Norwegian University of Science and Technology
Department of Electronics and Telecommunications

TTT4120 Digital Signal Processing
Problem Set 4

Problem 1 (2 points)

Given a filter with transfer function

$$H(z) = \frac{1}{1 - az^{-1}}$$

(a) Draw the pole-zero plot for the filter given $a = 0.9$ and $a = -0.9$.

Determine the filter type for two filters? Explain using the pole-zero plot.

(b) Verify the results in $1(a)$ with *pezdemo*. The demo can be downloaded from the course home page.

Problem 2 (2 points)

Consider a causal digital filter with transfer function

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

(a) Find the transfer function of the inverse filter of $H(z)$.

(b) Is the inverse filter stable? Justify the answer.

(c) Is the inverse filter a minimum-phase filter?

(d) Does the inverse filter have a linear phase characteristics? Justify your answer.

Problem 3 (2 points)

In the recording/mastering of sound signals or during playback, it is often desired to alter the characteristics of the sound at different frequencies. For example, we may wish to highlight the lower/middle frequencies, while we may wish to reduce the presence of high frequencies.

This can be done by using so-called "shelving" filters. Figure 1 shows a low-frequency shelving filter implementation. The filter $A(z)$ is :

$$A(z) = \frac{\alpha - z^{-1}}{1 - \alpha z^{-1}}$$

The parameters α and K are used to *tune* the filter.

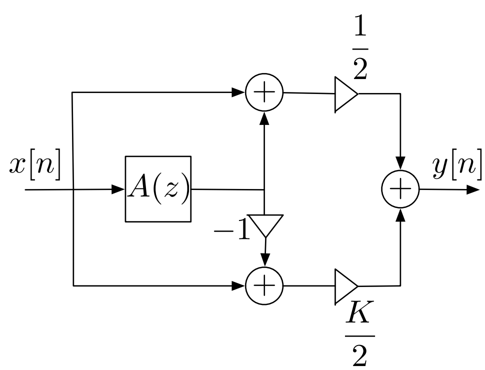

Figure 1: Low-frequency shelving filter

(a) What type of filter is $A(z)$, (Highpass, Lowpass, Bandpass, Bandstop or Allpass)? Justify your answer.

(b) The filter in Figure 1 consists of a sum of two branches (upper and lower).

- Use the Matlab function `freqz` or the Python function `scipy.signal.freqz` to plot the magnitude responces of the two branches given $\alpha = 0.9$ and $K = 1$.
- What types of filters do the upper and lower branches represent?

(c) The Matlab-script `LFshelving.m` and the Python-script `LFshelving.py` implement the entire filter in Figure 1 and plot its magnitude response. Furthermore, they use the filter to modify the music file `pluto.wav` and play both the original and modified music file.

- Let $K = 3$. Plot the magnitude response of the filter and listen to the original and modified music file when α is equal to 0.5, 0.7 and 0.9, respectively.

- Let $\alpha = 0.7$. Plot the magnitude response of the filter and listen to the original and modified music file when K is equal to 0.5, 1 and 4, respectively.
- What do the parameters K and α control?

Problem 4 (4 points)

Given a sequence $d[n]$ as:

$$d[n] = A_x \cos(2\pi f_x n) + A_y \cos(2\pi f_y n), \quad 0 \leq n \leq L-1$$

where $A_x = A_y = 0.25$, $f_x = 0.04$, $f_y = 0.10$ and $L = 500$.

The sequence $d[n]$ is contaminated with additive noise $e[n]$, that is, the observed signal is

$$g[n] = d[n] + e[n].$$

(a) Use Matlab or Python to generate and plot sequences $d[n]$ and $g[n]$ and their magnitude spectra, $|D(f)|$ and $|G(f)|$. (Use FFT length N=2048) A segment of the noise $e[n]$ of length L can be generated by the Matlab command `randn(1,L)` or by the Python command `np.random.normal(size=L)`.
Compare the plots before and after adding the noise.

(b) To isolate the two sinusoids from the noisy signal $g[n]$ we want to design two digital resonators with transfer functions $H_x(z)$ and $H_y(z)$. The resonators should have zeros at $z = 1$ and $z = -1$. Use common sense to figure out how close to the unit circle the poles should be.

- Write the expressions for $H_x(z)$ and $H_y(z)$.
- Read about the Matlab functions `poly`, `roots`, `zplane` and `freqz` or the Python functions `np.poly`, `np.roots` and `scipy.signal.freqz`.
- Plot the zeros and poles of the resonators. Use the Matlab function `zplane` or the Python function `np.roots` to calculate the poles and zeros, and then you can use the following code to plot them on the Z-plane:

```
fig, ax = plt.subplots()

# plot circle
theta = np.linspace(-np.pi, np.pi, 1000)
ax.plot(np.sin(theta), np.cos(theta), '--k')
ax.set_aspect(1)

# plot poles and zeros
ax.plot(np.real(poles),np.imag(poles),'Xb',label='Poles')
```

```
ax.plot(np.real(zeros),np.imag(zeros),'or',label='Zeros')
ax.set_xlabel('Real part')
ax.set_ylabel('Imaginary part')
```

- Use the Matlab function `freqz` or the Python function `scipy.signal.freqz` to plot $|H_x(f)|$ and $|H_y(f)|$.

(c) Use the two filters designed in 2b) to filter the noise contaminated signal $g[n]$ (use the Matlab function `filter` or the Python function `scipy.signal.lfilter`)

Plot the outputs from the filters $q_x[n]$ and $q_y[n]$ as well as their amplitude spectra $|Q_x(f)|$ and $|Q_y(f)|$.

Are the resulting plots what you expected?

(d) We wish to combine the two digital resonators in order to isolate both sinusoids.

- Plot the magnitude response of the resulting system.
- Find its zeros and poles. (Hint. You can use the functions `poly` and `roots`)
- Plot the zeros and poles on the Z-plane, and discuss their placement.
- Plot the output from the combined filter, and the its magnitude spectra.
- Compare the plots with the plots of $d[n]$ and $g[n]$ and their magnitude spectra.