ECE2170: Digital Filtering on the Red Pitaya

# Goals

## Observe behavior for multiple digital filters implemented via the Red Pitaya

# Setup

## Background

### LTI systems in the z-domain

Reminder all discrete time LTI systems can be represented in a canonical form of:

Where the coefficients of each expression can be viewed as the coefficients in the difference equation:

## Hardware configuration

This configuration will require an additional piece of equipment, a second red pitaya. One red pitaya will be used as in the oscilloscope/signal generator or the spectrum analyzer modes, while the other will be used in the LTI DSP workbench. Connect the red pitayas such that the IN1 of the LTI DSP device is connected to OUT1 of the generator. You can also use a T-joint to connect the OUT1 of the generator board to IN1 of itself to see the response of the circuits more clearly and to measure the Frequency response with the Bode analyzer. Diagram

Description automatically generated

# Tasks

## All-Pass Filter – Delay Element

Enter the following transfer function into the LTI workbench:

This is accomplished by setting . This kind of filter is called an all pass filter, due to its input/output relation of simply passing the output. Note that this is a special class of all-pass filter, namely a delay filter. This kind of filter purely provides a delayed version of the input as it’s output.

### Measurement

1. Note down the Bode plot from the LTI workbench for both gain and phase(Note for some filters, you may want to do this over two screenshots, as the magnitude and phase are plotted on a common axis, even if one is in dB and the other is in degrees), and describe what happens as the frequency of the input signals are increased or decreased. (Hint: refer to the frequency response from the LTI workbench)
2. Show the resulting waveform in the Red Pitaya/Spectrum Analyzer scope object for
   1. A sine wave of 1kHz is applied
   2. A square wave of 1kHz is applied
   3. A triangle wave of 1kHz is applied

### Analysis

1. Is the transfer function

also an all pass filter? Why or why not?

1. If I wanted to attenuate the incoming by 50% (multiply by 0.5) what would the general all-pass filter function be?
2. Write out the difference equation for a general all pass filter.

## Moving average filter

An -tap moving average filter has the form:

This is accomplished by setting where . This kind of filter is called a moving average or boxcar filter, due to its nature of taking a local average of samples at every sample point. Oftentimes the size of is known as the window size.

### Measurement

1. Note down the Bode plot from the LTI workbench for both gain and phase for the provided value of , describing what happens as the frequency of the input signals are increased or decreased.

   2. What are the trends as gets larger?
2. Show the resulting waveform in the Red Pitaya/Spectrum Analyzer scope object for:
   1. A sine wave of 1kHz is applied
   2. A square wave of 1kHz is applied
   3. A triangle wave of 1kHz is applied

### Analysis

1. What class of filter does this look like? (high-pass, low-pass, band-pass, band-stop)
2. What does the window size say about the filter’s performance?
3. Write out the difference equation of this filter.

## Low pass filter

Enter the following transfer function into the LTI workbench:

### Measurement

1. Note down the Bode plot from the LTI workbench for both gain and phase. (Note for some filters, you may want to do this over two screenshots, as the magnitude and phase are plotted on a common axis, even if one is in dB and the other is in degrees)
2. Show the resulting waveform in the Red Pitaya/Spectrum Analyzer scope object for:
   1. A sine wave of 1kHz is applied
   2. A square wave of 1kHz is applied
   3. A triangle wave of 1kHz is applied
3. Describe what happens as the frequency of the input signals are increased or decreased.

### Analysis

1. Write out the difference equation of this filter.
2. In the previous lab, we showcased that low-pass filters can be used to approximate integral operations. At what frequency does this filter do a passable job of implementing this operation?

## 1st difference filter

Enter the following transfer function into the LTI workbench:

This is accomplished by setting .

### Measurement

1. Note down the Bode plot from the LTI workbench for both gain and phase, and describe what happens as the frequency of the input signals are increased or decreased.
2. Show the resulting waveform in the Red Pitaya/Spectrum Analyzer scope object for:
   1. A sine wave of 1kHz is applied
   2. A square wave of 1kHz is applied
   3. A triangle wave of 1kHz is applied

### Analysis

1. What does removing the common factor of do to the filter? Why do you think the factor of was included?
2. Write out the difference equation of this filter.
3. In the previous lab, we showcased that high-pass filters can be used to approximate derivative operations. At what frequency does this filter do a passable job of implementing this operation?

## Feedback

Enter the following transfer function into the LTI workbench:

### Measurement

1. Note down the Bode plot from the LTI workbench for both gain and phase. (Note for some filters, you may want to do this over two screenshots, as the magnitude and phase are plotted on a common axis, even if one is in dB and the other is in degrees)
2. Show the resulting waveform in the Red Pitaya/Spectrum Analyzer scope object for
   1. A sine wave of 1kHz is applied
   2. A square wave of 1kHz is applied
   3. A triangle wave of 1kHz is applied
3. Describe what happens as the frequency of the input signals are increased or decreased.

### Analysis

1. Write out the difference equation of this filter.
2. ~~In the previous lab, we showcased that low-pass filters can be used to approximate integral operations. At what frequency does this filter do a passable job of implementing this operation?~~