Digital Signal Processing

Lab 09 Introduction
Higher Order Recursive Filters

Lab 09 Objectives

- Design and implement higher order recursive filters using MATLAB tools
- Use the DFT (in the form of the FFT) to plot frequency response given an impulse response
- Explore stability issues with very high order filters using the direct form implementation

Lab 09 Objectives

- Introduce the use of 2nd order stages to improve stability of higher order filters

- Design a Chebyshev bandpass filter. Implement using second order stages

Lab 09 Steps

- Design and implement
  - Butterworth Low pass filter
  - Chebyshev Low pass filter
  - Chebyshev High pass filter
  - Investigate stability of high order filters
  - Higher order filter as second order stages
  - Bandpass Chebyshev filter
  - Test bandpass filter with swept sine waves

Higher Order Recursive (IIR) Filters

- Adding poles and zeros to the transfer function can improve the frequency response of the filter

Zeros are the roots of the numerator

$$H(z) = \frac{\sum_{k=0}^{M-1} a_k z^{-1}}{1 - \sum_{k=0}^{N-1} b_1 z^{-1}}$$

Poles are the roots of the denominator

Implementing the High Order Recursive Filters

- Recursive filters compute the next output using:
  - The filter input values $x[n], x[n-1]$ ...
  - *<u>and</u>* the past filter output values $y[n-1], y[n-2]$, ...

- The difference equation is:

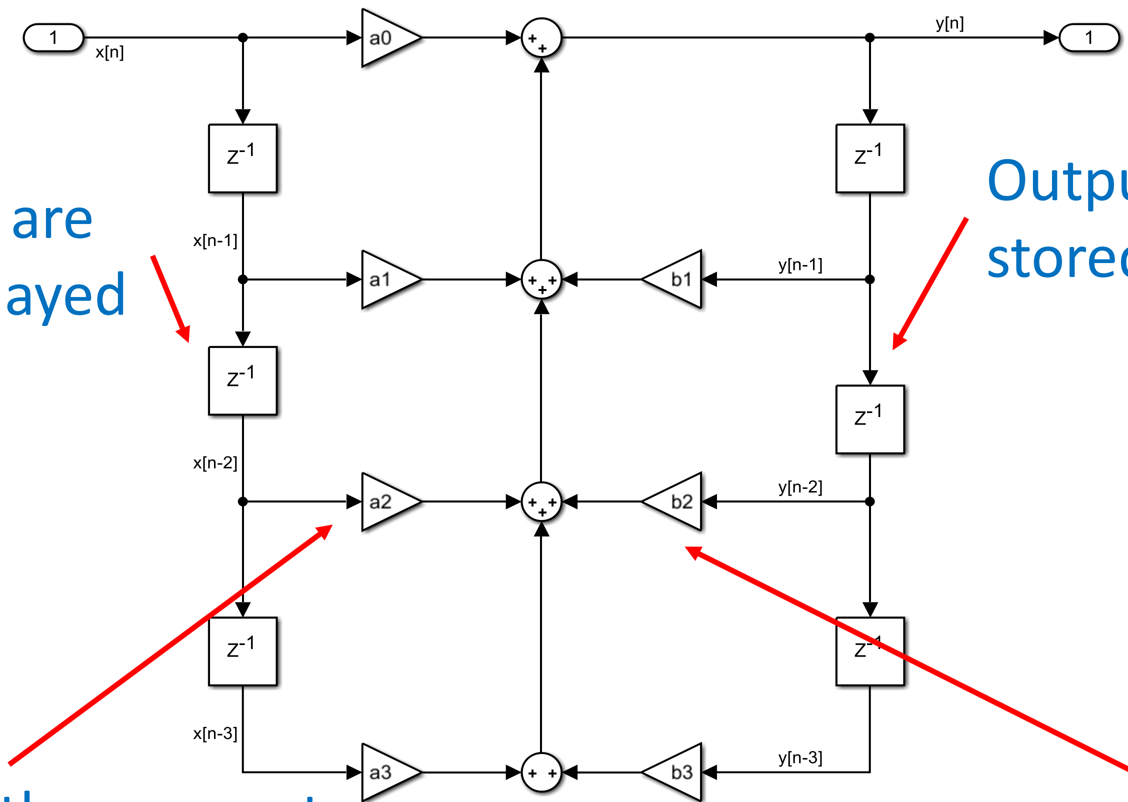
$$y[n] = a_0 x[n] + a_1 x[n-1] + a_2 x[n-2] + a_3 x[n-3] + \cdots$$
$$b_1 y[n-1] + b_2 y[n-2] + b_3 y[n-3] + \cdots$$

