

T-61.3010 Digital Signal Processing and Filtering

Version 10.00ENG (27-Mar-2009)

T-61.3010 Course assignment 26.3.2009-29.5.2009.

Read these instructions carefully through. Questions and comments can be sent to address t613010@cis.hut.fi! More information and material, Matlab files etc can be found in <http://www.cis.hut.fi/Opinnot/T-61.3010/Harjoitustyo/>.

It is recommended to work in groups of two persons. In Q1 each member has to evaluate sequence of his/her own, in Q2 and Q3 one signal a group is enough.

The work has to be returned EITHER by 27th April 2009 ("soft deadline", two extra points to MTE) OR by 29th May 2009 at 16.00 in **electrical PDF format** into Rubyric system <https://rubyric.cs.hut.fi/submit/72> using Haka/WebLogin authentication.

An accepted course assignment is valid according TKK rules **one year** for final exams **till May 2010** (5/2009, 9/2009, 12/2009, 1/2010, 5/2010). It is **not** valid as such for mid term exams in 2010.

The first page of the returned document has to be a cover page, which includes the following information of the group members: name, student ID, email, study programme. Write the IDs also with BIG LETTERS in the left upper corner.

The document has to be written properly. In order to refer, the document has to contain page numbers and figure titles (captions). Matlab figures have to have correct titles and labels in axis. They are exported to files (PNG, GIF) from Matlab so that they can be embedded in the document. Therefore, use Microsoft Word, L^AT_EX, OpenOffice or corresponding.

The Matlab code has to be returned within the document, as last pages in the document. Save the code as **scripts** or **functions**, e.g. `part1.m`, which can be called in Matlab with `part1`. This makes changes easier. **Do not save Matlab files** into any Internet page. Protect the files from the others - copying is naturally prohibited. Possible co-operation must be declared.

The assessment. The only grade for the work is "accepted". The possible corrections must be done within two weeks.

The minimum resources needed for the work: Matlab 6.0 and Signal Processing Toolbox 5.0 (Release 12) or newer, sound card, **microphone** (Section 1), headphones/loudspeakers. Additional files information and hints in URL <http://www.cis.hut.fi/Opinnot/T-61.3010/Harjoitustyo/>

There are Matlab / assignment receptions during the end of course. It is useful **to read all Matlab exercises in Spring 2009 before starting**.

40389S

61301O

T-61.3010 Digital Signal Processing and Filtering

Version 10.00ENG (26.3.2009)

T-61.3010 Course assignment 26.3.-29.5.2009.

Day of starting :	23.3.2009
Day of returning :	18.4.2009

Student #1	
Name	Matti Meikäläinen
Student number	40389S
Email	matti@hut.fi
Depart. and Year	TLT 3
When final exam or mid term exams done or going to be done?	____.200__

Student #2	
Name	Veijo Vapaamatkustaja
Student number	61301O
Email	veijo@hut.fi
Depart. and Year	Ko N
When final exam or mid term exams done or going to be done?	____.200__

Assistents fill the rest :

Assistant:	Parviainen, Reyhani, Similä
To be corrected and returned by	___ / ___ 200__
Accepted:	___ / ___ 200__

Notes by assistant:

Summary

Learning outcomes, time management, receptions

After assignment you can create LTI filters within Matlab. You have made a program for an automatic pattern recognition.

Recording and analysis in Q1 1 h. Filter design Q2 6 h. DTMF signal detection Q3 9 h. Documentation 4 h. If you are stuck at some point, come to Matlab receptions.

Documentation

Return a report in PDF format to <https://rubyric.cs.hut.fi/submit/72> using Haka/WebLogin authentication.

Q1 Speech signal

Record a sentence “Now it’s time to collect speech.” Draw a spectrogram and show phonemes against spectrogram. See, e.g., Matlab #2.

Q2 Filter comparison: IIR vs FIR

Analyse music/speech sample, write down filter specifications. Create both IIR and FIR filters which suppress noise. Compare filters. See, e.g., Matlab #4 (IIR, FIR).

Audio file for a group http://www.cis.hut.fi/Opinnot/T-61.3010/Harjoitustyo/Q2_K2009/

Q3 DTMF signal detection

Analyse DTMF signal and detect the digit sequence by a Matlab program. See, e.g., Matlab #5 (energy).

