ECE2170: FIR vs. IIR filters - Red Pitaya

# Goals

## Observe the characteristic behaviors for Finite Impulse response (FIR) and Infinite Impulse response (IIR) filters implemented via the Red Pitaya

# 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:

Depending on the configurations of the transfer function, the filter can show various behaviors. This lab will demonstrate the differences between two large classes of filters, Finite Impulse Response (FIR), and Infinite Impulse Response (IIR).

## 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

## Running Sum Filter

This is a classical FIR filter that operates with the following difference equation

This has the z domain transfer function of simply the sum each delay element multiplied by unity.

Where the final transfer function expression is simply the expansion of the M-th partial sum of a geometric series for .

### MATLAB Analysis

1. In the provided matlab script, vary the number of taps (value of ) for the running sum filter and comment on as to how the:
   1. Impulse response changes
   2. Frequency response changes
   3. Pole zero plot behaves
2. If we assign all , this is the moving average filter from the previous lab, how does this valuing of the values change:
   1. Impulse response
   2. Frequency response
   3. Pole zero plot

### Red Pitaya

In the red pitaya’s LTI workbench, we can construct arbitrary transfer functions using the coefficients where with the caveat that . Expanding the transfer function of the running sum filter to accommodate this maximal number of taps yields the following transfer function

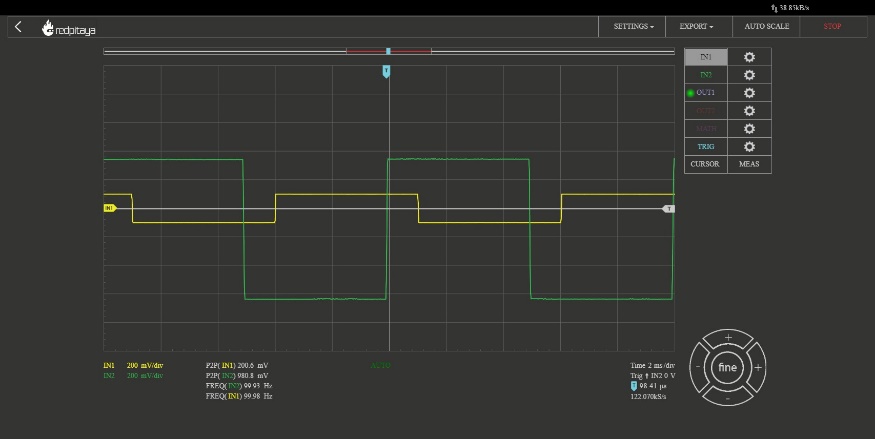
This shows that all values are 1, and that .

1. Plot the frequency response of this filter when entered into the red pitaya LTI workbench.

Chart

Description automatically generated with medium confidenceChart, line chart

Description automatically generated

1. Show to resulting filtered waveforms/spectra to a:
   1. Square wave within the filter bandwidth
   2. 
   3. Square wave outside of the filter bandwidth

Diagram

Description automatically generated

## Resonant Filter

This is a classical IIR filter that operates with the following difference equation

Which describes a feedforward of the input with a delayed version of the output. Intuitively, for periodic signals, this implies that the filter will, when supplied a signal with fundamental period will have reinforcing effect, whereby each of the previous peaks of a signal will be summed with the current peak of the signal to provide large gain at this specific frequency. This behavior is known as resonance, and is commonly used to design many kinds of filters. Mapping this to the z-domain provides the following equation:

Which after some algebra, provides the transfer function of:

### MATLAB Analysis

1. In the provided matlab script, vary the feedback delay (value of ) for the resonant filter and comment on as to how the:
   1. Impulse response changes
   2. Frequency response changes
   3. Pole zero plot changes

### Red Pitaya

In the red pitaya’s LTI workbench, we can construct arbitrary transfer functions using the coefficients where with the caveat that . Expanding the transfer function of the resonant filter to accommodate this maximal number of taps yields the following transfer function

This shows that , and that and that and .