- This is referred to as the direct form of the IIR

High Order Recursive Filters Block Diagram

Input samples are stored and delayed

Output samples are stored and delayed



Multiplied by the numerator coefficients

Multiplied by the denominator coefficients

Using MATLAB to Find Filter Coefficients

- There is a MATLAB routine that is available to find the recursion equation coefficients
  - <IIR_Designer.mlx>

- Located in MyCourses in the Class and Lab sections
  - MATLAB Tools\Filter Design Tools

How to Use the MATLAB Code

Typically the breathing rate system is using a sampling rate of $10Hz$ or $600BPM$.

samplingFreq 600

Choose the filter type (Butterworth or Chebyshev) and the filter topology (LPF, HPF, BPF, BSF)

Sample rate set to 600 bpm

filterType Butterworth

filterTopology Lowpass

An additional parametr for the Chebyshev filter is the amount of ripple in the passband. If Chebyshev is the filter type selected then set the amount of ripple in decibels (dB).

rippleDb 0.5

Define the filter type and topology

The order of the filter will determine the number of poles in the filter. Th ... ases the filter com

If Chebyshev define the ripple (dB)

filterOrder 5

Choose the corner frequency (sometimes called the cutoff frequency). This determines the location of the passband of the filter. In the case of the lowpass filter you only need to sp frequency and the upper corner frequency. The figures below describe these values. The units for the corner frequency will be the same as the sampling frequency.

If the filter is a LPF or HPF, enter the corner frequency below

Set the filter order

cornerFreq 70

If the filter is a BPF or BSF, enter both the upper and the lower corner frequecies below

lower_bpf_bsf_cornerFreq 50

upper_bpf_bsf_cornerFreq 80

How to Use the MATLAB Code

Typically the breathing rate system is using a sampling rate of $10Hz$ or $600BPM$.

samplingFreq 600

Choose the filter type (Butterworth or Chebyshev) and the filter topology (LPF, HPF, BPF, BSF). Here are some examples of various filter types and topologies.

filterType Butterworth

filterTopology Lowpass

An additional parametr for the Chebyshev filter is the amount of ripple ... dB).

If LPF or HPF set the corner frequency (same units as sampling frequency)

rippleDb 0.5

The order of the filter will determine the number of poles in the filter. The gre... lter com...

If BPF or BSF set the upper and lower Corner frequency (same units as sampling frequency)

filterOrder 5

Choose the corner frequency (sometimes called the cutoff frequency). This ... ed to sp... frequency and the upper corner frequency. The figures below describe thes...

If the filter is a LPF or HPF, enter the corner frequency belo...

cornerFreq 70

If the filter is a BPF or BSF, enter both the upper and the lower corner frequecies below

lower_bpf_bsf_cornerFreq 50

upper_bpf_bsf_cornerFreq 80

IIR Designer Results

Here is the z-transform of the filter:

```
Tfilt =

  0.00246 + 0.0123 z^-1 + 0.0246 z^-2 + 0.0246 z^-3 + 0.0123 z^-4 + 0.00246 z^-5
  -------------------------------------------------------------------------------
       1 - 2.641 z^-1 + 3.12 z^-2 - 1.954 z^-3 + 0.641 z^-4 - 0.08709 z^-5

Sample time: 0.1 seconds
Discrete-time transfer function.
```

Z-transform

The complex Zeros of the transfer function (roots of the numerator) are shown below:

```
zeros_b = 5×1 complex
   -1.0008 + 0.0006i
   -1.0008 - 0.0006i
   -0.9997 + 0.0009i
   -0.9997 - 0.0009i
   -0.9990 + 0.0000i
```

Complex zero-locations

The complex Poles of the transfer function (roots of the denominator) are shown below:

```
poles_a = 5×1 complex
    0.6158 + 0.5273i
    0.6158 - 0.5273i
    0.4821 + 0.2552i
    0.4821 - 0.2552i
    0.4452 + 0.0000i
```

Complex pole-locations

The complex Poles of the transfer function can also be expressed in terms of frequency by computing the magnitude of the complex poles:

```
pole_wn = 5×1
    0.8107
    0.8107
    0.5455
    0.5455
    0.4452
```