Audio file for a group http://www.cis.hut.fi/Opinnot/T-61.3010/Harjoitustyo/Q3_K2009/

Q4 Feedback

Course feedback.

Appendix: Matlab code of your own

Matlab code is inserted in the end of document.

1 Q1 Speech

If doing as a pair, each member reads a sentence and analyses it. Sentence for Finnish students “Nyt on siis aika kerätä puhetta” and for non-Finns “Now it’s time to collect some speech”.

NOTE! Jos olet osallistunut puheenkeräykseen, saat tämän osion parilla napin painalluksella ilman Matlabia: lue erilliset ohjeet.

Q1.1 Record a sentence and save it in WAV format. Check that you have good quality in your signal.

Q1.2 Read the WAV file into Matlab, into a vector x .

Plot the spectrogram (**spectrogram**) and align phonemes (“letters”) with the spectrogram. For instance, show at which time moment /s/ is situated in the spectrogram, etc.

Q1.3 Try to find some voiced parts (often vowels) of the signal, i.e., which have a certain (quasi) fundamental frequency (pitch). What were typical values in your case?

NOTE! Puheenkeräyksessä “Pitch Contour”.

Some commands needed: `wavread`, `spectrogram`, `colormap`, `colorbar`.

2 Q2 Comparison of FIR and IIR filters

You are not allowed to use GUIs like SPTool or FDATool, but command line functions, see Matlab #4. You can use `speksit.m` and `speksitFIR.m` from the course web site.

Download an audio file `signalQ2_X.wav`, where X is your student ID, from http://www.cis.hut.fi/Opinnot/T-61.3010/Harjoitustyo/Q2_K2009/. Read the signal and the value of the sampling frequency into Matlab. There are two audio samples at **different frequency bands**. In other words, two music/speech samples are multiplexed.

Your task is to remove (attenuate) “metallic noise” using a lowpass filter.

In Q2.1 analyse the signal and find out appropriate specifications for filters. They include cut-off frequencies for passband and stopband, and maximum deviations (ripples) at both bands. You may have to iterate this after filtering the listening to signal.

Create two different types of filters. First, create a digital lowpass IIR filter in Q2.2. Second, create a digital lowpass FIR filter in Q2.3. Finally, in Q2.4 you should compare these filters.

Q2.1 Analyse first the signal `signalQ2_X.wav` (listen, spectrum, spectrogram, filtering) and find out the correct specifications for the filters. Try to find a correct cut-off frequency – you may have to iterate a few times.

You have to use the transition band (band between cut-offs) of 500 Hz and the passband ripple is 1 dB. The high-frequency noise should be disappeared and “natural voice” should be as clear as possible. The other specifications you can set by yourself. In order to compare filters **specifications for the filters in Q2.2 and Q2.3 have to be the same** (or as similar as possible).

Draw the figure with specifications (e.g. `speksit.m`). Explain briefly why you chose the specifications as you did.

Q2.2 Compute a **lowpass IIR** filter. Digital IIR filters are often derived from analog counterparts using bilinear transform. First, one estimates the required order of $H(z)$, and then computes coefficients of $H(z)$. Choose one of the following IIR function pairs: Butterworth (`butter` and `butter`), Chebyshev I (`cheb1ord` and `cheby1`), Chebyshev II (`cheb2ord` and `cheby2`), Elliptic (`ellipord` and `ellip`).

Compute coefficients of $H(z)$. Write down the order of the lowpass filter. Plot the magnitude response and check that it fulfills the specifications.

Apply your filter into the voice file (`filter`). “Metallic” noise should be attenuated greatly.

Q2.3 Compute a **lowpass FIR** filter using as similar specifications as possible in Q2.2 (Q2.1). Digital FIR filters do not have any analog counterparts. They can be implemented using, e.g., window method (`fir1`), frequency sampling (`fir2`), least-square method (`firls`), Parks-McClellan (`firpmord` and `firpm`), or any other.

Write down the order of the filter. Compute the coefficients of the filter.

Check, if the phase response is linear or not (most algorithms provide linear-phase filters). Check that the specifications are fulfilled (mostly) – if not, determine new order or cut-off frequency. Filter the signal also with FIR filter.

Q2.4 Compare the filters in Q2.2 (IIR) and Q2.3 (FIR). Compare the orders of the filters.

Plot a short part (e.g. 20 ms in the middle of the signal) of the original, IIR filtered and FIR filtered signals into the same figure in order to show how much signals have been delayed.

