↖ staple all papers together in the end

return: 23.3. 2020, 14.4. 2020, 4.5. 2020, 25.5. 2020, **591111**

ELEC-C5230 Digital Signal Processing Basics

The course may be passed by returning these homework assignments and passing compulsory check-up exam. Please, circle the return date above.

Student number:	591111
Name:	
E-mail:	
Signature:	
Target grade:	
Possible co-operation:	
WWW:	`https://mycourses.aalto.fi/course/view.php?id=24631`

Grading table

Exercise no.											TOT
Student points*											
TA points											

*** self-assessment; fill in the first two rows.**

Instructions

- **COMPULSORY TASK**: print this page and the list of problems **separately on A4 paper**. If the cover page is missing, total points will be decreased by 1 point.
- use ruled or squared A4 and write the answers **by hand or computer** in a readable and sufficiently large font; solution to a new problem should start from a new page; write your student number and problem number on each paper; leave some space at the margins
- fill this **front page** and staple together with your answers **every time** you return the assignments; use the same order in your answers as given in the list of problems. If the returned pages are in wrong order such that it hinders the assistants' work, total points will be decreased by 1 point.

- write the solutions as you would in an exam; **it is not sufficient to have just the answer**, also derivations with sufficient detail are required; the answer should be understood also without the problem statement
- return all solutions at the same time at given dates to the course return.
- **solutions have to be returned personally**, but it is allowed (even encouraged) to solve them in a group; however, you are not allowed to directly copy the solutions or publish them for others, see for example, Academic Integrity in Aalto University `https://into.aalto.fi/display/ensaannot/Aalto+University+Code+of+Academic+Integrity+and+Handling+Violations+Thereof`
- help can be found from the lecture slides, course book and additional material given in MyCourses; you may also ask advice at the weekly exercises and MyCourses discussion group
- course e-mail address is `ELEC-C5230@aalto.fi`

General Information

MyCourses pages for the course are `https://mycourses.aalto.fi/course/view.php?id=24631`. Course requirement is solving sufficient amount of homework assignments (see the table below) and taking part in a "check-up exam" (1-2h) during May 2020 (see MyCourses for a schedule).

The homework problems are divided into three categories: 100-, 200- and 300-series. Solving the 100-series entirely is the minimum requirement for passing the course. The solution deadlines are 23.3.2020, 14.4.2020, 4.5.2020 and 25.5.2020.

There are two alternative weekly exercise sessions, which are held on Tuesdays at 12-14 and on Thursday at 10-12. Sessions on Tuesdays are held in English. During each session, some of the assignments will be solved and there is also possible for ask about the assignments. Participating in the sessions is voluntary, but highly recommended.

Additionally, course can be completed with project work and passing an exam. However, this is not the preferable way. Further instructions can be found in MyCourses.

Grading

Course can be passed by returning the homework assignments and participating in the check-up exam. This is the preferable way to complete the course. **In order to pass, all 100-series assignments have to be returned**. No additional points are given for the solutions of 100-series. Final grade depends on which assignments are returned and "how well" they are solved. The work is considered to be unsatisfactory if the solutions are incorrect, incomplete, not returned in time or solved mainly with the help of the course assistants.

At the beginning, you need to **set your own goal for the course grade.** The self-desired grade can be altered during the course. During the check-up exam in May, the grade goal and accomplished work will be evaluated. It is recommended that students who have signal processing in some form as major or minor studies set the goal at least to 3. Note that if the desired and evaluated grade do not match, it is possible to request additional assignments to solve in order to improve the evaluated grade. Negative total points (due to penalty) must be compensated by earning points from the 200- and 300-series in order to pass.

The following table describes roughly how grading is performed. Note that the Assignment-column is cumulative.

Grade	Description	Assignment	Initial threshold
0	minimum requirements are not met	all 100-series assignments have not been solved satisfyingly	
0 → 1	minimum requirements met after required corrections	additional assignments are returned within two weeks of the check-up exam	0
1	minimum requirements: analysis of LTI systems and time-frequency analysis of digital signals, basic Matlab use for DSP	all 100-series problems have been solved in a satisfying way	0
2	minimum requirements met well	100-series assignments solved + additional 200-series assignments	12 p
3	basic level: course goals met well	200-series solved well (at least 3/4 of the points in 200-series)	24 p
4	own interest on subject is clearly showing	in addition some 300-series assignments	38 p
5	outstanding achievement	well solved 300-series assignments	52 p

Assignments have been divided in three categories that reflect the final grade goal: (1) Solve 100-series **basic assignments**, if your grade goal is 1–2; (2) If grade 2–3 is desired, solve some additional 200-series **follow-up assignments**; (3) To achieve grades 3–5, return (some) 300-series **advanced assignments** as well.

How to Return the Homework Assignments

Print and fill the cover page every time you return the paper homework solutions. If cover page is missing, total points will be decreased by 1 point. Keep track of the assignments you have solved and returned.

Deadlines:

- Paper return I: Mon 23.3.2020 at 12.15. Assignments marked as: (*DL 23.3.* **[code]**)
- Paper return II: Tue 14.4.2020 at 12.15. Assignments marked as: (*DL 14.4.* **[code]**)
- Paper return III: Mon 4.5.2020 at 12.15. Assignments marked as: (*DL 4.5.* **[code]**)
- Paper return IV: Mon 25.5.2020 at 12.15. Assignments marked as: (*DL 25.5.* **[code]**)

There are several ways to return your solutions. They can be returned at weekly exercise sessions or during lectures. There is also a return box titled ELEC-C5230 located on the third floor of the T-building (the exact place will be announced in MyCourses).

In addition, the paper assignments can also be returned electronically in MyCourses. Read the instructions (in MyCourses) carefully before uploading any files!

Some exercises have a hint box. Marking **[Txx]** or **[Pxx]** refers to exercise xx in the example material that is available in MyCourses at page "Additional material".

The number of homework problems is large — so you should start solving them well in advance before the deadlines, especially if the grade goal is higher. The cumulative workload for the course has been thought to be: (a) Lectures and paper exercises, where some 100-series assignments are present, 2 cr. (b) Own preparations and solutions, 2 cr. (c) Check-up exam, Matlab-exercises etc., 1 cr.

It is impossible for us to check all solutions (70 students and 2 teaching assistants). Regrettably, detailed feedback cannot therefore be given. If you wonder whether you have solved some problem correctly, you may discuss it with your course mates or try the solution, for example, with Matlab. Ask about the assignments from TA's at the weekly exercises. You may also discuss, ask questions and answer questions posed by others in **MyCourses**. Enroll to the course in Oodi and log in at `http://mycourses.aalto.fi`.

Check-up Exam in May 2020

Group "check-up" exam in May 2020 presents a possibility for correcting some technical and logical mistakes that have been noticed. The goal of this "exam" is also to verify that the students have understood the problems they have returned (for example, someone else didn't do them). After this session, the final grade is agreed upon. If the grade goal is higher than what the returned assignments would give, the student is possible to get additional problems to solve that will yield the desired grade. The check-up exam will take around 1–2 hours and it is done with mobile devices. More information can be found in MyCourses.

Instructions for Matlab exercises

The Matlab assignments do not always explicitly and clearly specify what you should return. So, in principle, you need to document and return *everything what you have done and learnt*: Answer to all questions in the assignments (even to the rhetorical ones), present solid evidence that you have done and understood all the steps stated in the assignments, i.e., copy&paste and print everything you have produced (all codes, commands, outputs and plots), and collect the material together.

Especially, the Matlab commands `publish` and `diary` are useful for reporting the solutions.

However, please use your common sense: you do not need to print the same code multiple times, e.g., if you only change some variable value, and multiple plots can be combined on each print-out or, even, each figure. Furthermore, *you are allowed to use Matlab in all assignments unless mentioned otherwise.*

Return I, DL 23.3.2020

Table 1: Bookkeeping. Check the instructions to see the information about the points and grades.

Nro	min – max points	DL	Actual time used (h)	Points (23.3.)	Points (14.4.)	Points (4.5.)	Points (25.5.)
101	0	23.3.2020		*	*	*	*
102	0	23.3.2020		*			
103	0	23.3.2020		*			
104	0	23.3.2020		*			
105	0	23.3.2020		*			
106	0	23.3.2020		*			
107	0	23.3.2020		*			
108	0	23.3.2020		*			
109	0	23.3.2020		*			
110	0	23.3.2020		*			
111	0	23.3.2020		*			
112	0	23.3.2020		*			
113	0	23.3.2020		*			
201	0–1	23.3.2020		*			
202	0–1	23.3.2020		*			
203	0–1	23.3.2020		*			
204	0–1	23.3.2020		*			
205	0–1	23.3.2020		*			
206	0–1	23.3.2020		*			
207	0–1	23.3.2020		*			
TOT.							

Return II, DL 14.4.2020

Table 2: Bookkeeping. Check the instructions to see the information about the points and grades.

Nro	min – max points	DL	Actual time used (h)	Points (23.3.)	Points (14.4.)	Points (4.5.)	Points (25.5.)
114	0	14.4.2020		*	*	*	*
115	0	14.4.2020			*		
116	0	14.4.2020			*		
117	0	14.4.2020			*		
118	0	14.4.2020			*		
119	0	14.4.2020			*		
120	0	14.4.2020			*		
121	0	14.4.2020			*		
208	0–1	14.4.2020			*		
209	0–2	14.4.2020			*		
210	0–1	14.4.2020			*		
211	0–2	14.4.2020			*		
212	0–1	14.4.2020			*		
213	0–3	14.4.2020			*		
301	0–1	14.4.2020			*		
302	0–3	14.4.2020			*		
303	0–1	14.4.2020			*		
TOT.							

Return III, DL 4.5.2020

Table 3: Bookkeeping. Check the instructions to see the information about the points and grades.

Nro	min – max points	DL	Actual time used (h)	Points (23.3.)	Points (14.4.)	Points (4.5.)	Points (25.5.)
122	0	4.5.2020		*	*	*	*
123	0	4.5.2020				*	
124	0	4.5.2020				*	
125	0	4.5.2020				*	
126	0	4.5.2020				*	
127	0	4.5.2020				*	
214	0–1	4.5.2020				*	
215	0–1	4.5.2020				*	
216	0–1	4.5.2020				*	
217	0–1	4.5.2020				*	
218	0–1	4.5.2020				*	
304	0–1	4.5.2020				*	
305	0–1	4.5.2020				*	
306	0–2	4.5.2020				*	
307	0–2	4.5.2020				*	
308	0–6	4.5.2020				*	
309	0–4	4.5.2020				*	
TOT.							

Return IV, DL 24.5.2020

Table 4: Bookkeeping. Check the instructions to see the information about the points and grades.

Nro	min – max points	DL	Actual time used (h)	Points (23.3.)	Points (14.4.)	Points (4.5.)	Points (25.5.)
128	0	25.5.2020		*	*	*	*
129	0	25.5.2020					*
130	0	25.5.2020					*
131	0	25.5.2020					*
132	0	25.5.2020					*
133	0	25.5.2020					*
219	0–1	25.5.2020					*
220	0–1	25.5.2020					*
221	0–1	25.5.2020					*
222	0–4	25.5.2020					*
223	0–2	25.5.2020					*
310	0–1	25.5.2020					*
311	0–2	25.5.2020					*
312	0–3	25.5.2020					*
313	0–4	25.5.2020					*
314	0–1	25.5.2020					*
TOT.							

Assignments

You have to solve all of the 100-series assignments in order to pass the course. You don't get any points by solving them.

Return I, DL 23.3.2020

Basic Assignments, 100-series

101. (*DL 23.3. 2020, compulsory,* [**B9916**])

Do a time-use plan (at the beginning). Reserve time from your calendar to complete the course, log your time usage during the course, and report it during each exercise return. State your current target grade on the cover page. You can use the tables found in the beginning of this booklet to evaluate the target grade.

102. (*DL 23.3. 2020, compulsory,* [**B1571A**]) Consider the following two complex numbers

$$\begin{aligned} z_1 &= 5+2j \\ z_2 &= -2-6j \end{aligned}$$

a) Draw the vectors z_1 and z_2 separately in complex plane.
b) Draw and compute the sum $z_1 + z_2$.
c) Draw and compute the weighted sum $z_1 - 2z_2$.
d) Draw and compute the product $z_1 \cdot z_2$.
e) Compute and reduce the division $z_1/z_2 = a + bj$, i.e., find a and b.

103. (*DL 23.3. 2020, compulsory,* [**L0297**]) Sketch the following sequences around the origin

a) $x_1[n] = \sin(0.1\pi n)$
b) $x_2[n] = \sin(2\pi n)$
c) $x_3[n] = \delta[n-1] + \delta[n] + 2\delta[n+1]$
d) $x_4[n] = \delta[-1] + \delta[0] + 2\delta[1]$
e) $x_5[n] = \mu[n] - \mu[n-4]$
f) $x_6[n] = x_3[-n+1]$

104. (*DL 23.3. 2020, compulsory,* [**M2064**]) A pure sinusoidal sequence can be written as

$$x[n] = A \cdot \cos(\omega n + \theta) \;=\; A \cdot \cos(2\pi(f/f_T)n + \theta)$$

where $\omega = 2\pi f/f_T$ is normalized angular frequency with frequency f and sampling frequency f_T, amplitude A, and phase shift θ. There is an example in Figure 1, where $\omega = 0.2\pi$, $\theta = 0$ and $A = 1$. The sequence can be written with δ functions and plain numbers as

$$\begin{aligned} x[n] &= \ldots + \cos(-0.2\pi)\delta[n+1] + \delta[n] + \cos(0.2\pi)\delta[n-1] + \cos(0.4\pi)\delta[n-2] + \ldots \\ x[n] &\approx \{\ldots, 0.809, \underline{1}, 0.809, 0.309, \ldots\} \end{aligned}$$

where the origin is shown as underlined. The code for generating signal:

```
n     = [-2 : 15];
A     = 1;
theta = 0
omega = 0.2*pi;
x     = A * cos(omega * n + theta);
figure(1); clf;               % open/activate Fig. 1, clean it
stem(n,x);
axis([-1.5 10.5 -1.1 1.1]);   % zoom [xmin xmax ymin ymax]
grid on; xlabel('n'); ylabel('x[n]'); title('Sequence x[n]');
print -dpng myCosineSequence.png  % save into an image file
```

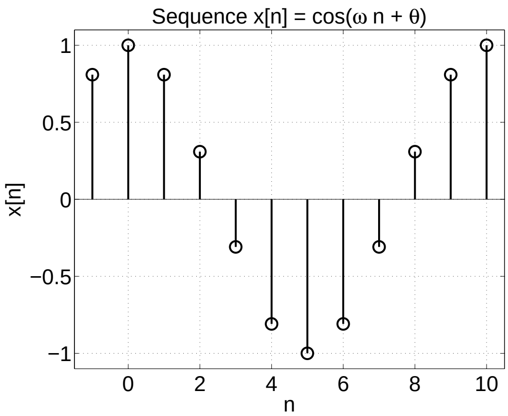


Figure 1: Problem 104: Sequence $x[n] = \cos(0.2\pi n)$. Notice that a period of cosine (2π) includes now 10 samples, i.e., $\omega = 0.2\pi$.

Task: The sampling frequency tells us how many samples there are during one second. (How many values have been sampled in one second.) If the sampling frequency $f_T = 10000$ Hz, (a) how many samples are they in 0.5 seconds: ______ samples; (b) how long would 20000 samples last: ______ seconds.

Task: Generate a pure sinusoidal sequence whose frequency is $f = 440$ Hz, amplitude $A = 2$, length is 0.5 seconds, and sampling frequency is $f_T = 16000$ Hz. Visualize it with `stem` and `plot`, and listen with `soundsc(x, fT)`.

Task: Do the same but now $f = 15560$ Hz.

