

T-61.3010 Digitaalinen signaalinkäsittely ja suodatus

T-61.3010 Harjoitustyö 26.3.2009-29.5.2009. / Versio 10.00 (26.3.2009)

Lue nämä ohjeet huolellisesti läpi. Kysymyksiä ja kommentteja voi lähettää osoitteeseen t613010@cis.hut.fi Lisätietoja, muu materiaali, Matlab-tiedostoja ja mahdolliset korjaukset löytyvät osoitteesta <http://www.cis.hut.fi/Opinnot/T-61.3010/Harjoitustyö/>.

Harjoitustyö tehdään mieluiten 2 hengen ryhmässä. Paria voi etsiä esimerkiksi nyssiryhmässä opinnot.tik.t613010. Tarvittaessa myös yksilötyö on mahdollinen. Kohdassa Q1 jokaisen tulee luoda omat signaalinsa, kohdissa Q2 ja Q3 riittää yksi per ryhmä.

Työhön arvioitu aika 20 tuntia.

Palautus joko viimeistään 27.4.2009 ("soft deadline", 2 lisäpistettä välikoesuoritukseen) tai perjantaihin 29.5.2009 klo 16.00 mennessä **sähköisesti PDF-muotoisena** Rubyric-järjestelmään <https://rubyric.cs.hut.fi/submit/72> käyttäen Haka/WebLogin-tunnistautumista.

Hyväksytty harjoitustyö kelpaa tutkintosaännön mukaisesti kurssin **osasuorituksena yhden vuoden ajan** eli kevään 2010 tenttiin asti (5/2009, 6/2009, 9/2009, 12/2009, 1/2010, 5/2010). Tämä hyväksytty harjoitustyö **ei kelpaa** sellaisenaan muiden välikokeiden kuin kevään 2009 välikokeiden kanssa.

Voit vastata suomeksi / Du kan svara på svenska / You can write in English. Käytä samaa kieltä läpi koko dokumentin.

Palautettavan dokumentin ensimmäinen sivu tulee olla päällyslehti, josta käy ilmi ryhmän jäsenien nimet, opiskelijanumero, sähköposti, koulutusohjelma, ja jonka vasemmassa yläreunassa tulee olla suurella lueteltuna opiskelijanumerot.

Dokumentin tulee olla **ytimekäs ja kattava: lukijan on voitava ymmärtää työn kulku ilman tehtävänantoa**. Viittamista varten dokumentissa pitää olla sivu- ja kuvanumerot. Matlab-kuvissa tulee olla tarvittavat otsikot, esimerkiksi onko x-akselilla indeksinumeroita vai sekunteja. Matlab-kuvat kannattaa ensin tallettaa sopivaksi tiedostoksi (PNG, GIF) ja liittää sitten dokumenttiin. Täten dokumentin laatimiseen tulee käyttää joko Microsoft Wordiä, L^AT_EX_A, OpenOfficea tai vastaavaa. Dokumentin valmis L^AT_EX-pohja on saatavilla kurssin kotisivulta.

Talleta kussakin tehtäväosiossa/tehtävässä/funktiossa käytetty **Matlab-koodi skripti-** tai/**ja funktiotiedostoihin**. Esim. skriptitiedosto `osio1.m` ajetaan komennolla `osio1`. Näin nopeutat pienten muutosten tekemistä ja helpotat assistentteja mahdollisten virheiden selvittämisessä. Esitettyihin tuloksiin johtanut koodi pitää palauttaa dokumentin lopussa kokonaisina tiedostoina ("insert file"). Matlab-tiedostoja **ei saa** laittaa Internet-sivuille eikä niitä lähetetä sähköpostilla tms. Suojaa tiedostosi muilta – kopiointi on luonnollisesti kiellettyä. Mahdollisesta yhteistyöstä on ilmoitettava selkeästi.

Työn arvostelu. Työ arvostellaan joko "hyväksytty" tai "korjattava". Bumerangi tulee korjata kahden viikon kuluessa ilmoituksesta.

Työn suorittamiseen tarvittavat vähimmäistietokoneressurssit: Matlab 6.0 ja Signal Processing Toolbox 5.0 tai uudemmat, äänikortti, mikrofoni, kuulokkeet/kaiuttimet.

Ohjausta harjoitustyön tekemiseen järjestetään kootusti kevään aikana "Matlab-vastaanoitoilla" – seuraa Noppaa Samoin **kevään 2009 Matlab-harjoitukset kannattaa lukea läpi ennen työn aloittamista**.

