UNIVERSITY OF HAMBURG, DEPARTMENT OF INFORMATICS, SIGNAL PROCESSING GROUP

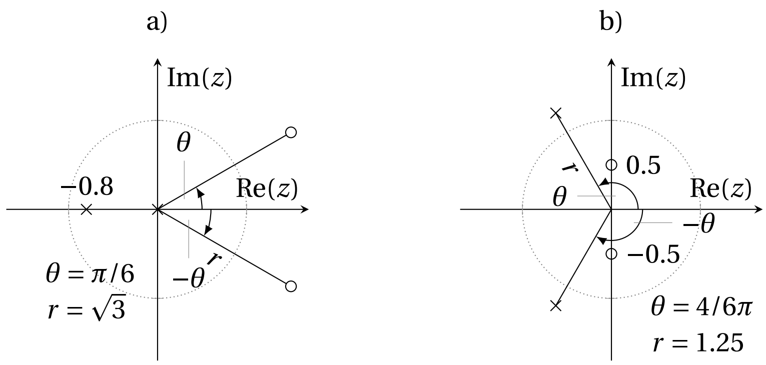
Digital Media Signal Processing — Assignment V

Robert Rehr and Lasse Vetter

May 22, 2017

1 LTI SYSTEMS AND THE z-TRANSFORM

Consider the following pole-zero plots describing causal linear time-invariant systems.



For each diagram, solve the following tasks:

1. Is the system stable?
2. Determine the system's difference equation under the assumption that $H(z)|_{z=1} = 1$.
3. Sketch the realization of the system in direct form II.

2 RELATIONSHIP BETWEEN THE z-TRANSFORM AND FREQUENCY

Consider the following z-transforms of a causal impulse response $h(n)$.

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

For all z-transforms, perform the following tasks:

- Compute the poles and zeros of the z-transform and sketch them in a pole-zero plot. What is the phase θ and the absolute value r of the poles? Are these systems stable?
- Describe the system by a difference equation. Use the difference equation to the compute impulse response $h(n)$ for $0 \le n \le 8$. For this, assume that the system is relaxed. Is the $h(n)$ periodic? If yes, what is the period length?
- Use the z-transform pairs to determine the complete impulse response $h(n)$. (For 1.2, consider pair 7 on slide 30 of Chapter 3). What is the relationship between θ, the previously computed period length, and the obtained impulse response?

3 From an Ideal Low-pass Filter to its Digital Implementation

We have already applied different impulse responses using convolution. Here we want to derive the impulse response for an ideal low pass filter. We then want to discretize the theoretical impulse response, truncate it, and implement it as an FIR filter. Ultimately, we will reduce the introduced artifacts, which are a byproduct of the truncation, by windowing.
NOTE: While here we get a first intuition of the concept of windowing, the theory will be covered later in the lecture.

1. Follow the outlined procedure to derive the ideal low pass filter:
   - Design a low pass prototype in the continuous frequency domain, as shown in Figure 3.1.
   - Derive the impulse response by applying the inverse Fourier transform.
   - Find a discrete time description for the impulse response, so we can apply it in the discrete time domain. Note that we need to scale the sampled impulse response by the factor $g = 1/F_s = T_s$, to preserve the signal's energy.
2. Use the given IPython notebook and enter the expression you found in 1. It's recommended to use numpy's sinc function, as it accounts for the discontinuity at $n = 0$. CAUTION! The numpy sinc function is defined as $\text{sinc}(n) = \sin(\pi n)/(\pi n)$. click through to rest of the notebook to evaluate the low pass.
3. Try different lengths for the impulse response by changing the `NSAMPLES` variable.
4. Choose different cutoff frequencies and check if they correspond correctly to your calculations.

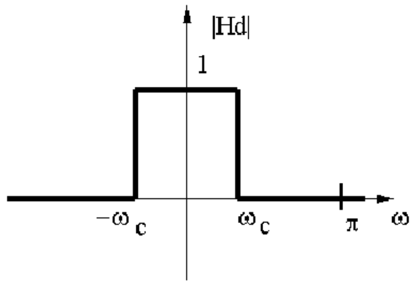


Figure 3.1: Ideal low pass filter transfer function

4 Biquad Implementation

The Biquad transfer function is given by

$$H(z) = \frac{b_0 + b_1 z^{-1} + b_2 z^{-2}}{1 + a_1 z^{-1} + a_2 z^{-2}}.$$

To implement the filter:

1. Derive the difference equation.
2. Draw the block diagram in direct form II.
3. Implement a class `Biquad` in Python, so you can process on a sample by sample basis. Your class should have a function of the form `y = process(x)` where `x` is a single input sample and `y` a single output sample. Make sure you handle the storage of the previous samples.
4. How could you test your implementation?