105. (*DL 23.3. 2020, compulsory,* [**L0162**]) Which of the following signals are periodic? Determine the length of the fundamental period for periodic signals.

   a) $x(t) = 3\cos(\frac{8\pi}{31}t)$
   b) $x[n] = 3\cos(\frac{8\pi}{31}n)$
   c) $x(t) = \cos(\frac{\pi}{8}t^2)$
   d) $x[n] = 2\cos(\frac{\pi}{6}n - \pi/8) + \sin(\frac{\pi}{8}n)$
   e) $x[n] = \{\ldots, \underline{2}, 0, 1, 2, 0, 1, 2, 0, 1, \ldots\}$
   f) $x[n] = \sum_{k=-\infty}^{+\infty} \delta[n-4k] + \delta[n-4k-1]$

106. (*DL 23.3. 2020, compulsory,* [**B9921**]) Sampling.

   a) See the demo about sampling
   using the Matlab script `demosampling4.m`.

Consider a sine signal at frequency 8000 Hz with the sampling frequency $f_T = 10000$ Hz. Because $8000 > 10000/2$ the sequence is aliased into the "baseband" of $0 \ldots 5000$ Hz. Where does the 8000 Hz component alias to? Check the result in Figure 2.

$$
\begin{aligned}
x(t) &= \cos(2\pi \cdot 8000 \cdot t) \\
x[n] = x(nT) = x(n/f_T) &= \cos(2\pi \cdot (8000/10000) \cdot n) \\
&= \cos(2\pi \cdot (8000/10000) \cdot n - 2\pi \cdot (10000/10000) \cdot n) \\
&= \\
&=
\end{aligned}
$$

b) What if the frequency of the analog signal is $f = 13000$ Hz with the same $f_T = 10000$ Hz. What does happen? **(A)** Signal vanishes totally, **(B)** Signal does not alias to "baseband" at all, **(C)** Signal aliases and can be found after ideal reconstruction at 7000 Hz, **(D)** None of above.

c) Sampling with too low sampling frequency can be utilized in some cases, see "Sampling of Bandpass Signals" (*Mitra 2Ed Sec. 5.3, p. 310 / 3Ed Sec. 4.3, p. 184*). Digital signals can also be resampled without any D/A-A/D conversions, see "Multirate processing" (*Mitra 2Ed Sec. 10 / 3Ed Sec. 13*).
Consider a sequence $x[n] = \cos(2\pi \frac{246700}{f_T} n)$, where $f_T > 1000000$ Hz (no aliasing). It is a sinusoidal that oscillates some 250000 times a second. Which sinusoidal sequence (in "baseband" where $\omega \in (0, \ldots, \pi)$ do you get when f_T is set to $f_T \leftarrow 2500$ Hz.

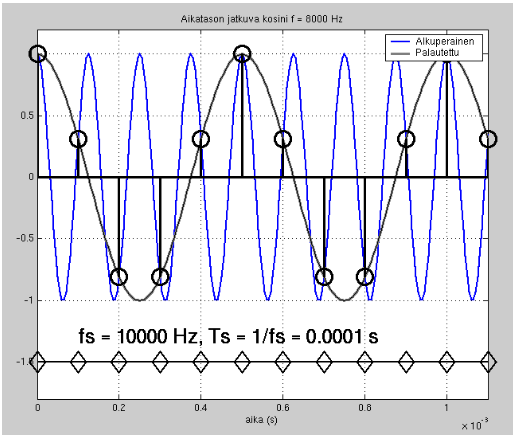


Figure 2: Problem 106: an example of `demosampling4.m` with $f = 8000$ Hz and $f_T = 10000$ Hz.

[T47].

107. (*DL 23.3. 2020, compulsory,* [**L0244**]) Express the input-output relations of the discrete-time systems in Figure 3.

108. (*DL 23.3. 2020, compulsory,* [**B9920**]) Use here the following definition of convolution (not Matlab):

$$y[n] = h[n] \circledast x[n] = \sum_{k=-\infty}^{\infty} h[k]x[n-k]$$

a) Determine convolution for two sequences found in Figure 4.

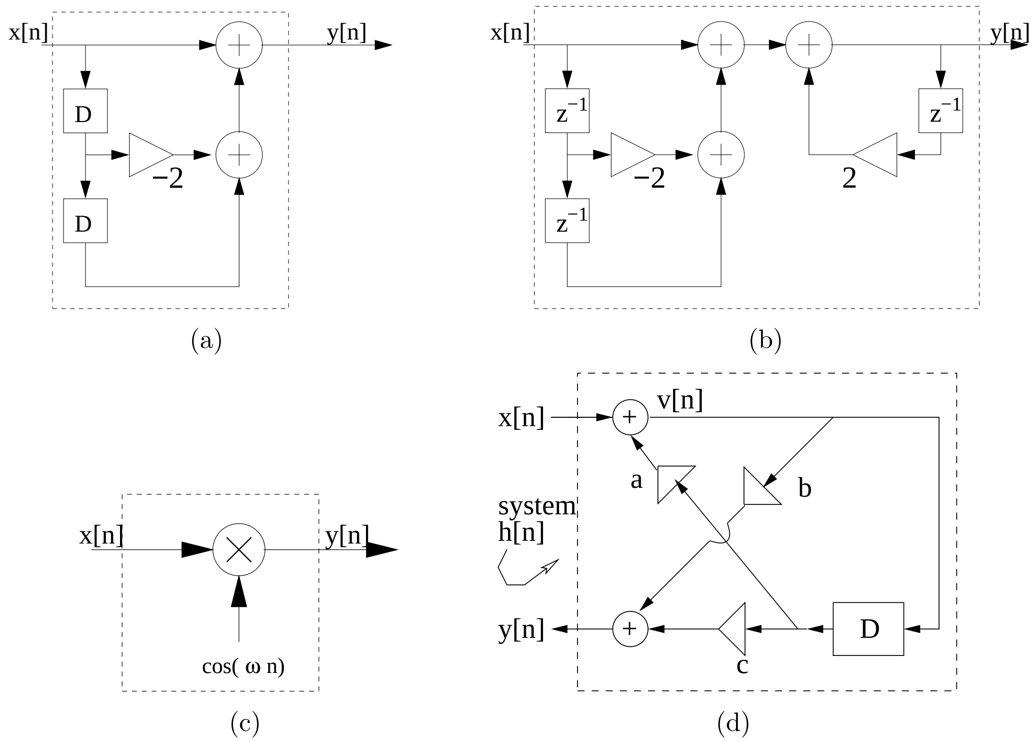


Figure 3: Discrete-time systems for Problems 107 and 204

b) Compute convolution of the input sequence $x[n] = 2\delta[n-2] + 3\delta[n-3]$ and the impulse response $h[n] = -4\delta[n-1] + \delta[n-2]$.

c) Compute the product of polynomials $(3x^3 + 2x^2) \cdot (x^2 - 4x)$.

d) Can you see the connection between (b) and (c)?

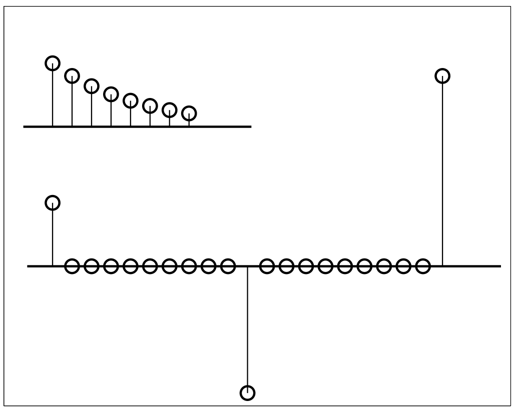


Figure 4: Problem 108(a): two sequences to be convolved.

> **[T30, T31, T32]**. Matlab command `conv`. Try also `http://www.jhu.edu/~signals/` – Joy of Convolution (Discrete Time).

109. (*DL 23.3. 2020, compulsory,* **[L0239]**) Linear convolution of two sequences is defined as (*Mitra 2Ed Eq. 2.64a, p. 72 / 3Ed Eq. 2.73a, p. 79*)

$$y[n] = h[n] \circledast x[n] = x[n] \circledast h[n] = \sum_{k=-\infty}^{\infty} x[k]\, h[n-k]$$

a) Using the above definition of convolution (not Matlab), compute $x[n] \circledast h[n]$, when $x[n] = \delta[n] + \delta[n-1]$, and $h[n] = \delta[n] + \delta[n-1]$.
What is the length of the convolution result?

b) Using the above definition of convolution (not Matlab), compute $x_1[n] \circledast x_2[n]$, when $x_1[n] = \delta[n] + 5\delta[n-1]$, and $x_2[n] = -\delta[n-1] + 2\delta[n-2] - \delta[n-3] - 5\delta[n-4]$.
What is the length of the convolution result? Where does the output sequence start?

c) Using the above definition of convolution (not Matlab), compute $h[n] \circledast x[n]$, when $h[n] = 0.5^n \mu[n]$, and $x[n] = \delta[n] + 2\delta[n-1] - \delta[n-2]$.
What is the length of the convolution result?

110. (*DL 23.3. 2020, compulsory,* **[L0258]**) Compute discrete-time Fourier transform (DTFT) for each of the following sequences using the definition

$$X(e^{j\omega}) = \sum_{n=-\infty}^{\infty} x[n]e^{-j\omega n}$$

a) $x_1[n] = \delta[n-2]$

b) $x_2[n] = 0.5^n \mu[n]$

c) $x_3[n] = a[n] \cdot \cos(\frac{\pi}{5}n)$, where $a[n]$ is a real-valued sequence whose DTFT is $A(e^{j\omega})$.

111. (*DL 23.3. 2020, compulsory,* **[L0282]**) Consult the transform table and find the DTFTs of sequences

a) $x_3[n] = a[n] \cdot \cos(0.2\pi n)$

b) $x_4[n] = 0.25^{n-1}\mu[n-1]$

c)

$$x_5[n] = \begin{cases} 0, & n < -1 \quad \vee \quad n \geq 6 \\ 2, & -1 \leq n < 1 \\ 3, & 1 \leq n < 4 \\ 1, & 4 \leq n < 6 \end{cases}$$

112. (*DL 23.3. 2020, compulsory,* **[M2061]**) Download the file `kiisseli.wav`. In this demo we are analyzing the audio sample in time and frequency domain with the following code.

Task a: What is the sampling frequency of the audio file: ____ Hz. When having `nbits` bits for each value $(-1, \ldots +1)$, how many quantization levels exists: ____. The spectrum in Figure 12 (Matlab code) seems to be symmetric before zooming (`axis`), why so? Why is it not useful to compute DFT over the whole signal?

Task b: Find a "quasi-periodic" part of the signal, e.g. /i/. Create a vector `x_i` of length of 0.05 seconds, plot it, and compute spectrum of the windowed /i/. Show the fundamental period T_0 ____ s, and frequency f_0 ____ Hz, of /i/ in Figures 14 and 16.

Task c: What can you see in the spectrogram, Figure 17? You can also run `specgramdemo(x, fT)` from command line.

```
%% M2061.m Analyzing signal kiisseli.wav

%% TASK a --> Read and listen to the signal
audioinfo('kiisseli.wav')
[x, fT] = audioread('kiisseli.wav');   % download the file!
fT                            % sampling frequency in Hertz
```

```
M = length(x)              % length of x[n]

%% x[n] is nothing but numbers!
x                          % all samples!    % control-c to stop
x(1), x(M)                 % x[0], x[M-1], first and last value
soundsc(x, fT);            % x[n] -> audiocard D/A -> headset
%% Play with numbers I
soundsc(x.*linspace(1,0,M)', fT);  % linearly spaced from 1 to 0
%% Play with numbers II
soundsc(x, 0.6*fT);
%% Play with numbers III
soundsc(flipud(x), fT);    % ud == up-down

%% Plot the signal waveform in time domain
M   = length(x);
t   = [ 0 : M-1 ] / fT;    % time-axis in sec; if index numbers: t=[1:M];
figure(11); clf;           % open/activate Figure No 1, clean it
plot(t, x);
grid on;
xlabel('time (sec)'); title('/kiisseli/');

%% Frequency domain
xF  = fft(x);              % Discrete Fourier transform of signal x, [0,2pi)
MF  = length(xF);          % ... should be M == MF in this case
mag = 20*log10(abs(xF));  % in decibels; if in LINEAR SCALE: mag = abs(xF)
w   = fT * [0 : (MF-1)]/MF; % frequency axis in Hertz [0,fT)
figure(12); clf;
plot(w, mag);              % DFT-spectrum -- usefulness?!?
grid on;
xlabel('frequency (Hz)'); ylabel('dB'); title('DFT of /kiisseli/');
axis([0 fT/2 min(mag) max(mag)]); % zoom only frequencies from 0 to 0.5 fT

%% Window function 'hamming'
figure(13); clf;
stem(hamming(512));

%% TASK b --> Spectrum for /i/
ind1 = 99999;             % ** FIND a correct INDEX NUMBER for /i/
x_i  = x(ind1:ind1+511); % crop a small part of /i/ from /kiisseli/
figure(14); clf;          % ** PLOT signal /i/ in time-domain
x_iw = x_i .* hamming(length(x_i));
figure(15); clf;          % ** PLOT windowed signal /i/ in time-domain
                          % ** COMPUTE spectrum for windowed /i/
figure(16); clf;          % ** PLOT spectrum of win'ed /i/ in frequency-domain

%% TASK c --> Time-frequency-domain: short-time Fourier-transform STFT
figure(17); clf;
spectrogram(x, 512, 256, 512, fT, 'yaxis');   % spectrogram
title('Spectrogram of /kiisseli/');
colorbar;                     % adds a colorbar: color <=> value

%% If you need to print a grayscale document, then change palette
colormap(gray);               % probably better if grayscale printing
colorbar;                     % adds a colorbar: gray value <=> value
```

113. (*DL 23.3. 2020, compulsory,* [**B1800**])

The filtering can be considered as

$$\text{input signal} \quad x \rightarrow \boxed{\text{filter} \quad h} \rightarrow \text{output signal} \quad y$$

Load an audio file `knallipiip.wav` into Matlab. Listen to that and draw the spectrogram. Run the `for` loop (filter) as shown in the code below. After that listen to the output and draw its spectrogram. Note that the coefficients of x sequences in `for` loop are the same as in $H(\omega) = 1 - 1.176e^{-j\omega} + e^{-j2\omega}$.

Return the appended code and spectrograms and give short comments what happened.

```
[x, fT] = audioread('knallipiip.wav');
soundsc(x, fT);
figure(18);
spectrogram(x, 512, 256, 512, fT, 'yaxis'); colorbar;

y = zeros(size(x));
for k = (3 : length(x))
   y(k) = x(k) - 1.176*x(k-1) + x(k-2);
end;
```

Follow-up Assignments: 200-series

201. (*DL 23.3. 2020, 0–1,* [**B9918**])

a) Compute and draw some values of

$$x[n] = e^{-j(0.3\pi n+\pi)} = e^{j(-0.3\pi n-\pi)} = e^{j(-0.3\pi n)} \cdot e^{j(-\pi)}$$

Make sure that you understand how to deal with exponential functions $A \cdot e^{j\omega+\theta}$. Here $A = 1$ (unit circle with radius $r = 1$). Remember to set "RAD" (instead of "DEG") in your calculator if needed. Draw sample values also in Figure 5 (no need to return this figure).