In case of linear-phase FIR in Q2.3, confirm that the signal is delayed by $M = N/2$ samples, or M/f_s seconds, where N is the order of the filter.

Q2.5 **OPTIONAL, Not compulsory part, “just for fun”**. “High-frequency noise” is, in fact, music or speech, if it is demodulated (AM) and filtered. What does it contain? See, e.g., MTE2-exercises (VK2-pistelaskarit), problem 3.

3 Q3 DTMF signals

Explain briefly what are DTMF (dual-tone multifrequency) tones.

Download your own DTMF-tone `dtmf_X.wav`, (X is your student ID) from http://www.cis.hut.fi/Opinnot/T-61.3010/Harjoitustyo/Q3_K2009/. Here each tone is at least 70 ms long and the break between tones at least 40 ms.

Write down the plan for the program, with which you can automatically detect the numbers from DTMF-signal.

Implement the program and detect your own signal automatically. Write down the encoded sequence of numbers in the document.

4 Q4 Feedback

Q4.1 Which computer, operation system and Matlab version (`ver`)?

Q4.2 Any practical problems with computers?

Q4.3 What kind of help did you receive from other students or personnel? Co-operation? If groups have collaboration, it must be mentioned here.

Q4.4 Too much or too little instructions?

Q4.5 Did you find any help from the Matlab exercises? Did you find help from paper exercises? Did you take part in receptions for the project work?

Q4.6 What was the most interesting part of the work?

Q4.7 How much time did you use for the work?

Q4.8 Other comments?

Remarks

- Re-read the instructions on page 1.
- Notice that if the document is going to be printed by grayscale printer, you should not use colors to differentiate signals or other plots. Instead you can use grayscale values (`colormap(gray)`, `colorbar`), or different line style types (`plot(x, y, S)`, where $S = \{ '-', ' - ', ':', '-. ' \}$ meaning solid, dashed, dotted, and dash-dotted line. Use also command `legend({'exp11', 'exp12'})` to clarify the figure.
- Q2: After you have created a filter, you should always check if specifications are met, see [P66] in the example material. You should always choose the lowest possible filter order which fulfills specifications.
- Q2.3: In case of `fir1` you can estimate a sufficient filter order using Table 1 (*Mitra 2Ed Table 7.2 / 3Ed Table 10.2*). After creating the filter, check specifications, and iterate the order if needed.

Example: Using rectangular window the “transition band” is $\Delta\omega = |\omega_p - \omega_s| = 0.1\pi$. Then $M = 0.92\pi/0.1\pi \approx 9$ and the filter order $N = 2M = 18$. Cut-off frequency (-6 dB) is defined $\omega_c = (\omega_p + \omega_s)/2$.

Window	Matlab	Length of main lobe Δ_{ML}	Relative side lobe A_{sl}	Minimum stopband attenuation	Length of transition band $\Delta\omega$
Rectangular	<code>rectwin</code>	$4\pi/(2M+1)$	13.3 dB	20.9 dB	$0.92\pi/M$
Hann	<code>hann</code>	$8\pi/(2M+1)$	31.5 dB	43.9 dB	$3.11\pi/M$
Hamming	<code>hamming</code>	$8\pi/(2M+1)$	42.7 dB	54.5 dB	$3.32\pi/M$
Blackman	<code>blackman</code>	$12\pi/(2M+1)$	58.1 dB	75.3 dB	$5.56\pi/M$

Table 1: Some properties of window functions.

- Q2.4: Draw the magnitude responses of the filters into the same figure with specifications, e.g., `plot(w, abs(H1), '- ', w, abs(H2), '-.')`, where H1 and H2 are the frequency responses from `freqz`.
- Q2.4: Do not take the interval of 20 ms in the beginning of the file because most samples are zero because of fade-in.
- Q3: Mitra's book (*Mitra 2Ed Sec. 11.1 / 3Ed Sec. 15.1*) utilises Goertzel's method in case of DTMF recognition but ordinary DFT `fft` is also enough. Goertzel computes DFT recursively only at certain points whereas FFT is computed at once for all samples. The exact algorithm for computing DFT is not relevant in this case.
- Q3: There is some noise in the tone signal. Therefore you have to have a routine to judge whether a window frame contains a DTMF-tone or not, and the other routine to encode the tone.