COMP.SGN.100 Introduction to Signal Processing,
Exercise 10, 6.-8.10.2021

Pen & paper task solutions should be submitted to Moodle at least one hour before your exercise session. Matlab tasks are done during the exercise session.

Task 1. *(Pen & paper)* Design a low-pass filter (find out impulse response by pen & paper) using the window design method satisfying the following requirements:

Passband	[0 kHz, 12 kHz]
Stopband	[13.5 kHz, 16 kHz]
Passband ripple	0.1 dB
Minimum stopband attenuation	50 dB
Sampling frequency	32 kHz

Task 2. *(Matlab)* Design a Butterworth type IIR low-pass filter that preserves the frequencies between $0-4000$ Hz and removes the frequencies $7000-20000$ Hz when the sampling rate is 40000 Hz. The minimum stopband attenuation is 45 dB and the maximum passband ripple is 0.3 dB. Plot the impulse response (`impz`), the amplitude response (`freqz`) and the pole-zero plot (`zplane`).

Task 3. *(Matlab)* In the design of IIR filters, the most common method is *bilinear transform* that converts an analog filter into the corresponding digital filter. We implement this algorithm for the low-pass filter case in Matlab in this and the next task. Unfortunately, on this course we cannot get familiar with the theory behind the method, so this may seem like a collection of formulas. At this level, however, it is possible to gain an understanding of the complexity of the algorithm and the basic idea. The algorithm is in outline the following.

1. Convert the filter design requirements of the digital IIR filter to the requirements of the corresponding analog filter by prewarping

$$\begin{aligned}\Omega_p &= 1\\ \Omega_s &= c\tan(\omega_s/2),\end{aligned}$$

where the multiplier c is obtained by the formula

$$c = \frac{1}{\tan(\omega_p/2)}$$

and ω_p and ω_s are the passband and the stopband cut-off frequencies of the digital filter and Ω_p and Ω_s are the corresponding cut-off frequencies of the analog filter. Note that the frequencies in Hertz must first be normalized by the sampling rate and multiplied by 2π to get the normalized angular frequencies ω_p and ω_s.

2. Design the analog filter $H(s)$ that meets the design requirements.

3. Transform the analog filter into a digital one by the bilinear transform, i.e. by substituting for the variable s in $H(s)$ the expression

$$s = c\frac{z-1}{z+1},$$

resulting in the transfer function $H(z)$. The multiplier c was defined in step 1.

We consider these three steps in this and the next task. Create the file `design_lowpass.m` with the first line

```
function [N,D] = design_lowpass(fp, fs, Rp, Rs, Fs)
```

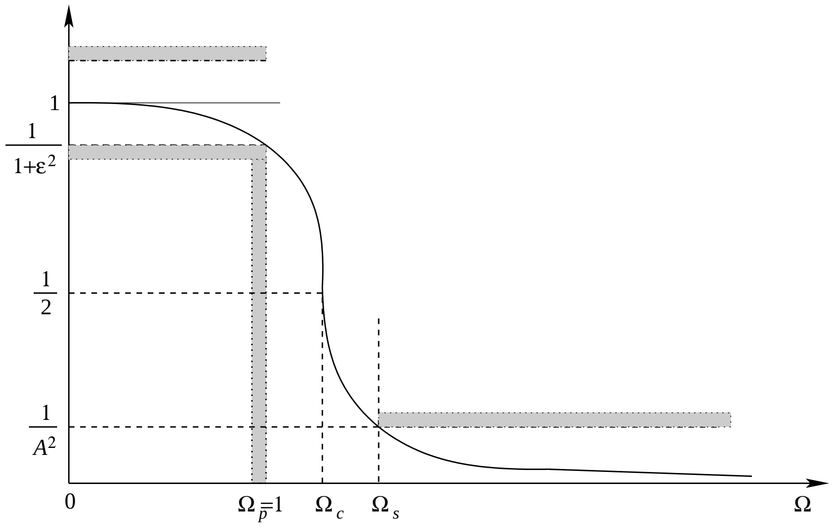
So we define a function whose parameters are the passband cut-off frequency (`fp`, Hertz), the stopband cut-off frequency (`fs`, Hertz), the maximum passband ripple (`Rp`, decibels, positive real number), the minimum stopband attenuation (`Rs`, decibels, positive real number), and the sampling rate (`Fs`). Now we should write this function that has these input parameters and outputs vectors `N` (transfer function numerator) and `D` (transfer function denominator). Below are instructions on how to proceed.

Convert first the digital filter cutoff frequencies to the corresponding cutoff frequencies of the analog filter.

The square of the amplitude response of the analog Butterworth filter is shown in the figure below. We can use the figure to solve the attenuation parameters:

$$\begin{aligned} \epsilon &= \sqrt{10^{\delta_p/10} - 1}, \\ A &= \sqrt{10^{\delta_s/10}}, \end{aligned}$$

where δ_p is the the maximum passband ripple of the digital filter and δ_s is the minimum stopband attenuation of the digital filter . Calculate ϵ and A.



Task 4. *(Matlab)* (Continuation of Task 3) The filter order can be estimated by the formula

$$M = \frac{\log_{10}\left(\frac{A^2-1}{\epsilon^2}\right)}{2\log_{10}\Omega_s}.$$

The result is the order of the transfer function, which must be rounded up to the nearest integer (`ceil`).

The transfer function $H(s)$ of the analog Butterworth filter satisfying the requirements is of the form

$$H(s) = \frac{(-1)^M p_1 p_2 \cdots p_M}{(s-p_1)(s-p_2)\cdots(s-p_M)},$$

where the poles $p_1, p_2, \ldots, p_M$ are obtained from the formula

$$p_k = \frac{1}{\epsilon^{1/M}} e^{\pi i[1/2+(2k-1)/(2M)]}.$$

Calculate the poles in your function and store them in a vector.

When we have the poles of the analog prototype filter, the corresponding digital filter poles are obtained by the bilinear transformation. The formula for the bilinear transformation was

$$s = c\frac{z-1}{z+1}.$$

Now we know the poles of the analog filter (s) and we want to know poles of the digital filter (z). The above formula is thus solved for z:

$$z = \frac{1+s/c}{1-s/c}.$$

Each pole p_k thus maps into the digital filter pole

$$p'_k = \frac{1+p_k/c}{1-p_k/c}.$$

Do the same using Matlab. Use elementwise division "`./`" to obtain the vector of digital filter poles from the vector of analog filter poles.

In the case of a digital low-pass filter, all zeros are at the point $z = -1$. Create a vector of these (of length M) and then use the command `zp2tf` to obtain the transfer function. The input parameters of the command are zeros, poles and gain at zero frequency. At this point, let the gain have value one.

Now you have the transfer function coefficients (the numerator `N` and the denominator `D`). Since the gain was set to one, it is probably incorrect. It can be found that the true gain expressed in Matlab syntax is `K=sum(N)/sum(D)`. The final result is now obtained by dividing the vector `N` by number `K`. The function automatically returns the vectors `N` and `D`.

Task 5. *(Matlab)* Design a low-pass filter using the function of the previous tasks and satisfying the following requirements.

Passband	[0 kHz, 9 kHz],
Stopband	[12.5 kHz, 16 kHz],
Passband ripple	0.4 dB,
Minimum stopband attenuation	25 dB,
Sampling frequency	32 kHz.

Plot the pole-zero plot as well as the amplitude response.