So, sometimes it is more useful to use rectangular coordinates (x, y) and sometimes polar coordinates (r, θ).

n	$x[n]$
0	$e^{-j(0.3\pi\cdot 0+\pi)} \approx$
1	$e^{-j(0.3\pi\cdot 1+\pi)} \approx$
2	$e^{-j(0.3\pi\cdot 2+\pi)} \approx$
3	$e^{-j(0.3\pi\cdot 3+\pi)} \approx$

b) What is the fundamental period N_0 of $x[n] = e^{-j(0.3\pi n+\pi)}$? You can start by substituting n by $n + N$ and keeping in mind that both $n, N \in \mathbb{Z}$. You can also use Figure 5.

$$\begin{aligned} x[n] &= e^{-j(0.3\pi n+\pi)} \\ x[n+N] &= e^{-j(0.3\pi(n+N)+\pi)} \\ &= e^{-j(0.3\pi n+\pi+ \qquad)} \end{aligned}$$

$$\ldots$$

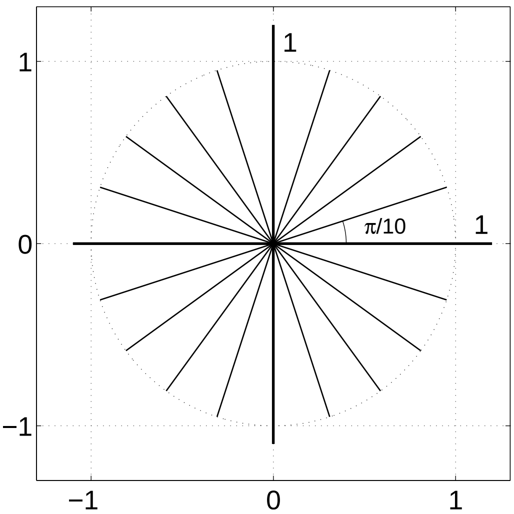


Figure 5: Problem 201: (a) unit circle, angle of each sector is $\pi/10$.

Exponent function: [**T2, T5**] Periodicity: [**T20**]

202. (*DL 23.3. 2020, 0–1,* [**M2062**]) One can hear dual tone multi frequency (DTMF) signal e.g. in traditional phone or mobile phone when number buttons are pressed. DTFM signals are sum of two sine components; lower and higher. $x[n] = \cos(2\pi(f_1/f_T)n) + \cos(2\pi(f_2/f_T)n)$, lower frequencies $\{697, 770, 852, 941\}$ higher frequencies $\{1209, 1336, 1477\}$.

	1209 Hz	1336 Hz	1477 Hz
697 Hz	1	2	3
770 Hz	4	5	6
852 Hz	7	8	9
941 Hz		0	

Implement function that takes phone number as a string and returns DTMF vector.

```
phonenmbr = '050 581 0518'
y         = myGenDTMF(phonenmbr);
soundsc(y, 8000);
plot(y);
```

Check file `myGenDTMF.m` in MyCourses and write the missing rows in `switch-case` structure. Make sure that your code works! Return your source code and spectrogram of your signal. (`spectrogram`).

> There are two alternative options:
> 1 Code everything by yourself. The required function call is `y = myGenDTMF('050 581 0518')`. It must return the required sequence `y`.
> 2 The second option is to check `myGenDTMF.m` with Matlab, and complete the code.
> Listen the results with `soundsc`.
> You may use Matlab switch-case-otherwise structure. See Matlab documentation (`help switch` or `doc switch` or Google `kvg.fi`: Matlab switch).

203. (*DL 23.3. 2020, 0–1,* [**B9919**]) Periods:

$$\exists\, T \in \mathbb{R} : x(t) = x(t+T), \quad \forall t \in \mathbb{R}$$

$$\exists\, N \in \mathbb{N} : x[n] = x[n+N], \quad \forall n \in \mathbb{Z}$$

Consider periodic sequences $x_1[n]$, $x_2[n]$, and $x_3[n]$, which have fundamental periods $N_1 = 6$ (stars), $N_2 = 7$ (circles), and $N_3 = 9$ (crosses), respectively. See an example in Figure 6. What is the fundamental period N_0 of the sum sequence $x[n] = x_1[n] + x_2[n] + x_3[n]$?

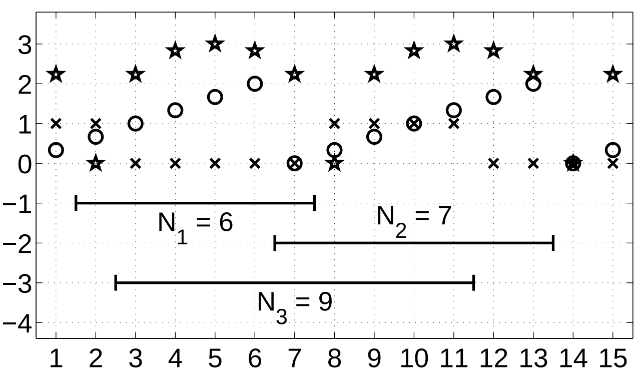


Figure 6: Problem 203: three sequences in Problem 203.

> [**T20d**].

204. (*DL 23.3. 2020, 0–1,* **[L0237]**) Impulse response $h[n]$ is the response of the system to the input $\delta[n]$.

 a) What is the impulse response of the system in Figure 3(a)? What is the connection to the difference equation? Is this LTI system stable/causal?

 b) What are the first five values of impulse response of the system in Figure 3(b)? Hint: Fetch the input $\delta[n]$ and read what comes out. Is it possible to say something about stability or causality of the system?

 c) What are the first five values of impulse response of the system in Figure 3(d)?

205. (*DL 23.3. 2020, 0–1,* **[G1180]**) Consider an analog signal $x(t)$ with band-limited spectrum $|X(j\Omega)|$ shown in Figure 7(a). If the signal is sampled with $f_T = 8$ kHz without anti-aliasing, how does the spectrum $|X(e^{j\omega})|$ look like?

An example of sampling in frequency domain is given in Figure 7(c). Sampling can be also considered as flipping the spectrum around each $f_T/2$-multiple ($f_T/2 = 6$ kHz) and summing the spectrum in the baseband $(0, \ldots, f_T/2)$.

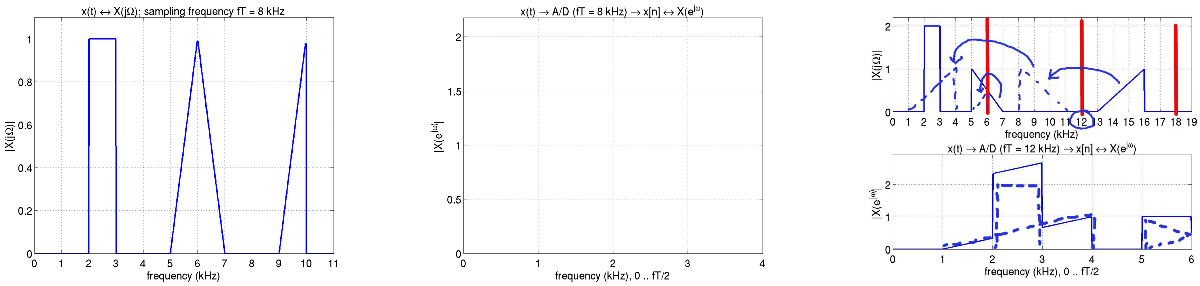


Figure 7: Problem 205: (a) analog spectrum $|X(j\Omega)|$ of Problem with $f_T = 8$ Hz, (b) your answer, (c) an example with $f_T = 12$ kHz.

[T48, T46, T47]

206. (*DL 23.3. 2020, 0–1,* **[L0257]**) Suppose that a real sequence $x[n]$ and its discrete-time Fourier transform (DTFT) $X(e^{j\omega})$ are known. The sampling frequency is f_T. At normalized angular frequency $\omega_c = \pi/4$: $X(e^{j(\pi/4)}) = 3 + 4j$. Determine

 a) $|X(e^{j(\pi/4)})|$

 b) $\angle X(e^{j(\pi/4)})$

 c) $X(e^{j(-\pi/4)})$

 d) $X(e^{j(\pi/4+2\pi)})$

 e) If $f_T = 4000$ Hz, what is f_c

207. (*DL 23.3. 2020, 0–1,* **[M2065]**) A normal procedure to "smoothen" signal is to take averages inside a certain time window. The simplest average is to add two adjacent values (window length 2) and divide the result by two (Moving Average 2, MA-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 a: Draw a flow/block diagram of the filter.

Task b: Write down a Matlab function `myMA2` which implements the MA-2 filter. Do not use here `conv` or `filter` but, for instance, `for` loops. First lines of the function file `myMA2.m` should be

```
function y = myMA2(x)
% MYMA2 computes two-point averaging filter, ''moving average''
% Usage:  [y] = myMA2(x);
```

Task c: Apply your filter `myMA2` to the signal `x`, e.g. `kiisseli.wav`. How does the output look like compared to the input? Listen to the signal before and after filtering. 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?

A script file `M2065.m` for the analysis, task #2:

```
%% M2065 Analysis of MA2 filter

%% Read or create a vector x
[x, fT] = audioread('kiisseli.wav');
n       = [ 1 : length(x) ];
y       = myMA2(x);             % calls function myMA2 with x

%% Signals in time-domain
figure(41); clf;
plot(n, x, 'b', n, y, 'k-.');
xlabel('time indices n');
grid on;
legend({'Original','Filtered'});

%% Filter analysis, see [T4]: H(w) = 2 - exp(-j*w)
w  = [0 : pi/256 : pi];
H  = 0.5 * (1 + exp(-j*w));     % frequency response
r  = abs(H);                    % amplitude response
figure(42); clf;
plot(w, r);
xlabel('norm. angular frequency \omega'); ylabel('|H(e^{j \omega})|');
grid on;

%% Listen to original:
soundsc(x, fT)
%% Listen to filtered:
soundsc(y, fT)
%% Listen to difference:
soundsc(x-y, fT)
```

Task d: Create a longer MA filter in order to get even smoother result.

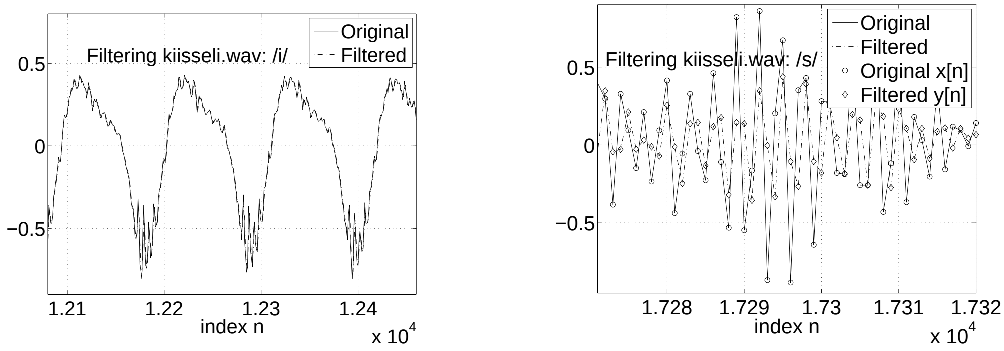


Figure 8: Problem 207: MA-filtering: (a) /i/, (b) /s/.

Return II, DL 14.4.2020

Basic Assignments, 100-series

114. (*DL 14.4. 2020, compulsory,* [**B9916**])

Do a time-use plan (at the beginning). Reserve time from your calendar to complete the course, log your time usage during the course, and report it during each exercise return. State your current target grade on the cover page. You can use the tables found in the beginning of this booklet to evaluate the target grade.

115. (*DL 14.4. 2020, compulsory,* [**M2039**]) Recall that a complex-valued function $H(\omega) = 1 - 1.176e^{-j\omega} + e^{-j2\omega}$ can be computed and visualized in many ways. It was mentioned that frequency responses (frequency-domain) of all LTI filters resemble $H(\omega)$.

Digital (causal) 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 (finite impulse response) and the latter IIR (infinite impulse response). The frequency response is received by substitution $z \leftarrow e^{j\omega}$. 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 [**T55**] we have a second-order LTI system with feedback as shown in Figure 9.

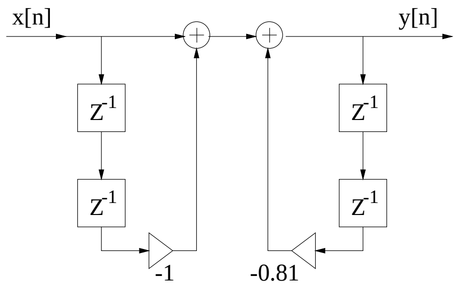


Figure 9: Problem 115: LTI system of Problem [**P55**].

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 at points `w = [0 : pi/512 : pi];` receiving frequency response `H = (1 - exp(-j*2*w)) ./ (1 + 0.81*exp(-j*2*w));` However, the filter can be analyzed using the following commands from Signal Processing Toolbox:

```
%% Analysis of LTI filter
% H(z) = [1 - z^(-2)] / [1 + 0.81 z^(-2)]  = B(z)/A(z)
% B(z) = [1  + 0 z^(-1)  -1 z^(-2)]     % numerator polynomial of H(z)
B      = [1    0          -1];
% A(z) = [1  + 0 z^(-1)  +0.81 z^(-2)]  % denominator polynomial of H(z)
A      = [1    0           0.81];
tf2latex(B, A)       % download from MyCourses
%% Default freqz-plot
figure(1); clf;
freqz(B, A);         % default magnitude (log) and phase response
%% Magnitude response |H(e^jw)|
figure(2); clf;
[H, w] = freqz(B, A);% (w==omega)==pi corresponds f==fT/2 Hz
plot(w, abs(H));     % linear y-axis; OR in decibels (w, 20*log10(abs(H)))
grid on;
xlabel('normalized angular frequency \omega (rad/sample)');
title('Magnitude response');
%% Phase response <H(e^jw)
```

```
figure(3); clf;
plot(w, angle(H));   % phase response
grid on; xlabel('angular frequency \omega (rad)');
title('Phase response');
%% Pole-zero plot
figure(4); clf;
zplane(B, A);        % pole-zero plot
roots(B)             % roots of polynomial B(z) == ``zeros''
roots(A)             % roots of polynomial A(z) == ``poles''
[Z,P,K] = tf2zp(B, A) % zeros, poles and gain.
%% Impulse response h[n]
figure(5); clf;
[h, n] = impz(B, A); % impulse response
stem(n, h);
grid on; xlabel('index n');
title('Impulse response h[n]');
imp2latex(h, n(1:10))   % download from MyCourses,
                        % print first 10 values of h[n]
```

Task: Consider the following LTI systems and classify if they are lowpass / highpass / bandpass / bandstop / allpass / comb / notch filter? What is the order of each filter? Are they FIR or IIR?

a) $H_1(e^{j\omega}) = \frac{1}{1+0.2e^{-j\omega}+1.2e^{-2j\omega}}$

b) $H_2(z) = \frac{0.07+0.1z^{-1}+0.1z^{-2}+0.07z^{-3}}{1-1.4z^{-1}+1.1z^{-2}-0.3z^{-3}}$

c) $H_3(e^{j\omega}) = 1 + e^{-4j\omega}$

d) Poles are at $p = 0.5 \pm 0.5j$ and zeros at $z = e^{\pm j0.25\pi}$. Use either `zp2tf` to find the transfer function $H(z)$, or compute `B` with `conv([1 -exp(j*0.25*pi)],[1 -exp(-j*0.25*pi)])`.

116. (*DL 14.4. 2020, compulsory,* [**M2040**]) Earlier `conv` was used to compute convolution. However, filtering can be efficiently executed using the command `filter`: `y = filter(B, A, x)`, where `B` and `A` are coefficients of the filter $H(z) = B(z)/A(z)$. 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.

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

Task: Consider an elliptic 3rd order filter $H(e^{j\omega})$

$$H(e^{j\omega}) = K \cdot \frac{1 - 2.26e^{-j\omega} + 2.26e^{-2j\omega} - e^{-3j\omega}}{1 - 0.6e^{-j\omega} + 0.72e^{-2j\omega} + 0.1e^{-3j\omega}}$$

Write down vectors `B` and `A`. Apply the filter for the signal `music.wav` using its sampling frequency using the command `filter`. Listen to the original signal and the filtered signal. What is the length of $x[n]$ and $y[n]$? $\mathrm{L}\{x[n]\} =$ ___, $\mathrm{L}\{y[n]\} =$ ___.

Analyze the filter $H(e^{j\omega})$ using the tools in Problem 115. Compute the magnitude response with frequency (Hertz) in x-axis: `[H, F] = freqz(B, A, NDFT2, fT)`, where `NDFT2` is the number of points where DFT is evaluated (default 512) and `fT` is the sampling frequency of the signal (default 2 corresponding $\omega = 2\pi$). Draw the amplitude response in decibel scale and Hertz in x-axis. What is the scaling factor K, so that $\max\{|K \cdot H(e^{j\omega})|\} = 1$ in linear scale or 0 decibels in logarithmic scale, try `max(abs(H))`.