40389S

613010

T-61.3010 Digitaalinen signaalinkäsittely ja suodatus

Versio 10.00 (26.3.2009)

T-61.3010 Harjoitustyö 26.3.-29.5.2009.

Aloituspäivä :	23.3.2009
Jättöpäivä :	18.4.2009

Opiskelija #1	
Nimi	Matti Meikäläinen
Opiskelijanumero	40389S
Sähköposti	matti@hut.fi
Tutkinto-ohjelma ja vsk	TLT 3
Milloin tentti tai välikoe on tehty tai aiotaan tehdä?	____.200__

Opiskelija #2	
Nimi	Veijo Vapaamatkustaja
Opiskelijanumero	613010
Sähköposti	veijo@hut.fi
Tutkinto-ohjelma ja vsk	Ko N
Milloin tentti tai välikoe on tehty tai aiotaan tehdä?	____.200__

Assistentit täyttävät loput :

Tarkastaja:	Parviainen, Reyhani
Palautettava korjattuna	__ / __ 200__
Hyväksytty:	__ / __ 200__

Assistentin kirjaamia huomioita:

Tiivistelmä harjoitustyöstä

Tämä on suomenkielinen tiivistelmä harjoitustyön tehtävistä. Tehtävien tarkennukset englanninkielisessä tehtävänannossa seuraavilla sivuilla.

Osaamistavoitteet, ajankäyttö ja työn ohjaus

Harjoitustyön jälkeen osaat luoda LTI-suotimen Matlab-ohjelmistolla. Lisäksi olet harjaantunut Matlabin käyttöön ohjelmoimalla automaattisen hahmontunnistustyökalun.

Nauhoittaminen ja osio Q1 1 h ohjatussa tilanteessa. Suotimen suunnittelu Q2 6 h. DTMF-signaalin selvittäminen Q3 9 h. Dokumentointi 4 h. Työhön varattu aika on 20 tuntia, jona aikana sinulla ja ryhmälläsi pitäisi olla mahdollisuus ratkaista tehtävät oppien samalla. Jos työ näyttää jämähtävän, tule Matlab-vastaanoille. Aika-arvio painuu pienemmäksi Matlab-(ohjelmointi)kokemuksen määrän suhteessa.

Harjoitustyötä ja \LaTeX :n käyttöä esitellään "Matlab-vastaanoilla", jotka löytyvät WebOodista ja Nopasta. Ensimmäiset vastaannot puheenkeräyksen yhteydessä 26.-27.3.2009.

Työn raportointi

Työstä tehdään raportti, joka palautetaan PDF-muotoisena sähköiseen Rubyric-järjestelmään <https://rubyric.cs.hut.fi/submit/72> käyttäen Haka/WebLogin-tunnistautumista.

Q1 Puheäänien taltiointi

Äänitä lause "Nyt on siis aika kerätä puhetta." Piirrä siitä spektrogrammi ja osoita kuvasta, missä kohtaa aika-akselia kukin foneemi ("kirjain") spektrogrammia on.

Jos osallistut puheenkeräykseen, saat spektrogrammin varsin automaattisesti, katso erilliset ohjeet.

Q2 Suodinvertailu: IIR vs FIR

Analysoi annettua musiikkisignaalia ja luo tämän perusteella suotimen vaatimusmäärittelyt (speksit). Luo speksit täyttävät IIR- ja FIR-alipäästösuotimet ja suodata häiriöääni pois. Vertaile suotimia. Katso mm. Matlab #4 (IIR, FIR).

Ryhmän oma äänitiedosto: http://www.cis.hut.fi/Opinnot/T-61.3010/Harjoitustyö/Q2_K2009/

Q3 DTMF-signaalin selvittäminen

Analysoi annettua DTMF-signaalia ja etsi siitä oikea numerosarja ohjelmallisesti. Katso mm. Matlab #5 (energia).

Ryhmän oma äänitiedosto: http://www.cis.hut.fi/Opinnot/T-61.3010/Harjoitustyö/Q3_K2009/

Q4 Palaute

Palaute tähän työhön liittyen.

Liite: Työssä käytetty koodi

Liitetään mukaan tekstinä dokumentin loppuun, ei WWW-sivulle.

VERSIOT

v 10.00 – julkaistu 2009-03-26.

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/Harjoitustyö/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 (**buttord** 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/Harjoitustyö/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.