1. Plot the frequency response of this filter when entered into the red pitaya LTI workbench.

Chart

Description automatically generated

Chart, line chart

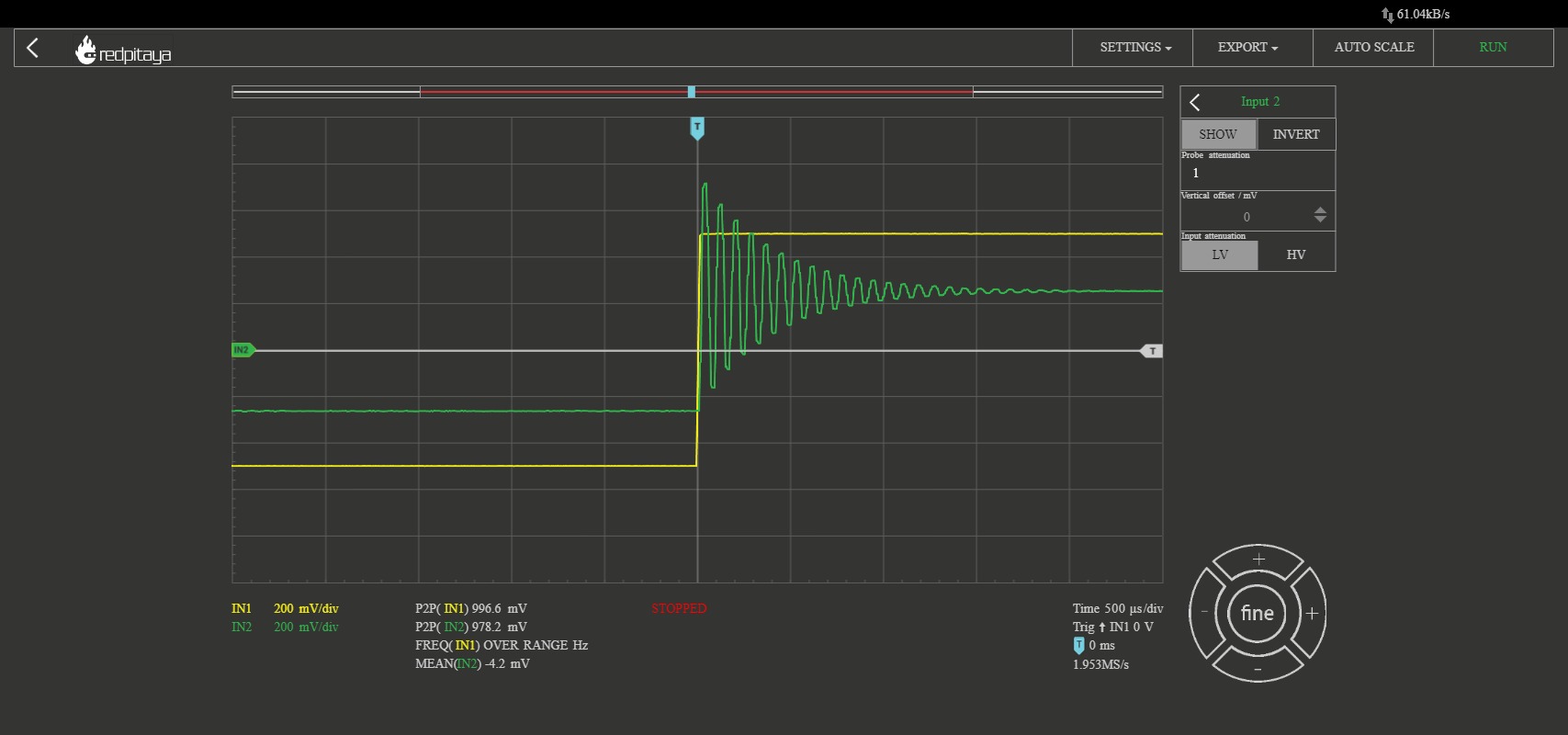
Description automatically generated

1. Show to resulting filtered waveforms/spectra to a:
   1. Square wave within the filter resonance

Chart

Description automatically generated

* 1. Step Response of the filter outside of the resonance



## Filter Cascade

As mentioned previously, cascading two filters is described simply by multiplying their transfer functions.

Perform analysis on the resulting cascaded filter where are the running sum filter with 6 taps (, and the resonant filter with order 6 (.

### MATLAB Analysis

Using the previous two filter transfer function in MATLAB, calculate the result of cascading the filters.