117. (*DL 14.4. 2020, compulsory,* [**L0327**]) Consider the pole-zero plots in Figure 10.

   a) What is the order of each transfer function?
   b) Are they FIR or IIR?
   c) Sketch the amplitude response for each filter.
   d) What could be the transfer function of each filter?

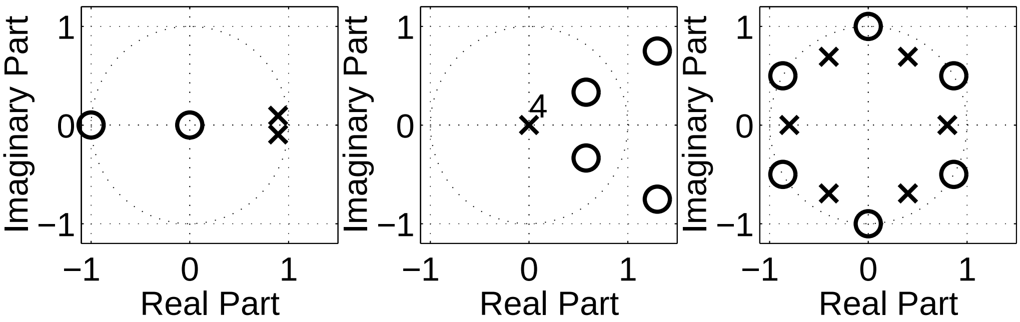


Figure 10: Pole-zero plots of LTI systems in Problem 117.

118. (*DL 14.4. 2020, compulsory,* [**G1400**]) A LTI system is given with its block diagram (direct form I type) in Figure 11. In order to analyze the system in Matlab, you have to find values for `B`, coefficients of numerator polynomial, and `A`, coefficients of denominator polynomial.

   In this problem you are allowed – and you have to use Matlab. Attach all figures in (b)..(e) in returnings.

   a) Determine `B` and `A` for Matlab. Is the system FIR or IIR?
   b) Plot the amplitude response both in linear and logarithmic (decibel) scale (in y-axis).
   c) Plot the phase response. Is the phase response linear?
   d) Plot the pole-zero-plot. Is the filter stable?
   e) Plot the impulse response $h[n]$. Do you find here whether the filter is stable?

119. (*DL 14.4. 2020, compulsory,* [**L0277**]) A LTI filter is characterized by its difference equation

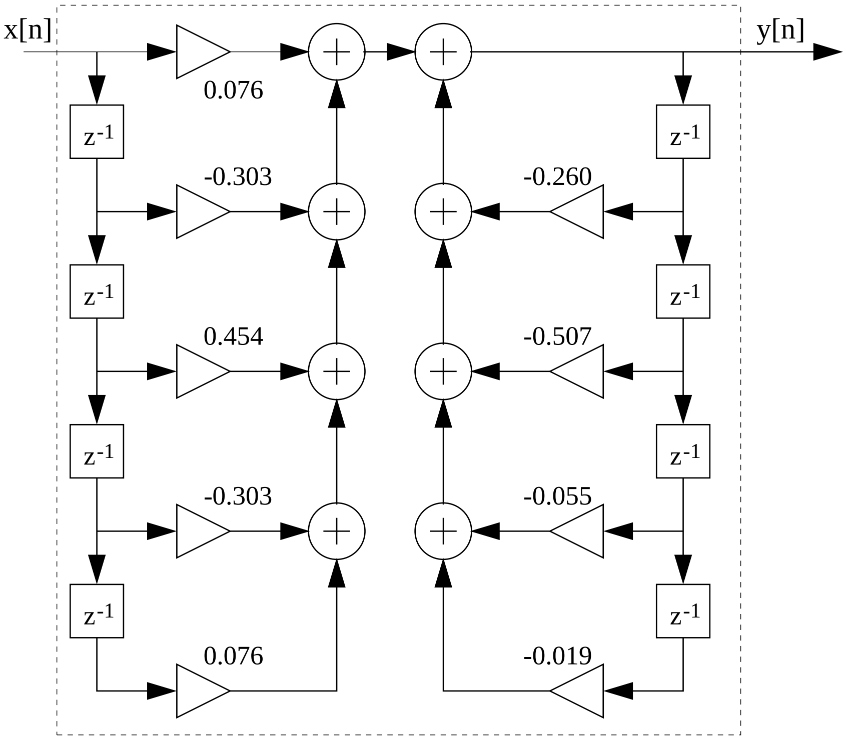
$$y[n] = 0.25x[n] + 0.5x[n-1] + 0.25x[n-2]$$

   a) Draw the block diagram
   b) What is the impulse response $h[n]$
   c) Determine the frequency response $H(e^{j\omega}) = \frac{\sum p_k e^{-j\omega k}}{\sum d_k e^{-j\omega k}}$
   d) Determine the amplitude response $|H(e^{j\omega})|$
   e) Determine the phase response $\angle H(e^{j\omega})$
   f) Determine the group delay $\tau(\omega) = -\frac{\mathrm{d}\angle H(e^{j\omega})}{\mathrm{d}\omega}$

120. (*DL 14.4. 2020, compulsory,* [**B9926**]) Compute by hands a 4-point discrete Fourier transform (DFT)

$$X[k] = \sum_{n=0}^{3} x[n] W_4^{nk}, \quad k = 0, \ldots, 3, \quad W_N = e^{-j2\pi/N}$$

.



.

Figure 11: Problem 118: LTI system from which variables `B` and `A` are to be found.

for a sequence

$$x[n] = 2\delta[n] + \delta[n-2] + 3\delta[n-3] = \{\underline{2},\ 0,\ 1,\ 3\}$$

> See, e.g., **[T13,T50,T51]**.
> If W_4 is tricky, just open it: $W_4 = e^{-j2\pi/4} = e^{-j\pi/2} = -j$ (each W_4 90 degrees clockwise at the unit circle).
> If $\sum_n$ is tricky, just open it one by one, e.g., $k = 0$:
>
> $$X[0] = x[0]W_4^{0\cdot 0} + x[1]W_4^{1\cdot 0} + x[2]W_4^{2\cdot 0} + x[3]W_4^{3\cdot 0} = \ldots$$
>
> You SHOULD CHECK the result by typing `xF = fft([2 0 1 3])` in Matlab. This gives you four values, $X[0]$, $X[1]$, $X[2]$, and $X[3]$. For a real sequence $x[n]$ always: $X[0]$ is real and it is the sum of all $x[n]$; $X[i] = X[N-i]^*$ are complex conjugates; $X[N/2]$ is real.

121. (*DL 14.4. 2020, compulsory,* **[B9954]**) In this problem you analyse the signal through in small time frames (windows). If one uses Fourier analysis, the signal inside each frame should have a stable frequency content. In automatic speech recognition a typical frame length is 10 ms. If the sampling frequency is 16000 Hz (16000 samples per second), it corresponds 160 samples. Efficient algorithms for DFT apply frame lengths of power 2 (64, 128, 256, 512, 1024, . . .).

Download your **personal** signal helsinkiin2265.wav from MyCourses.

Open it in Matlab as vector `x`. Use the code below or follow the instructions and write the code by yourself.

(a) Compute DFT of length $N = 256$ with overlapping frames $V = 128$. First frame is components with indices $(1, N)$, second $(N-V, 2N-V-1)$, third $(2N-2V, 3N-2V-1)$, etc. DFT for a frame m:

$$xF_m[k] = \sum_{n=0}^{N-1} x_m[n] e^{-j2\pi nk/N}, \quad 0 \le k \le N-1$$

(b) Compute IDFT of logarithm of absolute values of spectral components computed in (a):

$$xC_m[n] = \sum_{k=0}^{N-1} \log(|xF_m[k]|) e^{j2\pi nk/N}, \quad 0 \le n \le N-1$$

(c) Check that `xC` is purely real-valued (`sum(imag(xC))` should be almost zero).

(d) Visualize the results. You should now have a spectrogram and a cepstrum. Pay attention to vowels in spectrogram (comb-shape). Cepstral components are needed in automatic speech recognition.

Some code available:

```
m = 1;                  % column index for images iF and iC

for k = (1 : (N-V) : length(x)-N)
  tmp = x(k : k+N-1);  % a N-length frame of signal
  xF  =                % _DFT_ of signal 'tmp'
  xC  =                % _inverse DFT_ of _logarithm_ of _absolute_ value of spectr
  iF(:,m)  = abs(xF(1:N/2));  % absolute of first half of spectral components
  iC(:,m)  = real(xC(1:N/2)); % real part of first half of xC components
  m = m+1;
end

figure(61); clf;
imagesc(iF);
colormap(gray);
colorbar

figure(62); clf;
imagesc(iC);
colormap(gray);
colorbar
```

Follow-up Assignments: 200-series

208. (*DL 14.4. 2020, 0–1,* **[B9946]**) Using Matlab, create the following sequences and their discrete Fourier transforms, and plot them in the time and frequency domain. Consider the effect of sequence length N, when $N = 8$, $N = 16$, $N = 256$, $N = 2048$. Compute each time N-point DFT with the same N. Confirm also that inverse DFT (`ifft`) of transformed sequence $X[k]$ yields back the original $x[n]$.

  a) $x_1[n] = \sin(0.22\pi n)$
  b) $x_2[n] = \cos(0.22\pi n)$
  c) $x_3[n] = \sin(0.25\pi n)$,

  Explain briefly in your answer: (i) differences and similarities between $x_1[n]$ and $x_2[n]$, (ii) effect of N in time and frequency domain (trade-off between time and frequency), (iii) spread of frequencies, when comparing / analyzing $x_1[n]$ and $x_3[n]$ (suom. taajuusvuoto).

  Note that `fft` is computed $k = [0, N-1]$, that is, $\omega_k = [0, 2\pi \cdot (N-1)/N]$ (Nyqvist frequency in middle). Example code:

```
N  = 8;                   % four cases i) 8, ii) 16, iii) 256, iv) 2048
n  = [0 : N-1];           % index (time) vector
x  = sin(0.22*pi*n);      % sequence x[n]; cases a), b), c)
xF = fft(x);              % N-point DFT
figure(1); plot(n, x);  grid on;         % time-domain
figure(2); plot(n, abs(xF)); grid on;    % frequency-domain magn.resp.
figure(3); plot(n, angle(xF)); grid on;  % frequency-domain phase resp.
figure(4); stem(n, abs(xF)); grid on;
```

  [T37].

209. (*DL 14.4. 2020, 0–2,* **[B9923]**) Consider the following cases of linear-phase FIR filters.

  a) Consider an impulse response

  $$h[n] = 2\delta[n-4] + 3\delta[n-5] - 3\delta[n-6] - 2\delta[n-7]$$

  Sketch the (anti)symmetric sequence $h[n]$. Show the symmetry point. Due to symmetry the filter has linear phase response. Write down the frequency response in the format

  $$H(e^{j\omega}) = e^{-jS\omega} \cdot \big(A_1 \sin(V_1\omega) + A_2 \sin(V_2\omega)\big)$$

  and determine coefficients S, A_1, V_1, A_2, and V_2.

  b) Compute $\angle H(e^{j\omega})$ and from that the group delay $\tau(\omega) = -\frac{\mathrm{d}}{\mathrm{d}\omega}\angle H(e^{j\omega})$.

  c) Consider another 16-th order linear-phase filter with $\angle H(e^{j\omega}) = -8\omega$ (and no zeros at the unit circle). See also Figure 12.

  A sequence $x_1[n] = 1 \cdot \cos(\omega_1 n)$ with normalized angular frequency $\omega_1 = 2\pi(f/f_T) = 0.05\pi$ is filtered giving an output sequence $y_1[n] = A_1 \cdot \cos(\omega_1 n + \theta_1)$, where A_1 is a constant from $1 \cdot |H(\omega = 0.05\pi)|$. Phase shift has been $\theta_1 = \angle H(\omega = 0.05\pi) = -8 \cdot 0.05\pi = -0.4\pi$, that is, sequences has been delayed by 8 samples ($8 \cdot (1/f_T)$ in seconds).

Another sequence $x_2[n] = 1 \cdot \cos(\omega_2 n)$ with $\omega_2 = 0.15\pi$ (three times higher frequency) is filtered. The output is $y_2[n] = A_2 \cdot \cos(\omega_2 n + \theta_2)$, where A_2 is another constant and $\theta_2 = \angle H(\omega = 0.15\pi) = -8 \cdot 0.15\pi = -1.2\pi$ (three times phase shift of $y_1[n]$). How much is this sequence delayed both in samples and seconds? What if the phase response had not been linear?

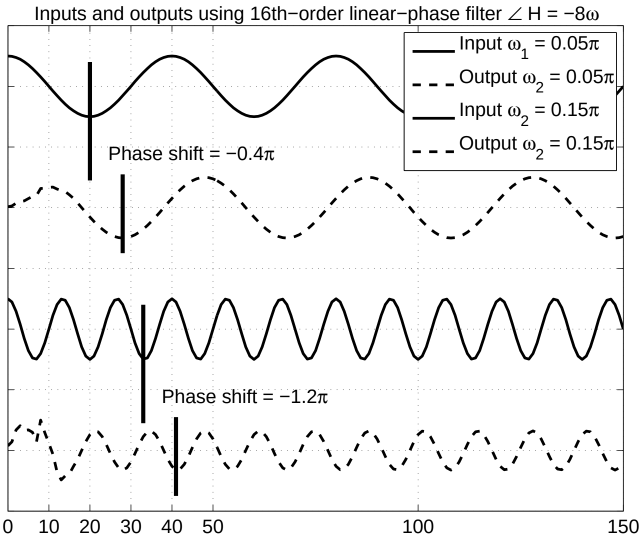


Figure 12: Problem 209: Two inputs and corresponding outputs from 16th-order linear-phase filter with $\angle H(e^{j\omega}) = -8\omega$.

(a) Note that the phase response is $\angle H(e^{j\omega}) = -S\omega$ and the group response $\tau(\omega) = S$. Hint: find the symmetry point of $h[n]$ which is then S. Apply Euler's formula $e^{j\omega} - e^{-j\omega} = 2j\sin(\omega)$.

[T58, T59]

210. (*DL 14.4. 2020, 0–1,* **[B9961]**) Consider circular shift and circular convolution.

a) Consider sequence $x_1[n] = -2\delta[n] + \delta[n-1] + 3\delta[n-2]$.
What is sequence $x_y[n]$ after *circular shift* by 3 delays and sequence length $N = L\{x_1[n]\}$: $x_7[n] = x_1[< n - 3 >_N]$?

b) Consider sequences $x[n]$ and $h[n]$ (this exercise package) with sequence length $N = 4$: $x[n] = \{\underline{0}, 0, 2, 3\}$ and $h[n] = \{\underline{0}, -4, 1, 0\}$. Compute *circular convolution* $y_C[n] = h[n] \circledast_4 x[n]$.

Using Matlab

c) compute the same $y_C[n] = h[n] \circledast_4 x[n]$ through frequency domain with commands

```
xF = fft(x, 4)
hF = fft(h, 4)
yF = xF .* hF
y4 = ifft(yF, 4)
```

d) Compute $y_C[n] = h[n] \circledast_8 x[n]$. Confirm that now $y_C[n] \equiv x_L[n]$ (result from linear convolution).

Circular convolution, check e.g. **[T52, T51]**. Example: from Friday three days forward is Monday. Monday $x[0]$, Friday $x[4]$, Sunday $x[6]$; $x[< 4+3 >_7] = x[0]$. The results of b-c doesn't differ much from linear convolution because lines "real" lengths are 2 and 2, so the length of linear convolution is $2+2-1 = 3 < 4$. You can try sequences of **[T52]**.

211. (*DL 14.4. 2020, 0–2,* **[B9922]**) This problem requires Matlab.

Create a sequence of random noise $s[n]$ (e.g., `randn`) with length 1024. Draw a spectrum $|S(e^{j\omega})|$ in range $\omega \in (0, \ldots, \pi)$. Create a moving average filter (MA-N) $H(e^{j\omega})$ and plot the magnitude response $|H(e^{j\omega})|$. Apply the filter to the noise $s[n]$ in order to get filtered noise $r[n]$. Plot the spectrum $|R(e^{j\omega})|$. Repeat for a couple of different noise sequences and for a couple of different values of N.

