Lab 4

Discrete-Time Signals and Systems and the Z-Transform

Exercise 1 (4 points)

Matlab filename must be `exer01.m`.
Consider the following difference equation of a LTI system:

$$y[n] + 0.8 \cdot y[n-1] + 0.01 \cdot y[n-2] = x[n] - x[n-1]$$

Make use of Matlab and the Z-Transform and its properties to answer the following questions.

1. Obtain analytically the response $y[n]$ of the relaxed system to a unit step $u[n]$. Write the equation in analytical form using real values.
2. Plot your answer and compare it with the output of Matlab's `filter()` function.
3. Repeat the same steps when the initial conditions are set to $y[-1] = 1$ and $y[-2] = 2$. Remember to use the `filtic()` function to calculate the value of the variable that represents the initial conditions of the `filter()` function.
4. Calculate the impulse response $h[n]$ of the system analytically (again using real values) and using the `filter` function.
5. Discuss the stability of the system.

Exercise 2 (3 points)

Matlab filename must be `exer02.m`.
Determine the Inverse Z-Transform, the Region of Convergence (ROC) and plot the Pole/Zero Diagram of the following Z-Transforms.

1. $X(z) = \dfrac{z^2}{2z^2 - z + 3}$, Assume $x[n]$ is anti-causal

2. $X(z) = \dfrac{z^{-2}}{(1-0.9z^{-1})(1+0.65z^{-1})(1+0.7z^{-1})^2}$, Assume $x[n]$ is non-causal

3. $X(z) = \dfrac{1+2z^{-1}}{1-0.8z^{-1}+0.3z^{-2}}$, Assume $x[n]$ is causal

NOTE: The function `residuez()` can be used to obtain the poles and zeros (read carefully the Matlab help for `residuez()`. Check also the `zplane()` function for further inspiration).

Exercise 3 (3 points)

From an audio signal a fragment is extracted. This fragment has a length of 1 second and gets degraded with additive white gaussian noise. The position of the fragment inside the original audio signal is unknown. You are asked to develop an algorithm that estimates in which position (in units of *time*) of the audio signal the fragment starts. For this, you are given the function *degrade_fragment* whose declaration is as follows:

```
[original_signal, fs, noisy_signal] = degrade_fragment(student_id_number)
```

where *original_signal* is the original audio signal, *fs* is the sampling frequency and *noise_signal* is the noisy fragment. The function's only parameter is your student identification number and requires the sample file *audio_file.wav* to be in Matlab's workspace.