1. Calculate the result of cascading the filters.
2. Write out the resulting transfer function
3. Plot and comment on the:
   1. Impulse response shape w.r.t either of the before filters
   2. Frequency response w.r.t either of the before filters
   3. Pole zero plot w.r.t either of the before filters

### Red Pitaya

Enter the previously calculated transfer function into the Red Pitaya.

1. Plot the frequency response of this filter

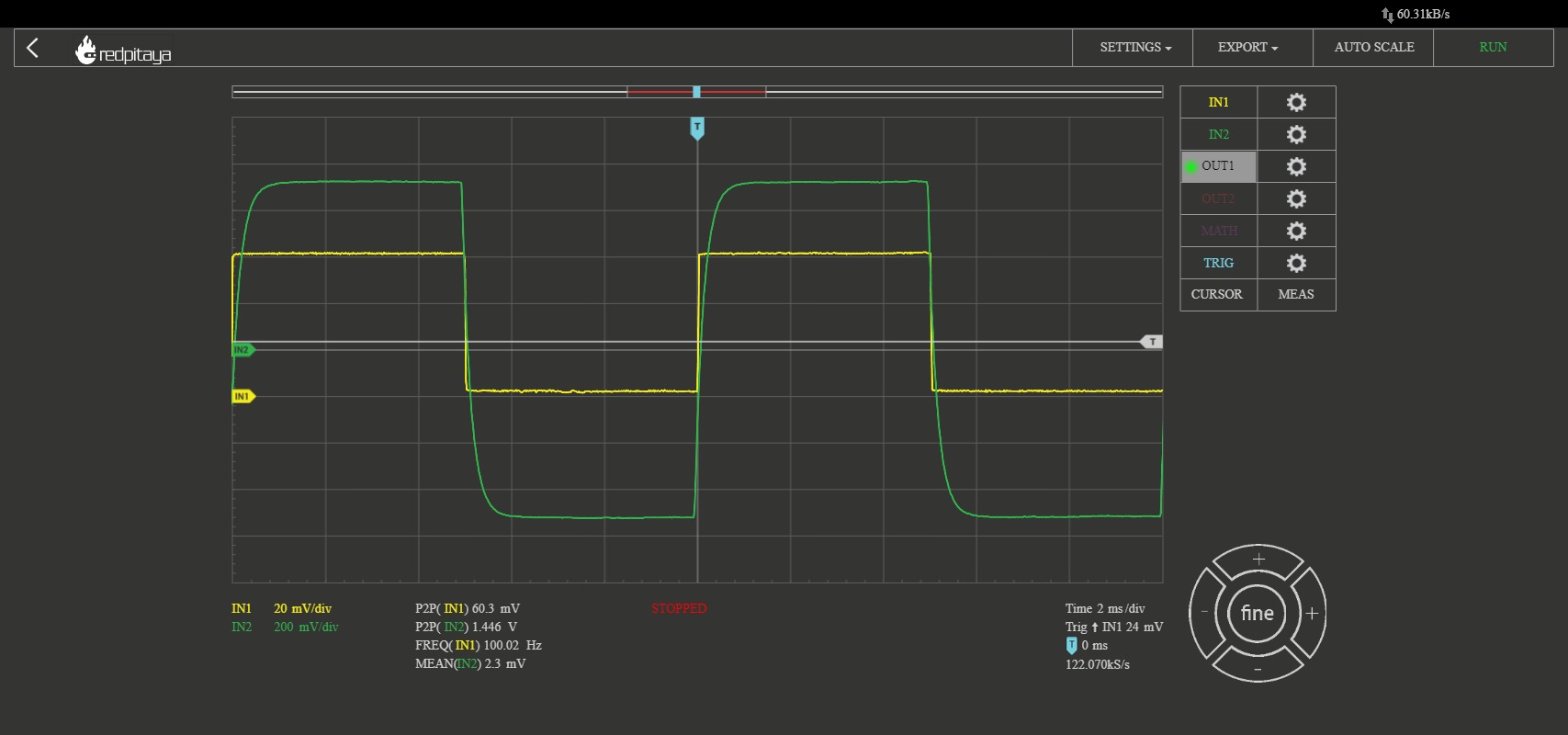
Chart, line chart

Description automatically generated

Chart, line chart

Description automatically generated

1. Show to resulting filtered waveforms/spectra to a square wave at:
   1. Square wave within the filter bandwidth



* 1. Square wave outside of the filter bandwidth