Can you show that $r[n] = h[n] \circledast s[n] \overset{F}{\leftrightarrow} R(e^{j\omega}) = H(e^{j\omega}) \cdot S(e^{j\omega})$? Attach to your returnings one good example of three spectra.

Check MA-N filter **[T21]** ([L0279]) and Matlab demo [M2058].

212. (*DL 14.4. 2020, 0–1,* **[M3015]**) An example of filter design and filtering using `signalAnalyzer` and
`filterDesigner` GUIs. **Attention! You have to use command-line function in your course assignment (Q1-Q4).** Analyze first the input signal and then design a highpass filter.

Task a: Filter out "metallic noise" of a demo audio file!

(0) Download. Download an audio file `M3015.wav` from MyCourses and import it to Matlab by
`[x, fT] = audioread('M3015.wav');` What is the sampling frequency of the signal? _____ Hz.
Listen to the signal:
`soundsc(x,fT);` Can you hear the music and some "metallic" noise?

(1) Import and time-domain analysis of the signal. Open a GUI for signal analysis by typing `signalAnalyzer(x)`. Click `Time Values` in `ANALYZER` tab to set the sample frequency (rate) of the signal and change the x-axis from samples to time. How long is the signal? _____ s.

(2) Frequency-domain analysis of the signal. Click `Spectrum` in `DISPLAY` tab to get a spectrum estimate. There are two audio channels in the signal. The first channel (music) is up to some 6 kHz and the other one ("metallic noise") above that. Write down "exact" values: channel #1: 0 – ____ Hz (audio that you hear), channel #2: ____ – ____ Hz ("metallic noise"), channel #3: _____ – $fT/2$ Hz ("silence"), min and max values in decibels: ____ and ____.

(3) Filter specifications and implementation. Open a GUI for filter design by typing `filterDesigner`. You are first interested in the audio in the channel #2. You want to filter out the music and leave "metallic noise". You want to have a filter with **specifications** like in Figure 13: (a) highpass, (b) IIR, (c) minimum order, (d) sampling frequency (Fs) of the signal (`fT`), (e) Fstop and Fpass cut-off frequencies as written above, (f) minimum stopband attenuation (Astop), how many dBs you want to attenuate?), and (g) maximum

passband variation (Apass), say, 2 decibels. Any other values you can choose by yourself. After giving specifications, click button `Design Filter`. Check that your implementation fulfills the specifications!
What is the order of the filter? ____
Export the filter to workspace: `File` → `Export...`

Design a filter to extract the music as well. Now you want to have a lowpass filter and you can use the same values as in the highpass filter, just reverse Fpass and Fstop. Export the lowpass filter to workspace with different variable names, for example SOS1 and G1.

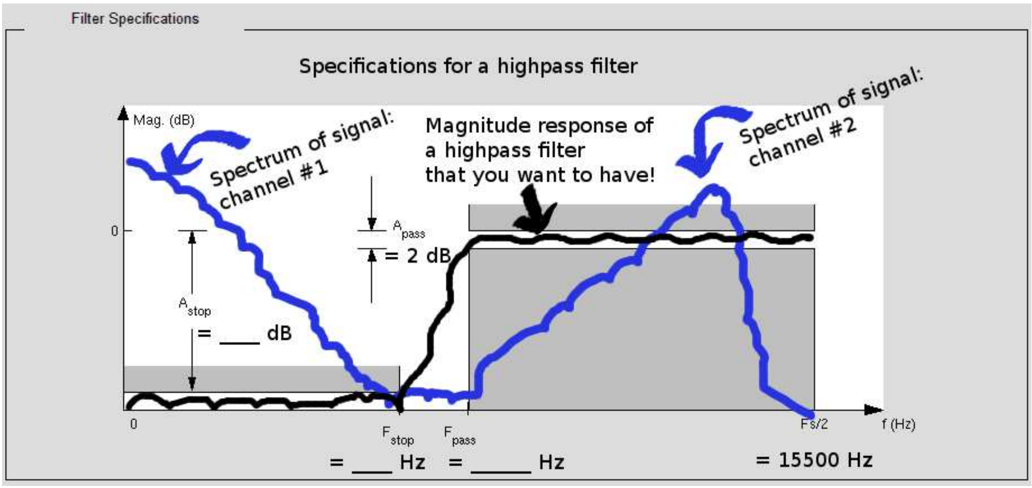


Figure 13: Problem 212: Filter specifications for a highpass filter. Stopband from 0 Hz to F_{stop} Hz with minimum stopband attenuation of A_{stop} (___) dB. Passband from F_{pass} Hz to $fT/2$ (15500) Hz with maximum passband ripple of A_{pass} (2) dB. Between F_{pass} and F_{stop} there is a "don't care region". Notice that in the stopband the attenuation is allowed to be more than A_{stop}; it has to be at least A_{stop}. Notice also that the filter order grows as $|F_{pass} - F_{stop}|$ decreases, or A_{stop} increases.

(4) Filter the signal. You can filter the signal by using function `sosfilt`: `noise=sosfilt(SOS,x);` `music=sosfilt(SOS1,x);`
Listen to the filtered signals by using function `soundsc` and use `signalAnalyzer` GUI to look at the spectra of the filtered signals to confirm that the filters worked as expected.

213. (*DL 14.4. 2020, 0–3,* **[G9905]**) Learn the connection between the magnitude response $|H(e^{j\omega})|$ and positions of poles and zeros in z-plane by playing with `filterDesigner`! Open the GUI by typing `filterDesigner` in Matlab.

Start by specifying the sampling frequency ($f_T = 14000$ Hz in the first task). Set Fpass < Fstop < Fs/2 and click "Design Filter" button on the bottom. Otherwise, the values do not matter as these choices will be removed later.

Click on the icon "Pole-Zero Editor" on the left-bottom corner. This gives you an interactive tool to see the connection between $|H(e^{j\omega})|$ and positions of poles and zeros. You can alter top figure by clicking icons in the top row, e.g., "magnitude response", "phase response", "group delay", "impulse response", "filter coefficients". Choose "Magnitude Response". If the range of y-axis becomes too small or large, click another icon and then "Magnitude Response" again to fix it.

In the interactive part (bottom figure and buttons and editing tools on the left), first, clear all existing crosses and circles. This can be done by choosing "Delete Pole-Zero" icon and then clicking them one by one or by drawing a rectangle around them.

Now you can add, change, and remove poles (crosses) and zeros (circles) in z-plane in the bottom figure. Just click with mouse and drag. You can instantly see the shape of magnitude response in the top figure.

You probably want to have conjugate pairs in order to have a real-valued impulse response. A single real-valued pole or zero can be set exactly at x-axis by editing values in left. The order of the filter is the maximum of the number of poles and zeros. It can be seen in "Current Filter Information" on the left-top frame.

Using `filterDesigner`'s **Pole-Zero Editor**, that is, by setting a set of poles and zeros,

a) create a lowpass filter with the sampling frequency $f_T = 14000$ Hz. Let the cut-off frequency (passband/stopband) be $f_c = 2000$ Hz. Implement both FIR (only zeros) and IIR (both zeros and poles) filter.

b) create a bandstop filter with $f_T = 34000$ where stopband is $4500 \ldots 5800$ Hz. Create a 8th order IIR.

Return figures from the both cases (pole-zero plot and magnitude response). Write down the filter coefficients of your filter $H(z) = \frac{B(z)}{A(z)} = \ldots$, which you can get by clicking the icon `[b,a]` in the top row.

If you want to filter a signal, export the filter by clicking `File` → `Export...`. The sampling frequency in filterDesigner should be chosen to be that of the signal.

Advanced Assignments: 300-series

301. (*DL 14.4. 2020, 0–1,* [**B3221**]) Write down the transfer function $H(z)$ ("siirtofunktio"), sketch the pole-zero plot and the amplitude response from that for the following filters having poles p_i and zeros z_i:

   a) $p_{1,2} = 0.3 \pm 0.9j$, $z_1 = 1$,

   b) $p_{1,2} = 0.9e^{\pm j0.25\pi}$, $p_3 = -0.9$, $z_{1,2} = e^{\pm j0.25\pi}$, $z_3 = 0$.

302. (*DL 14.4. 2020, 0–3,* [**T1003**]) Consider a stable and causal discrete-time LTI system S_1, whose zeros z_i and poles p_i are at

$$\begin{array}{ll} \text{zeros:} & z_1 = 1, \quad z_2 = 1 \\ \text{poles:} & p_1 = 0.18, \quad p_2 = 0 \end{array}$$

   Add a LTI FIR filter S_2 in parallel with S_1 as shown in Figure 14 so that the whole system S is causal second-order bandstop filter, whose minimum is approximately at $\omega \approx \pi/2$ and whose maximum is scaled to one. What are transfer functions S_2 and S? Show clear intermediate steps.

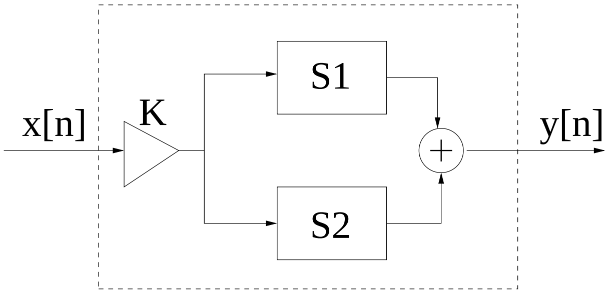


Figure 14: Problem 302: Filter S constructed from LTI subsystems S_1 and S_2.

303. (*DL 14.4. 2020, 0–1,* [**T1021**]) Consider a causal lowpass filter $H(z)$, whose passband ends at 4 kHz, stopband starts from 5 kHz and the sampling frequency is 12 kHz. The amplitude response is in Figure 15(a) and the start of the impulse response $h[n]$ in Figure 15(b). Modify the filter so that it can handle DAT-recordings with the sampling frequency of 48 kHz.

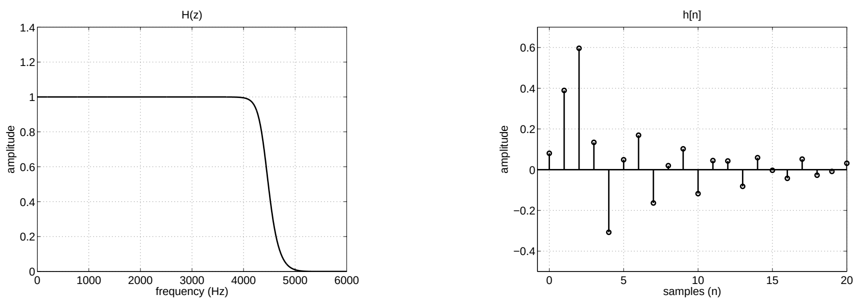


Figure 15: Problem 303: (a) $|H(e^{j\omega})|$, (b) $h[n]$.

   a) Increase the sampling frequency with the factor $L = 4$. Draw the amplitude spectrum of the upsampled filter $H(z^4)$ in range $0 \ldots 24$ kHz and the first ten values of the impulse response $h[n/4]$.

b) What has to be done so that the filter $H(z^4)$ works as a lowpass filter with the original cut-off frequencies?

Check e.g. **[T81]** and the Matlab exercises.

Return III, DL 4.5.2020

Basic Assignments: 100-series

122. (*DL 4.5. 2020, compulsory,* [**B9916**])

   Do a time-use plan (at the beginning). Reserve time from your calendar to complete the course, log your time usage during the course, and report it during each exercise return. State your current target grade on the cover page. You can use the tables found in the beginning of this booklet to evaluate the target grade.

123. (*DL 4.5. 2020, compulsory,* [**B9953**]) LTI systems can be examined from a several points of view. See Figure 16, where a digital linear and time-invariant (LTI) system can be expressed in several (pretty) equivalent forms. When you are given one form your task is to find others.

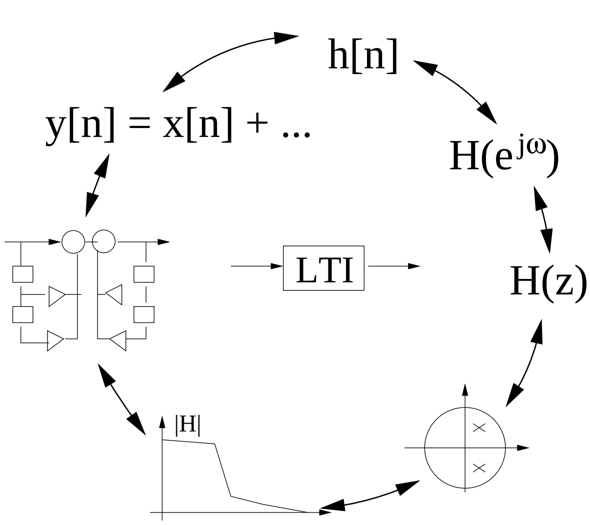


Figure 16: Problem 123: LTI system and its analysis.

Consider a digital LTI filter

$$H(z) = \frac{1 + 1.2z^{-1} + 0.8z^{-2}}{1 - 0.9z^{-1}}, \qquad |z| > 0.9$$

a) Compute poles and zeros, and sketch pole-zero plot.
b) Determine frequency response $H(e^{j\omega})$.
c) Sketch the magnitude response $|H(e^{j\omega})|$.
d) Determine gain K so that $K \cdot \max\{|H(e^{j\omega})|\} = 1$.
e) (Sketch the phase response $\angle H(e^{j\omega})$.)
f) Determine difference equation.
g) Draw the flow (block) diagram.
h) Determine impulse response both in recursive and non-recursive manner
i) Is the filter IIR or FIR?
j) What is the order of the filter?
k) Is the filter causal? Show it.

l) Is the filter stable? Show it.

m) How to enter into Matlab? Write down your code.

[T42, T53, T54, T55]

124. (*DL 14.4. 2020, compulsory,* **[G1500]**) Consider a second-order IIR type LTI system with the difference equation

$$y[n] + 0.8y[n-1] + 0.65y[n-2] = x[n] - 0.81x[n-2]$$

a) Draw the flow diagram in a "standard" Direct Form I format.

b) Determine the transfer function $H(z)$.

c) Compute poles and zeros, and sketch a pole-zero plot.

d) Sketch the magnitude response $|H(e^{j\omega})|$ using information from the pole-zero plot.

e) Describe your filter: FIR/IIR, order, stable/astable, causal/non-causal, lowpass / highpass / bandpass / bandstop / allpass.

f) Derive the partial fraction expansion (if needed) and determine the impulse response $h[n]$.

[T42, T53, T54, T55]

125. (*DL 14.4. 2020, compulsory,* **[L0508]**) The filter in Figure 17 is in canonic direct form II (DF II). Draw it in DF I. What is the transfer function $H(z)$?

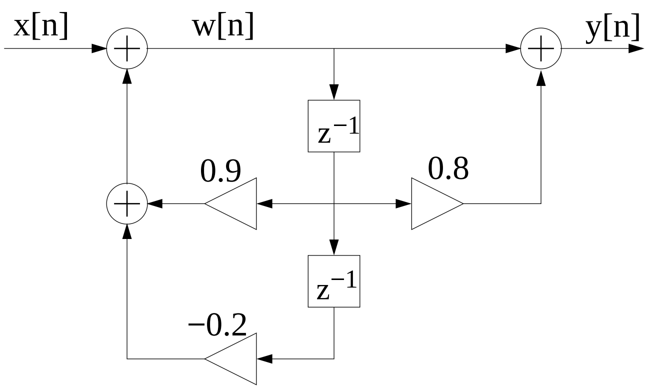


Figure 17: The block diagram of direct form II of Problem 125.

126. (*DL 4.5. 2020, compulsory,* **[M4005]**) FIR filters are often designed with linear scale in y-axis. See typical specifications of a FIR filter in Figure 18. The passband ripple varies between $1 - \delta_p \ldots 1 + \delta_p$. The stopband ripple is designed to be less than δ_s.