Pole-locations in terms of frequency

Butterworth Filter

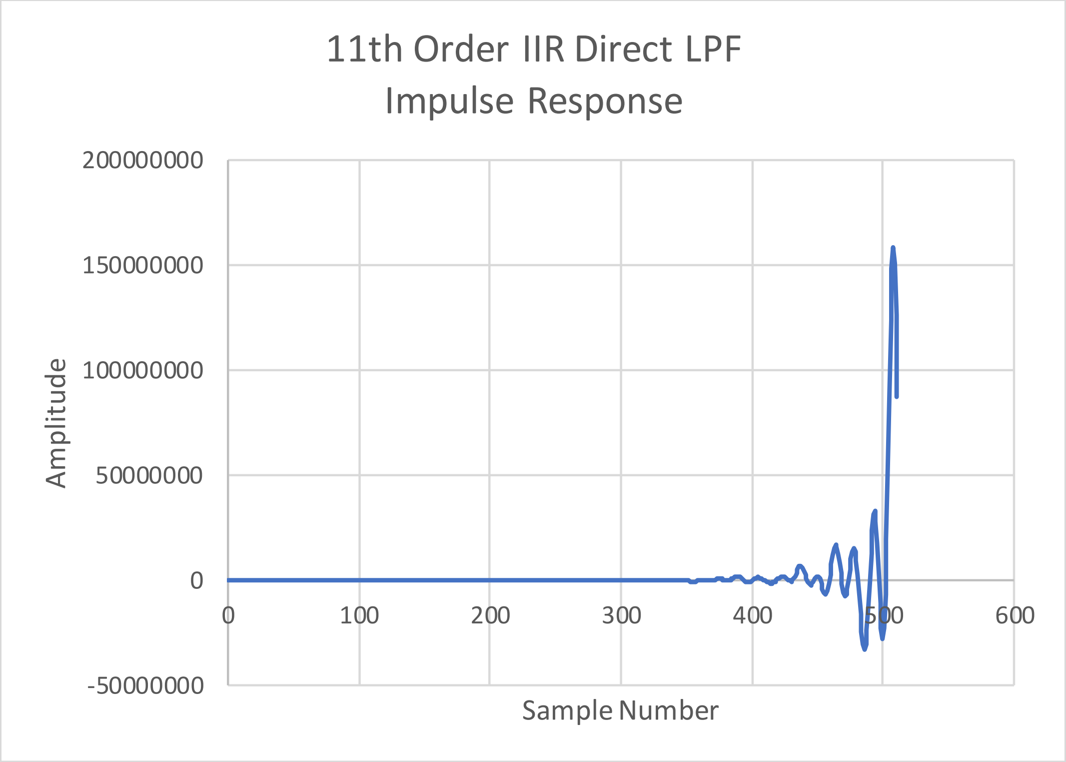
- C-Code variable declarations printed in the command window
- Copy to the Arduino IDE

```
****  C Code for Direct IIR Filter -- Copy to Arduino IDE ****

 // BWRTH HIGH, order 9, 12 BPM
 const int MFILT = 10;
 static float GAIN = 87.7188;
 static float b[] = {0.0079365, -0.0714286, 0.2857143, -0.6666667, 1.0000000, -1.0000000, 0.6666667, -0.2857143, 0.0714286, -0.0079365};
 static float a[] = {1.0000000, -8.2763713, 30.4708077, -65.4968459, 90.5808889, -83.5828216, 51.4579469, -20.3816702, 4.7127337, -0.4846682};
```

High Order Filter Impulse Response – IIR Direct

- 11^{th} Order Chebyshev LPF



Filter becomes unstable and impulse response grows without bounds

How Can we Make the Filter Stable?

- Factor the higher order transfer function into quadratic polynomials

$$H(z) = \frac{\sum_{k=0}^{M-1} a_k z^{-1}}{1 - \sum_{k=1}^{N-1} b_1 z^{-1}}$$

Second Order Section #1

Second Order Section #2

$$H(z) = \boxed{g_1 \frac{1 + a_{11}z^{-1} + a_{12}z^{-2}}{1 + b_{11}z^{-1} + b_{12}z^{-2}}} \times \boxed{g_2 \frac{1 + a_{21}z^{-1} + a_{22}z^{-2}}{1 + b_{21}z^{-1} + b_{22}z^{-2}}} \times$$

$$\boxed{g_3 \frac{1 + a_{31}z^{-1} + a_{32}z^{-2}}{1 + b_{31}z^{-1} + b_{32}z^{-2}}} \times \cdots$$

Second Order Section #3

Continues until all sets of 2 poles and zeros are accounted for. Could have an additional First Order Section

Lab 09 Duration and Write Up

- This lab is a 1-week lab

- It does require an IEEE format write up
  - <u>Be sure to include all plots and answers to questions and observations</u>

  - <u>In your conclusion write about how you might use filters in your breathing rate detection system</u>