;    % open Figure No 1, and clear it.
plot(t, x); grid on;
xlabel(’time (years)’), ylabel(’GWh’); title(’Energy consumption’);

% write down the code for MA-2, and draw the filtered signal in red
for k=1:length(x)
  y(k) = 0.5 * (x(k) + x(k-1));
end;
hold on;
plot(t, y, ’r’);   % ’r’ == color red

% write down the code for frequency-domain analysis
w = [0 : pi/64 : pi]; H = 0.5 * (1 + exp(-j*w));
plot(w, ???);      %  we want to plot |H(e^jw)| in function of w
grid on; title(’Amplitude response’);
```

Bonus problems

B1) Load a sequence into Matlab and listen to that by typing the commands:

```
load mtlb; x = mtlb;
soundsc(x, Fs);  % a female ‘‘Matlab’’
Fs
size(x)
```

Run the code above and below, and reply in your report: What is the sampling frequency? How long (in seconds) is the voice sample? How many data samples are there in 0.1 seconds? What is the correct value for N?

Let us add an echo effect. When giving an impulse, we would like to hear echoes after 0.1, 0.2, 0.3, and 0.4 seconds with weights 0.7, 0.4, 0.2, and 0.1, respectively. It can be coded in various ways. We shall consider the word “Matlab” as an input $x[n]$ to a LTI system with the impulse response $h[n]$. The convolution result $y[n] = h[n] \circledast x[n]$ is the output (echo voice).

$$\text{Impulse } \delta[n] \longrightarrow \boxed{\begin{array}{c}\text{Discrete-time}\\ \text{(LTI) system } h[n]\end{array}} \longrightarrow \text{Impulse response } h[n]$$

$$\text{Input sequence } x[n] \longrightarrow \boxed{\begin{array}{c}\text{Discrete-time}\\ \text{LTI system } h[n]\end{array}} \longrightarrow \text{Output sequence } y[n]$$

```
t = [0 : length(x)-1]/Fs;
plot(t,x); grid on; title(’A female /matlab/’); xlabel(’time (s)’);

N =                 % here add a correct number!
h = [1; zeros(N, 1); 0.7; zeros(N, 1); 0.4; zeros(N, 1); ...
     0.2; zeros(N, 1); 0.1];
y = conv(h,x);
pause;              % wait for pressing any key
soundsc(y, Fs);   % a voice with echo!
```

B2) See the demo DTMF which can be opened by `demos` and choosing “Toolboxes”, “Signal Processing Toolbox” and then the demo “DTMF”, or directly `phone`. Click on Sound box and you will hear the tones of the keys. DTMF tones are generated by adding two cosine signals (Mitra, Sec 1.4.2). In the top axis there is the signal as a waveform in the time-domain, whereas in the bottom axis there is the spectrum of the signal. The two frequencies can be read from the position of the peaks of the spectrum.

Bonus Task: Write down the code and a short report: Choose a number, pick up the two frequencies from the spectrum with command `ginput`. The values should be between 650 and 1500 Hz. Using these two frequencies synthesize a tone as a sum of two cosines. Check that your tone is equivalent to the tone from the key!

B3) Write down the code and a short report: Download a demo file `psyka.mat` from the course web page, and open it in Matlab with `load`. Create a program that implements the segmentation of words, i.e. removes the silence between the words. This can be done, for example, by computing the energy of a certain window. What is the “noise” in the silence part?