Parks-McClellan algorithm provides an optimal equiripple FIR filter. The estimation of the order is computed with `firpmord` and then the coefficients of the filter with `firpm`.

Task: Run the code `M4005.m`. Draw specifications for a FIR bandstop filter: sampling frequency $f_T = 16000$ Hz, stopband in range $f_s \in (3000, 3800)$ Hz, passband in ranges $f_{p1} \in (0, 2500)$ and $f_{p2} \in (4300, f_T/2)$, and the ripples are $\delta_p = 0.08$ and $\delta_s = 0.1$.

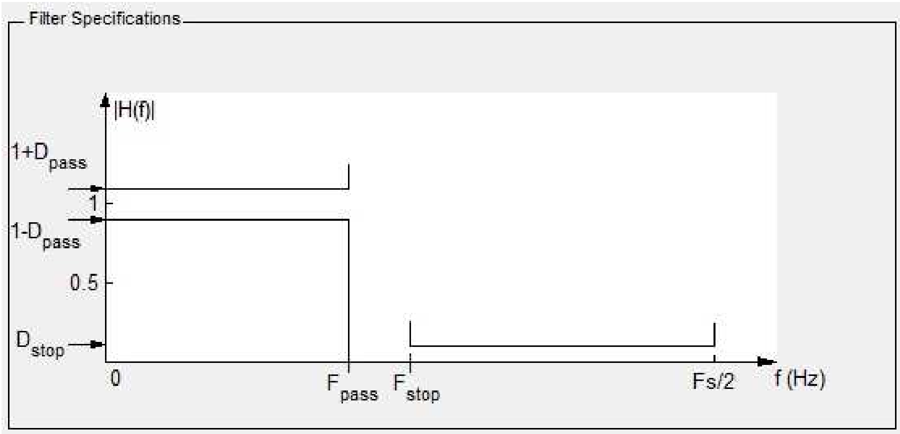


Figure 18: Problem 126: Filter specifications for a FIR filter.

Compute $H(z)$. Are the specifications fulfilled? Read `help firpmord`. What is the filter order? Test also what is the filter order if you decrease width of transition band (band between f_s and f_p) from 500 Hz to 250 Hz.

Filter the demo signal. Is the noise suppressed? What does variation $1 - \delta_p \ldots 1 + \delta_p$ correspond in IIR specifications where the maximum is scaled to 0 decibels (1 in linear scale) and α_{max} is given in power decibels $|H(e^{j\omega})|^2$?

127. (*DL 4.5. 2020, compulsory,* **[G2100]**) See the specifications of a highpass filter given in Figure 19 where x-axis is in range $(0, f_T/2)$ Hz. In addition, the magnitude response of an IIR filter fulfilling specifications is plotted.

Frequency values are often normalized according to the sampling frequency: $\omega = 2\pi f / f_T$, or in Matlab $f_{\texttt{Matlab}} = \omega/\pi$. In order words, half of the sampling frequency $f_T/2$ corresponds to π or 1, respectively.

Sampling frequency f_T	0	$f_T/4$	$f_T/2$	$3f_T/4$	f_T	Hz
Frequency f, $f_T = 20$ kHz	0	5000	10000	15000	20000	Hz
Angular frequency Ω	0	10000π	20000π	30000π	40000π	rad/s
Normalized angular frequency ω	0	$\pi/2$	π	$3\pi/2$	2π	rad/sample
Normalized frequency f_{norm}	0	0.5	1	1.5	2	1

Write down the values for the cut-off frequencies: $0 < f_s < f_p < f_T/2$, $0 < \omega_s < \omega_p < \pi$, and $0 < \texttt{Ws} < \texttt{Wp} < 1$.

	f, $[f]$ = Hz	ω, $[\omega]$ = rad/sample	`W`
passband cut-off	$f_p \approx$	$\omega_p \approx$	`Wp` $\approx$
stopband cut-off	$f_s \approx$	$\omega_s \approx$	`Ws` $\approx$

The corresponding Matlab command for plotting the magnitude and frequency specifications for IIR filters is `speksitIIR(Wp, Ws, 3, 50, 'high', 8000)`, where `speksitIIR.m` is from course web site, not a command of standard Matlab.

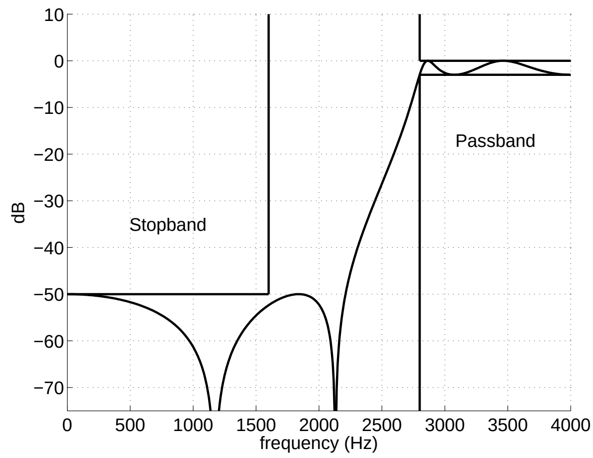


Figure 19: Problem 127: Filter specifications (`speksitIIR.m`) and the magnitude response (`freqz(B,A)`) of a filter meeting them.

Follow-up Assignments: 200-series

214. (*DL 4.5. 2020, 0–1,* [**B4010**]) Determine the transfer function $H(z)$ of the system in Figure 20. Hint: use temporary variables $w_k[n]$ after each summing unit k, write down difference equations, apply z-transform, and reduce to $H(z)$.

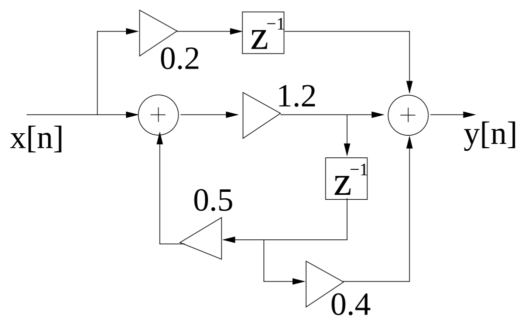


Figure 20: Problems B4010, B4011, B4012, and B4013: LTI system.

Check e.g. [**T64**].

215. (*DL 4.5. 2020, 0–1,* [**L0502**]) Analyze the digital filter structure shown in Figure 21 and determine its transfer function $H(z) = Y(z)/X(z)$.

   a) Is the system LTI?

   b) Is the structure canonic with respect to delays?

   c) Compute $H(z)H(z^{-1})$ (the squared amplitude response). What is the type of this filter (lowpass/highpass/bandpass/bandstop/allpass)?

216. (*DL 4.5. 2020, 0–1,* [**B4011**]) After determining $H(z)$ of the system in Figure 20 draw the flow diagram in direct form I (DF I), DF II and transposed DF I_t and DF II_t.

Direct form I: First "forward" FIR part and then "feedback" IIR part. Direct form II: IIR first, then FIR. Check e.g. [**T63**] and [**T63**].

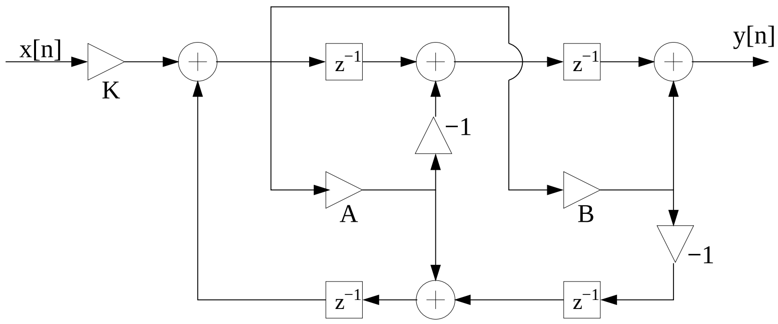


Figure 21: The flow diagram of the system in Problem 215.

.

217. (*DL 4.5. 2020, 0–1,* [**B4012**]) After determining $H(z)$ of the IIR system in Figure 20 compute poles and zeros, express the order of the filter, sketch the pole-zero diagram, sketch the magnitude response, check if the filter is stable or not, causal or not.

Analyzing the LTI system. Check e.g. [**T55**]. Important topic!

218. (*DL 4.5. 2020, 0–1,* [**B4013**]) After determining $H(z)$ of the IIR system in Figure 20 compute the impulse response $h[n]$ (in closed form).

Hint: z-transform formulas $a^n\mu[n] \leftrightarrow 1/[1-az^{-1}]$ and $ax[n-k] \leftrightarrow az^{-k}X(z)$, and

$$\begin{aligned} H(z) &= \frac{a+bz^{-1}}{1-cz^{-1}} \\ &= a\cdot\frac{1}{1-cz^{-1}} + (bz^{-1})\cdot\frac{1}{1-cz^{-1}} \\ &= a\cdot H_1(z) + (bz^{-1})\cdot H_1(z) \\ &\overset{z}{\leftrightarrow} a\cdot h_1[n] + b\cdot h_1[n-1] \\ H_1(z) &= \frac{1}{1-cz^{-1}} \\ &\overset{z}{\leftrightarrow} c^n\mu[n] = h_1[n] \\ h_1[n-1] &= c^{n-1}\mu[n-1] \qquad \text{check!!!} \end{aligned}$$

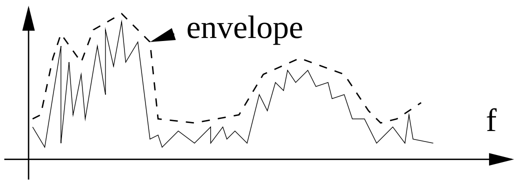
Express $h[n]$ in closed form (non-recursive).

Advanced Assignments: 300-series

304. (*DL 4.5. 2020, 0–1,* [**B9903**]) This problem requires Matlab.

Analyze small segments (20 – 50 ms) of a demo word `kiisseli.wav` (can be found in MyCourses). Find two segments, (a) a period part of the signal, e.g., /i/, and (b) an aperiodic part of the signal, e.g. /s/, and clip them into vectors (`xpart1`, `xpart2`). Apply discrete Fourier transform and plot the amplitude spectrum both in linear scale (`abs()`) and in logarithmic scale, decibels (`20*log10(abs())`).

Reply also in (a) what is the "fundamental period and frequency" (quasi-periodic = "almost" periodic part of signal). In case of logarithmic spectrum determine (by looking) some 2 – 5 "hills" (formants) in the envelope (*verhokäyrä*).



Return four figures of spectra (e.g. /i/ and /s/ in two scales), values of T_0 and f_0 for the periodic signal part, and Matlab code.

In Matlab, when the sampling frequency of the signal is 20000 Hz, `xpart = x(5000:5999);` clips a part from 0.25 s to 0.30 s.

If you want to plot four figures into the same window:

```
clf;
subplot(2, 2, 1) % subplot(# cols, # rows, which subfigure)
plot(w, ...)
```

305. (*DL 4.5. 2020, 0–1,* [**T1022**]) Transform the filter structure in Figure 22 to a canonic (with respect to delay units) filter structure having the same transfer function $H(z)$. What is $H(z)$ and the order of that? Draw the block diagram again in canonic form using, e.g. Direct Form II.

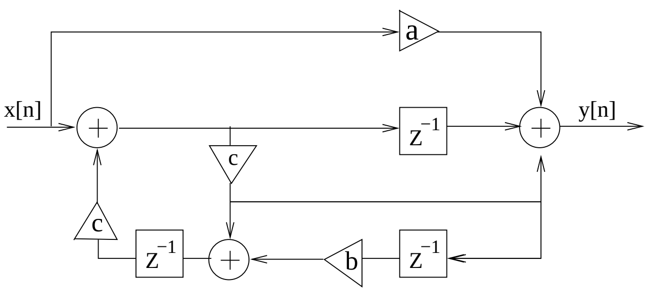


Figure 22: Problem 305: Filter.

306. (*DL 4.5. 2020, 0–2,* [**M5009**]) Consider a fourth order filter $H(z)$

$$H(z) = K \cdot \frac{b_0 + b_1 z^{-1} + b_2 z^{-2} + b_3 z^{-3} + b_4 z^{-4}}{a_0 + a_1 z^{-1} + a_2 z^{-2} + a_3 z^{-3} + a_4 z^{-4}}$$

in *Direct Form I structure* (DF I) as seen in Figure 23(a). If we are dealing with infinite wordlength, coefficients can have values such as $b_0 = \pi$. However, when implementing filters either in hardware or software (fixed-point), there will be a finite number of bits for representing each number, e.g. $b_0^{'} = 3.125 = 3 \cdot 2^0 + 1 \cdot 2^{-3}$. The change of coefficients ($b_0 \to b_0^{'}$) affects the computation of magnitude response as well as the locations of the roots in the pole-zero plot ($ax^2 + bx + c = a \cdot (x - d_1) \cdot (x - d_2)$).

Another type of structure with which filters can be realized is *second-order systems* (SOS). Figure 23(b) shows a structure for a fourth order filter,

$$H(z) = G \cdot H_1(z) \cdot H_2(z) = G \cdot \frac{b_{01} + b_{11}z^{-1} + b_{21}z^{-2}}{1 + a_{11}z^{-1} + a_{21}z^{-2}} \cdot \frac{b_{02} + b_{12}z^{-1} + b_{22}z^{-2}}{1 + a_{12}z^{-1} + a_{22}z^{-2}}$$

Task: Examine the filter

$$H(z) = \frac{0.5806 - 1.8325z^{-1} + 2.6023z^{-2} - 1.8325z^{-3} + 0.5806z^{-4}}{1 - 2.3933z^{-1} + 2.5562z^{-2} - 1.4889z^{-3} + 0.4302z^{-4}}$$

Compare behavior of the filter implementations (i) with no quantization, (ii) quantizing coefficients of the direct form (DF), and (iii) quantizing coefficients of second-order systems (SOS). You can run the code `M5009.m` and `a2dR.m` (by Mitra's book) from MyCourses.

a) Plot the pole-zero-diagram, amplitude response and start of the impulse response (`impz`) using the original fourth order direct form filter without any quantization (i). What kind of filter is it: lowpass / highpass / bandpass / bandstop / allpass / notch?

b) Compute the coefficients of the second order implementation of Figure 23(b) (`tf2sos`).

$$H(z) = \underline{\quad} \cdot \frac{\underline{\quad} + \underline{\quad}z^{-1} + \underline{\quad}z^{-2}}{1 + \underline{\quad}z^{-1} + \underline{\quad}z^{-2}} \cdot \frac{\underline{\quad} + \underline{\quad}z^{-1} + \underline{\quad}z^{-2}}{1 + \underline{\quad}z^{-1} + \underline{\quad}z^{-2}}$$

A direct form filter can be created from SOS using (`sos2tf`).

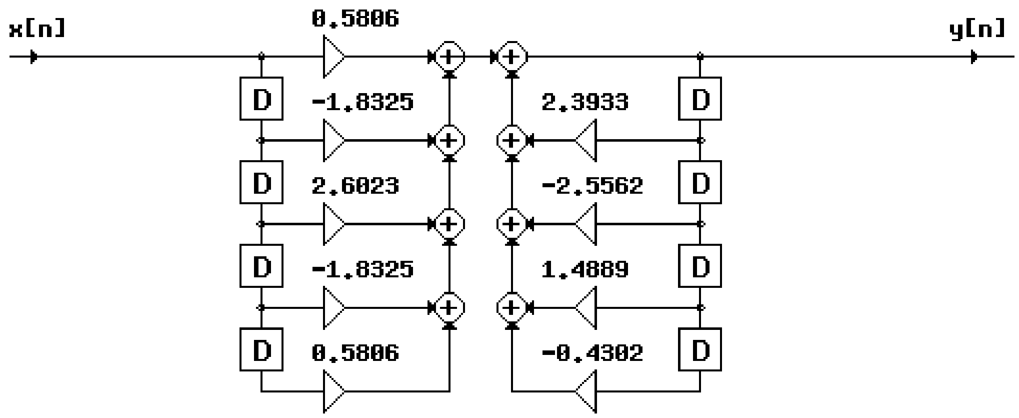
c) Run the code `M5009.m` and quantize coefficients both in direct form (Figure 23(a)) and in second order systems (Figure 23(b)). Decrease the number of bits from 8 step by step (variable name `Qbits`). Plot the location of poles and zeros for all cases. Plot the amplitude response for all cases. When do you find significant difference between (i), (ii) and (iii)? What happens and why? Any problems with stability?

307. (*DL 4.5. 2020, 0–2,* [**Q2**])

Download the DTMF-type audio file `dtmf_studentnumber.wav` from the MyCourses. One audio file per group is enough. The goal of this task is to analyze the given signal and to construct a Matlab program that finds the correct tones (i.e., numbers) present in it automatically, as detailed below.

Explain briefly what is DTMF (dual-tone multifrequency). The frequencies (LOW, HIGH) in the audio files are the same. Every audio signal is corresponding to a push of a button at least 70 ms long and there is a separation of at least 40 ms between each tone. Only the numbers $0, \ldots, 9$ are used (letters A-D are not present).

Design a Matlab function that recognizes automatically the tones that are present in the audio file. One possible solution can be found in Mitra's book (found in MyCourses) where DFT with $N = 205$ ($F_T = 8$ kHz) is used. However, there are many other alternatives as well. A key feature in a successful implementation is that the signal is processed in



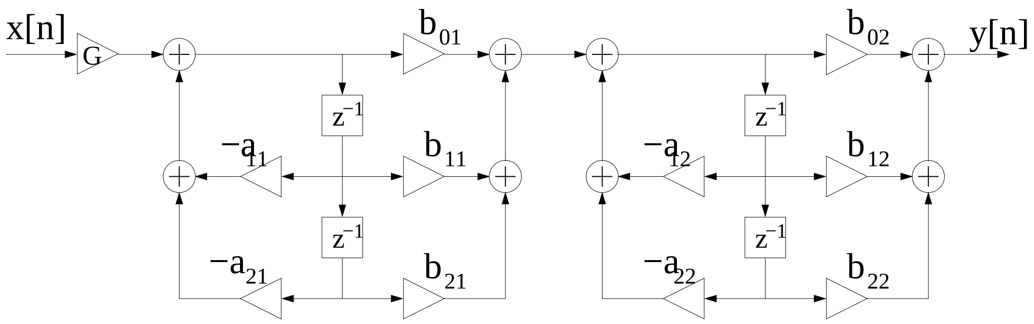


Figure 23: Problem 306: Filter $H(z)$ in (a) Direct Form I, (b) Second order systems (SOS) in cascade.

sufficiently short window using, e.g., a `for`-loop. If you borrow parts of the solution from someone else, write it down clearly.

You need to design the Matlab function in such a way that it can be called as: `[numberstring] = dtmf_X(file)`, where `X` is your student number, `file` is a string with the name of the audio file that is loaded and `numberstring` a decoded string of numbers (you can convert numbers to string with `num2str`). For example,

```
>> [numbers] = dtmf_12345A('dtmf_12345A.wav')
numbers = 9876543210
>> class(numbers)
ans = char
```

Notice that in order to avoid the same computations each time the function is called, you can calculate coefficients etc. that stay constant (i.e., they are independent of the audio file) also outside of the function. They can then be either written as fixed variables inside the function or taken as extra arguments when the function is called, e.g., `[numberstring] = dtmf_12345A(file, arg1, arg2, ...)`.

308. (*DL 4.5. 2020, 0–6,* [**Q4**])

Next we continue with the same topic as in **Q2**. The goal of this task is to fine tune and examine the DTMF decoding function you designed in **Q2**.

Test your Matlab based DTMF decoding function on the 12 different test signals `dtmf_TESTI_*.wav` found in MyCourses. You may use the script `dtmf_TESTAUS.m` for testing. Return in MyCourses a log file that contains a semicolon separated list with the content: (1) student number, (2) test signal number $(101, .., 112)$, (3) the running time of the function (use Matlab functions `tic` and `toc`), (4) was the decoding successful, and (5) what was the number string that the function output. Notice that some of the test signals are quite "pathological" and it may be impossible to attain perfect recognition in finite time.

Finally, analyze your Matlab function using `profile` and try to make it faster and more efficient than in any other group.

309. (*DL 4.5. 2020, 0–4,* **[QUE17]**)

Prepare up to eight multiple-choice questions for the final checkup exam and present their correct answers with brief explanations. The questions should consider Lectures 2-9 (at maximum one per each lecture). Questions that are original and good enough to be used in an actual checkup exam reward 0.5 bonus points. This assignment is also an alternative to corresponding lecture questionnaires that may also reward up to 4 points in total (only either half-point can be earned per each lecture).

Return IV, DL 25.5.2020

Basic Assignments: 100-series

128. (*DL 25.5. 2020, compulsory,* [**B9916**])

Do a time-use plan (at the beginning). Reserve time from your calendar to complete the course, log your time usage during the course, and report it during each exercise return. State your current target grade on the cover page. You can use the tables found in the beginning of this booklet to evaluate the target grade.

129. (*DL 25.5. 2020, compulsory,* [**M4002**]) IIR filters are designed with specifications in Figure 24.

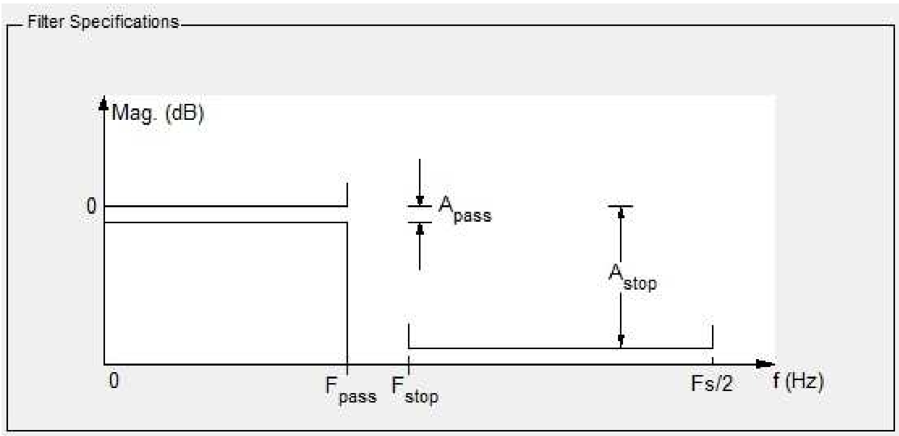


Figure 24: Problem 129: Filter specifications for an IIR filter. In Matlab variables `Wp` ($F_{pass} = 2f_p/f_T$) and `Ws` ($F_{stop} = 2f_s/f_T$), which are normalized cut-off frequencies, maximum passband ripple `Rp` (A_{pass}) and minimum stopband attenuation `Rs` (A_{stop}) in decibels.

Task: Load the m-file `M4002.m` and run the code. Verify first that the sampling frequency f_T of `avs1.wav` is 11025 Hz. Design an elliptic IIR lowpass filter with $f_p = 1800$ Hz (end of passband frequency) and $f_s = 2000$ Hz (start of stopband frequency). Passband ripple may be 2 dB, and minimum stopband attenuation at least 40 dB. Read `help design, ellipord | ellip`.

Write down the order of the filter and the coefficients of the transfer function. Plot also the pole-zero diagram, where the filter order is $\max\{\#\text{zeros}, \#\text{poles}\}$.

In the end, filter a demo signal and listen to the result.

> See Matlab exercises. Steps in filter design: (i) determine specifications; (ii) choose the type of filter (IIR/FIR, approximations); (iii) estimate order (`ellipord`); (iv) compute coefficients (`ellip`); (v) check that specifications are fulfilled!; (vi) apply.

130. (*DL 25.5. 2020, compulsory,* [**L0614**]) Use windowed Fourier series method and design a FIR-type (causal) lowpass filter with cutoff frequency $3\pi/4$. Let the order of the filter be 4.

See Figure 25, in left the amplitude response of the ideal lowpass filter $H(e^{j\omega})$ with cut-off frequency at $3\pi/4$. In right, the corresponding inverse transform of the desired ideal filter

$h_d[n]$, which is sinc-function according to the transform pair rect(.) $\leftrightarrow$ sinc(.):

$$h_d[n] = \{\ldots, -0.1592, 0.2251, \underline{0.75}, 0.2251, -0.1592, \ldots\}$$

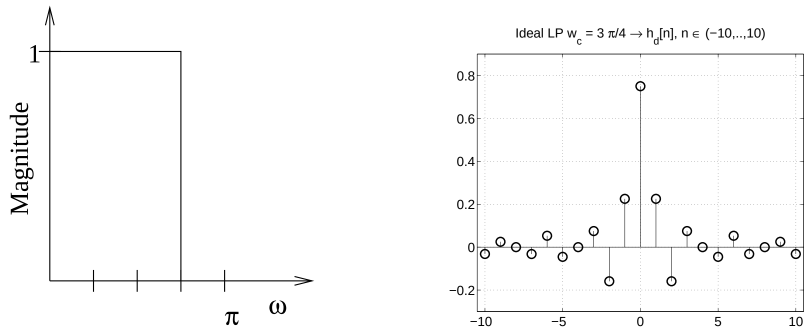


Figure 25: Problem 130: (a) The amplitude response of the ideal lowpass filter, and (b) the corresponding impulse response $h[n]$ values. The cut-off frequency is at $\omega_c = 3\pi/4$.

a) Use the rectangular window of length 5, see Figure 26(a). The window function is $w_r[n] = 1, -M \leq n \leq M, M = 2$

b) Use the Hamming window of length 5, see Figure 26(b). The window function is

$$w_h[n] \quad = \quad 0.54 + 0.46\cos\left(\frac{2\pi n}{2M}\right) \quad , \qquad -M \leq n \leq M, M = 2$$

which results to $w_h[n] = \{0.08, 0.54, \underline{1}, 0.54, 0.08\}$

c) Compare how the amplitude responses of the filters designed in (a) and (b) differ assuming that the window size is high enough (e.g. $M = 50$).

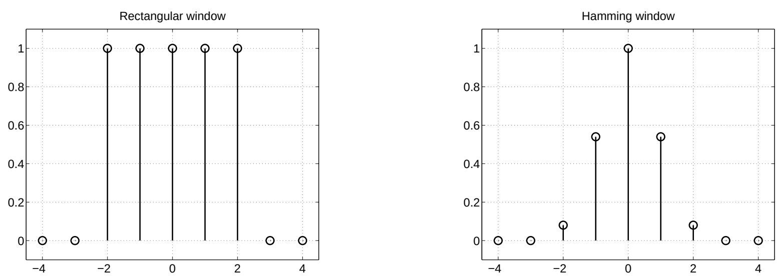


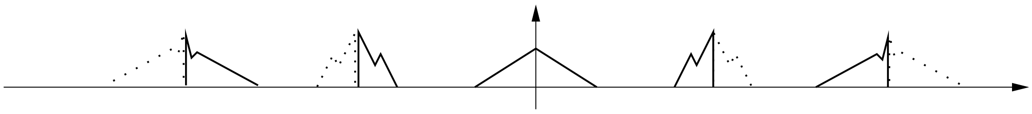
Figure 26: Problem 130: (a) rectangular window $w_r[n]$ of length 5, and (b) Hamming window $w_h[n]$ of length 5.

131. (*DL 25.5. 2020, compulsory,* **[T1045]**) Sketch roughly (without Matlab) the magnitude responses of digital Butterworth, Chebyshev I and elliptic filters in range $\omega \in [0 \ldots \pi]$ given the specifications below. Concentrate to show how the three types of approximations differ from each other. Filters have been designed first in continuous s-plane and then converted to z-plane via bilinear transformation.

a) 5th order lowpass filter with passband $[0 \ldots \pi/3]$ and stopband $[2\pi/3 \ldots \pi]$. Maximum passband ripple 2 decibels and minimum stopband attenuation 30 dB.

b) 4th order bandpass filter with first stopband $[0 \ldots \pi/4]$, passband $[3\pi/8 \ldots \pi/2]$, and second stopband $[5\pi/8 \ldots \pi]$. Maximum passband ripple 2 decibels and minimum stopband attenuation 20 dB.

132. (*DL 25.5. 2020, compulsory,* [**Q1**]) The goal of this task is to analyze the given audio signal `Q1_K_studentnumber`, that has three frequency bands, and design filters that remove an unwanted component in it, as detailed below.



Design such FIR and IIR lowpass filters for the sample so that after filtering the metallic humming / whirring noise has mostly disappeared. The given requirements are: maximum passband attenuation is 1 dB and transition band is 400 Hz. Choose the rest of the requirements as you see best and provide an explanation why you chose them in such a way. Finally, compared the FIR and IIR filter designs (how the magnitude responses compare, filter orders, etc.).

133. (*DL 25.5. 2020, compulsory,* [**B9994**])

Participate in the **check-up exam** in May. Information about the check-up exam will be available in MyCourses.

Follow-up Assignments: 200-series

219. (*DL 25.5. 2020, 0–1,* [**B6016**]) There are several ways to make specifications for a digital filter (*Mitra 2Ed Sec. 7.1.1, p. 423 / 3Ed Sec. 9.1.1, p. 489*). Two typical cases are given in Figure 27 with corresponding decibel values and frequency normalizations

$$\begin{aligned}
\alpha_p &= -20\log_{10}(1-\delta_p)\ \text{dB}, \quad \text{“peak passband ripple”} \\
\alpha_s &= -20\log_{10}(\delta_s)\ \text{dB}, \quad \text{“minimum stopband attenuation”} \\
\alpha_{max} &= -20\log_{10}\big(1/(\sqrt{1+\epsilon^2})\big)\ \text{dB} \\
&= 20\log_{10}\big(\sqrt{1+\epsilon^2}\big)\ \text{dB}, \quad \text{“maximum passband attenuation”} \\
\alpha_{max} &\cong -20\log_{10}(1-2\delta_p) \cong 2\alpha_p, \quad \text{if } \delta_p \ll 1 \text{ as typically} \\
\omega_p &= 2\pi(f_p/f_T) \quad \text{“normalized angular cut-off frequency for passband”} \\
\omega_s &= 2\pi(f_s/f_T) \quad \text{“normalized angular cut-off frequency for stopband”}
\end{aligned}$$

A digital lowpass filter is specified by $\delta_p = 0.05$, $\delta_s = 0.3$, $\omega_p = 0.2\pi$, and $\omega_s = 0.3\pi$ as in Figure 27(a). Define values α_{max} and α_s as shown in Figure 27(b).

Remark. In Matlab `Wp` $\leftrightarrow \omega_p/\pi$, `Ws` $\leftrightarrow \omega_s/\pi$, `Rp` $\leftrightarrow \alpha_{max}$, and `Rs` $\leftrightarrow \alpha_s$ for digital IIR filter functions `butter, cheby1, cheby2, ellip` as in Figure 27(b). Some FIR functions define ripples as in Figure 27(a).

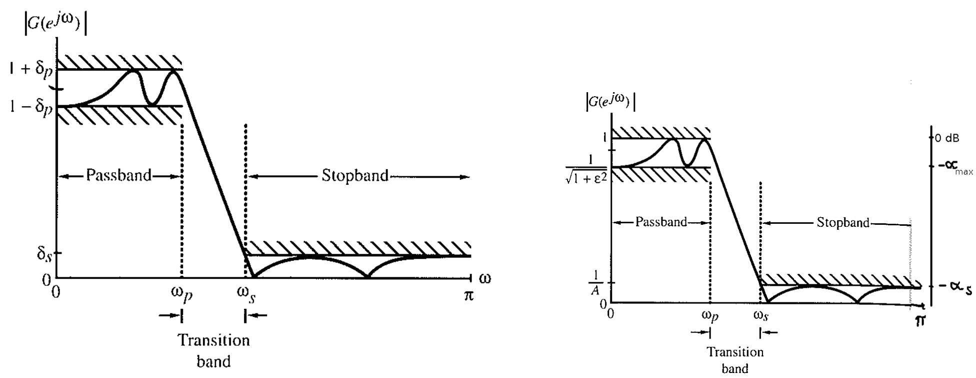


Figure 27: Problem 219: (a) Typical magnitude specifications for a digital FIR lowpass filter, and (b) normalized magnitude specifications for a digital IIR lowpass filter (*Mitra 2Ed Fig. 7.1,7.2, p. 424,425 / 3Ed Fig. 9.1, p. 490*). In passband $\alpha_{max} = -20\log_{10}(1/\sqrt{1+\epsilon^2}) \approx 2\alpha_p$ and maximum stopband magnitude is $\delta_s = 1/A$.

220. (*DL 25.5. 2020, 0–1,* [**M4006**]) **Task:** Look at the (IIR) filter specifications for each filter in Figure 28, read the values for the following quantities and convert them to values that Matlab requires (e.g. `help buttord`). See also Table 5 for conversion. Draw the specifications using `speksitIIR(Wp, Ws, Rp, Rs, type, fT)` (can be found in MyCourses).

- filter type: lowpass / highpass `'high'` / bandpass / bandstop `'stop'`
- sampling frequency $f_T =$ Hz, in Matlab it corresponds to 2.
- passband cut-off freq. $f_p =$ Hz, in Matlab `Wp = [ ]`
- stopband cut-off freq. $f_s =$ Hz, in Matlab `Ws = [ ]`
- maximum passband attenuation (maximum ripple) in decibels $\alpha_{max} =$ dB, in Matlab `Rp = [ ]`.

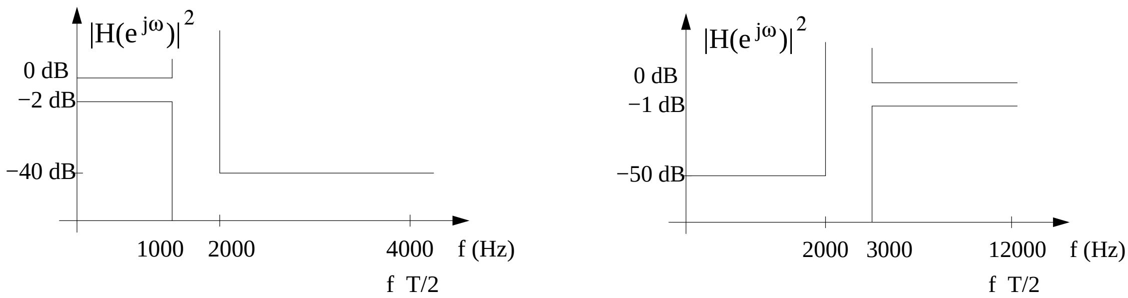


Figure 28: Filter specifications of two IIR filters of Problem 220. Here f_T is the sampling frequency.

- minimum stopband attenuation in decibels $\alpha_s =$ dB, in Matlab `Rs = [ ]`.

221. (*DL 25.5. 2020, 0–1,* **[T1046]**) Sketch roughly (without Matlab) the magnitude responses of digital Butterworth, Chebyshev I and elliptic filters in range $\omega \in [0 \ldots \pi]$ given the specifications below. Concentrate to show how the three types of approximations differ from each other. Filters have been designed first in continuous s-plane and then converted to z-plane via bilinear transformation.

    a) 8th order highpass filter with stopband $[0 \ldots \pi/4]$ and passband $[\pi/2 \ldots \pi]$. Maximum passband ripple 2 decibels and minimum stopband attenuation 40 dB.

    b) 6th order bandstop filter with with first passband $[0 \ldots \pi/4]$, stopband $[\pi/3 \ldots 2\pi/3]$, and second passband $[3\pi/4 \ldots \pi]$. Maximum passband ripple 2 decibels and minimum stopband attenuation 20 dB.

222. (*DL 25.5. 2020, 0–4,* **[Q3]**)

    Here we continue with the same topic as in **Q1**. The goal of this task is to analyze your audio file and demodulate the "noise" in band #2 so that it's spectrum is around the origin and the "noise" becomes audible.

    Use an IIR bandpass filter for the filtering of band #2. The given requirements are: maximum passband attenuation is 1 dB and transition bands are 400 Hz. Choose the rest of the requirements as you see best and provide an explanation why you chose them in such a way. The demodulation can be done using a multiplication with cosine or Matlab function `demod`. Notice that the subjective quality of the output is very sensitive to the demodulation frequency. After demodulation, filter the signal with a lowpass FIR filter with requirements: the filter has a linear phase and the transition band is 400 Hz.

223. (*DL 25.5. 2020, 0–2,* **[T1014]**) Consider an analog transfer function $H_a(s) = (s+a)/[(s+a)^2 + b^2]$, where coefficients a and b are real-valued. The pole-zero-plot of the filter (in s-plane) and amplitude response are as shown in Figure 29.

Table 5: Examples on frequency normalization, with the sampling frequency $f_T = 20$ kHz. Matlab uses f_{norm}. Angular frequency Ω is used with analog signals, ω with digital sequences.

Sampling frequency f_T (f_s!?)	0	$f_T/4$	$f_T/2$	$3f_T/4$	f_T	Hz
Frequency f, $f_T = 20$ kHz	0	5000	10000	15000	20000	Hz
Angular frequency Ω	0	10000π	20000π	30000π	40000π	rad/s
Normalized angular frequency ω	0	$\pi/2$	π	$3\pi/2$	2π	rad/sample
Normalized frequency f_{norm}	0	0.5	1	1.5	2	1

NOTE! You do not have to compute any z-plane transfer functions, or corresponding. Only sketching of figures is enough.

a) Sketch the amplitude response of a digital filter via impulse invariant method.

b) Sketch the amplitude response of a digital filter via bilinear transform.

c) Explain briefly, how the methods in a) and b) differ from each other.

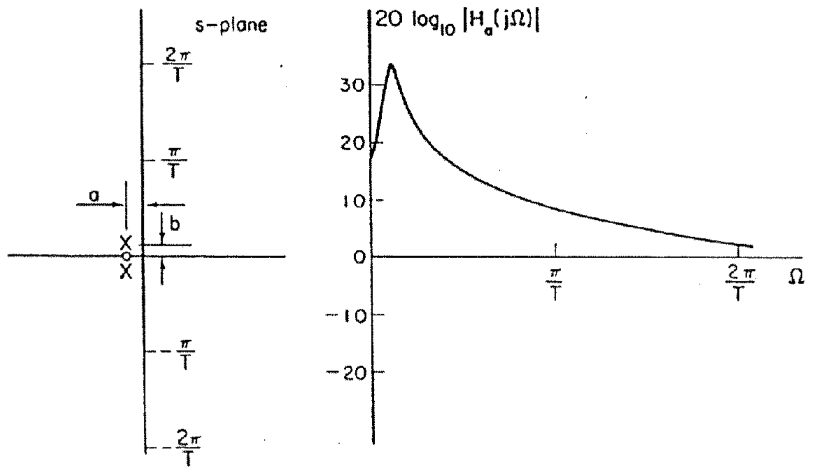


Figure 29: Problem 223: analog s-plane pole-zero-plot in left, and the amplitude response $|H(j\Omega)|$ in right. $\Omega = 2\pi f$ (rad/s), $\omega = 2\pi(\Omega/\Omega_T)$ (rad), where Ω_T angular sampling frequency.

Advanced Assignments: 300-series

310. (*DL 25.5. 2020, 0–1,* **[B7160]**) Fixed window functions are used for cutting a sequence into length of N. This is a typical case when, for instance, plotting a spectrum of a small time frame of an audio signal.

Some window functions (rectangular, Hamming, Blackman, etc) are discussed in (*Mitra 2Ed Sec. 7.6.4, p. 452 / 3Ed Sec. 10.2.4, p. 532*).

An example of Blackman window

$$w_B[n] = 0.42 + 0.5\cos(\frac{2\pi n}{2M}) + 0.08\cos(\frac{4\pi n}{2M})$$

with $M = 25$ ($N = 51$) is given in Figure 30(a).

In the magnitude response of a window function it can be seen a large *main lobe* around $\omega = 0$ and a series of *sidelobes* with decreasing amplitudes as seen in Figure 30(b). Window functions can therefore be characterized by main lobe width Δ_{ML} (rad), relative sidelobe level A_{sl} (dB), minimum stopband attenuation (dB), and transition bandwidth $\Delta\omega$ (rad), see (*Mitra 2Ed Table 7.2, p. 454 / 3Ed Table 10.2, p. 535*), or Table 6.

Window	$w[n]$, $-M \leq n \leq M$	Length of main lobe Δ_{ML}	Relative side lobe A_{sl}	Mininum stopband attenu-ation	Length of transition band $\Delta\omega$
Rectangular	1	$4\pi/(2M+1)$	13.3 dB	20.9 dB	$0.92\pi/M$
Hann	$0.5 + 0.5\cos(\frac{2\pi n}{2M})$	$8\pi/(2M+1)$	31.5 dB	43.9 dB	$3.11\pi/M$
Hamming	$0.54 + 0.46\cos(\frac{2\pi n}{2M})$	$8\pi/(2M+1)$	42.7 dB	54.5 dB	$3.32\pi/M$
Blackman	$0.42 + 0.5\cos(\frac{2\pi n}{2M}) + 0.08\cos(\frac{4\pi n}{2M})$	$12\pi/(2M+1)$	58.1 dB	75.3 dB	$5.56\pi/M$

Table 6: Problem 310: Properties of window functions.

Signal $x[n]$ is modified by the window $w[n]$

$$x_m[n] = x[n] \cdot w[n]$$

Multiplication in time domain corresponds convolution in frequency domain

$$X_m(e^{j\omega}) = X(e^{j\omega}) \circledast W(e^{j\omega})$$

Consider a case, when $x[n]$ consists of three cosine signals: two with equal amplitudes with similar frequencies and one with small amplitude with different frequency. See example in MyCourses (D11win.m).

Explain why the statements are true or false:

(A) Longer window length N enables more exact frequency resolution

(B) Narrow main lobe width enables more exact frequency resolution

(C) A FIR filter computed using window method has equal transition band length (band between passband and stopband) independent of the type of window function (rectangular, Hamming, etc)

(D) Weak sinusoidals are observed better with Blackman than with rectangular window

```
%% Additional code for plotting window functions
% if not having Mitra's book available

%% Window lengths
N1=50;
N2=512;

%% Window functions
wB1=blackman(N1);
wB2=blackman(N2);
wH1=hamming(N1);
wH2=hamming(N2);
wR1=rectwin(N1);
wR2=rectwin(N2);

%% Time and frequency domain plots of window functions
figure; stem(wB1); title('Blackman-window, N=50');
figure; freqz(wB1,1); title('Blackman-window, N=50');
% and similarly for the rest five cases
```

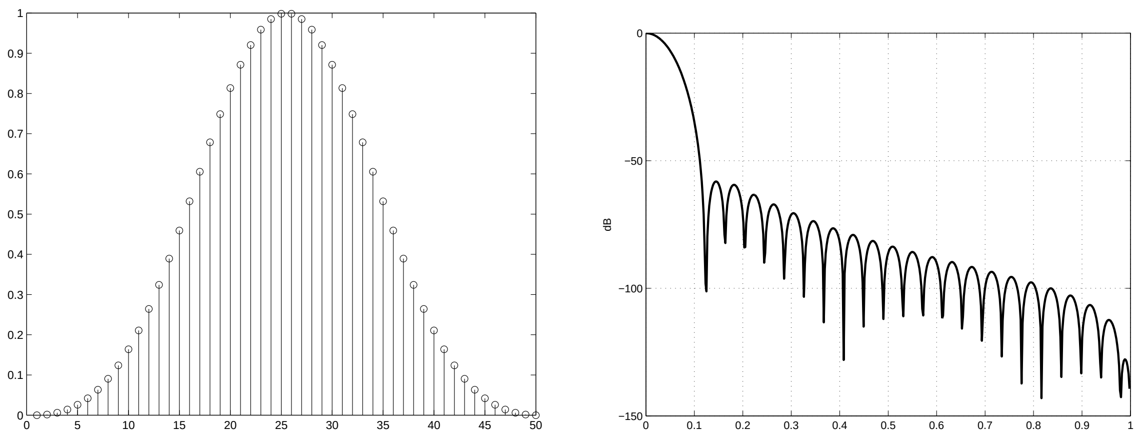


Figure 30: Problem 310: Blackman window, (a) $w[n]$, (b) $|W(e^{j\omega})|$.

311. (*DL 25.5. 2020, 0–2,* [**B9947**]) Write a short essay with some mathematics on *bilinear transform*. Explain the difference of the stability area of analog and digital filters in pole-zero plot. Confirm the transform and its inverse (domains $s \leftrightarrow z$) by computing some points by hand and/or Matlab.

> A group of polish scientist of analogic signal processing went for a visit to USA. When the pilot announced he is going to start landing, the whole group rushed to the other side of the plane. The stewardess got an answer: "No poles on the right side of the plane."

312. (*DL 25.5. 2020, 0–3,* [**B9943**])

In convolution the length of the output signal is the sum of the lengths of the signals that are being convolved minus one. Furthermore, convolution in time domain equals point-wise multiplication in frequency domain.

Consider a situation where an infinite signal $x[n]$ is filtered with a filter $h[n]$ "in real time" using buffers (e.g., 1024 samples), in a way that a part of $x[n]$ is read to the buffer and filtered with $h[n]$, thus getting a part of the output signal $y[n]$.

Implement a Matlab script, which filters a test signal (choose a suitable signal) in blocks, both in time and frequency domain. Different options to implement the filter: using overlap-add and overlap-save methods, using initial and final conditions (`zf`, `zi`) of Matlab's `filter` function, and frequency domain filtering (`fft(x,N)`). Write a short report about your Matlab implementation and your observations. Include also your Matlab script.

> In Mitra "overlap-add" and "overlap-save". For example "filtering in frequency domain" : `http://cnx.org/content/m10257/latest/`.

313. (*DL 25.5. 2020, 0–4,* [**B9948**]) This problem requires Matlab and the following toolboxes and their documentation. Type `doc` and see the contents of Signal Processing Toolbox (SPT), Audio Toolbox (AT), Fixed Point Designer (FPD) and (in the end of the course) DSP System Toolbox (DST).

Choose one topic below. Read the documentation, copy the code into your own `m`-file, run the code given step by step, and verify the results. In the end of code write comments on the demo / example. Include your experience and how it is connected to this DSP course. "Publish" your code in the Matlab Editor by clicking `File - Publish..`, print it, and attach to your returning.

You can get for each topic $0 - 1$ points, and at most four different topics, $0 - 4p$.

Topics:

a) SPT / Demos / Application .. / Single Sideband Modulation via Hilbert transform
b) SPT / Demos / Data F.. / Generating Guitar Chords using the Karplus-Strong Algorithm
c) SPT / Demos / Spectral .. / Measuring the Power of Deterministic Periodic Signals
d) SPT / Demos / Spectral .. / Linear prediction and Autoregressive modeling
e) SPT / Demos / Transforms .. / Discrete Walsh-Hadamard Transform
f) SPT / Examples / Cepstrum and Transforms / Cepstrum Analysis
g) AT / Application d.. / Acoustic Echo Cancellation
h) DST / Application d.. / Adaptive Noise Cancellation using RLS Adaptive Filtering
i) FPD / Demos / Compute Quantization Error

314. (*DL 25.5. 2020, 0–1,* [**BONUS**]) During the course, report up to five *new* errors, mistakes or problems that you have found anywhere in the course's electronic material (e.g., in this booklet or MyCourses) to the thread "Errors in Assignments (+1p bonus)" at MyCourses discussion forum. Check first if the issue is already reported by someone else since no credits will be granted for previously known ones. All kind of new errors are accepted (also typos and major grammar mistakes) although technical problems, major inconsistencies and blind references between tasks are preferred. Copy&paste&print a collection of your corresponding discussion forum messages into your last assignment return.

Before returning the assignments **REMEMBER TO FILL THE SEPARATE A4 COVER PAGE**. That is the first page of this personal assignment collection.