A blue and white logo

Description automatically generated with low confidence

System ID Assignment

Date of Submission: October 27th 2021

By

Andrew North & Taylor Ayars

ME520 Sensor Processing with Applications

Embry-Riddle Aeronautical university - Daytona Beach

# 1.0 System and Setup

## 1.1 Step 1

The dynamic system we are identifying is a lowpass RC filter. Combining multiple RC filters increases the circuits order. Here we have combined 2 RC filters, so we suspect the system will be 2nd order. The circuit can be seen below.

Diagram, schematic

Description automatically generated

Figure 1.1: 2nd order RC filter circuit

The resistance and capacitance values were chosen to allow frequencies under about 5 Hz to pass, while attenuating frequencies above about 10Hz. Capacitance values were kept to a minimum for safety.

R1 & R2 = 220Ω

C1 & C2 = 47 μF

Here is the Bode plot of this dynamic lowpass filter:

Chart

Description automatically generated

Figure 1.2: Bode Plot of Lowpass RC Circuit

In order to identify and model the system, we will look for the coefficients of a second order differential equation. By analyzing the systems response to known inputs, we can construct a model using the least squares technique.

We are using the sine sweep or linear chirp function as an input to excite all the dynamics in the system. The chip function was implemented in Arduino code, to produce a voltage signal, with a peak-to-peak amplitude of 5V, start frequency of 0.1 Hz, and stop frequency of 10 Hz. This range corresponds to the lower frequencies of interest in the pass band of the system.



Figure 1.3: Line 33 Arduino chirp code

A circuit board with wires

Description automatically generated with low confidence

Figure 1.4: Picture of our 2nd order lowpass RC filter circuit and the Arduino

A picture containing chart

Description automatically generated

Figure 1.5: Input and Output plots on Excel Data Streamer

Chart, line chart, histogram

Description automatically generated

Figure 1.6: Input and Output plots on Excel Data Streamer – points removed

The above plots how the systems response to an input sine sweep over ~4 seconds. Notice the phase shift over time.

# 2.0 Results and Discussion

## 2.1 Step 2

Implementing the least squares technique in MATLAB involves importing the above data as a Matrix M. Below are a few of the equations involved.

General difference equation for 2nd order system:

y(k+2)+a1y(k+1)+a2y(k)−b1u(k+1)−b2u(k)=e(k+2)yk+2+a1yk+1+a2yk−b1uk+1−b2uk=e(k+2)

Remove future dependency by substituting (k-2) in for k:

y(k)+a1y(k−1)+a2y(k−2)−b1u(k−1)−b2u(k−2)=e(k)yk+a1yk−1+a2yk−2−b1uk−1−b2uk−2=e(k)

The parameter set or vector θ:

θ=[a1,a2,b1,b2]T𝜃=[a1,a2,b1,b2]T

ϕ(i)𝜙i

 vector includes all outputs

y(i)yi

 and inputs

u(i)ui

 recorded back to the nth sample back in time:

ϕ(i)=[−y(i−1),−y(i−2),…,−y(i−n),u(i−1),…,u(i−1n)]T𝜙i=−yi−1,−yi−2,…,−yi−n,ui−1,…,u(i−1n)T

y(n)yn

 is our output at each point in time n:

y(n)=ϕT(n)θ+e(n;θ)yn=𝜙Tn𝜃+e(n;𝜃)

θLS=(ϕTϕ)−1ϕTY𝜃LS=(𝜙T𝜙)−1𝜙TY

Solving for the parameter vector in MATLAB produced our 4 parameters:

a1=−1.284     a2=0.154     b1=−0.237     b2=0.126a1=−1.284     a2=0.154     b1=−0.237     b2=0.126

Now we can represent the dynamic system as a transfer function and a difference equation.

The transfer function:

G(z)=Y(z)U(z)=b1z+b2z2+a1z+a2=−0.237z+0.126z2+1.284z+0.154Gz=Y(z)U(z)=b1z+b2z2+a1z+a2=−0.237z+0.126z2+1.284z+0.154

The general and specific difference equations for our 2nd order system:

y(k)=−a1y(k−1)−a2y(k−2)+b1u(k−1)+b2u(k−2)yk=−a1yk−1−a2yk−2+b1uk−1+b2uk−2

y(k)=−1.284y(k−1)−0.154y(k−2)−0.237u(k−1)+0.126u(k−2)yk=−1.284yk−1−0.154yk−2−0.237uk−1+0.126uk−2

We now have the output y(k) in terms of the previous two inputs and outputs. Including 2 measured outputs as the initial values, and the original input function, we can test our new model’s response:

Chart

Description automatically generated with low confidence

Figure 2.1: Least Squares Output vs Measured Output and Input

Here we can see the original input and output data “System Input” and “System Output”. Our new model’s response is also show as “Simulated Output”, based on the same input. Visually the model appears to be pretty accurate.

Checking the amount of error in the system by graphing the residuals:

Chart

Description automatically generated

Figure 2.2: 2nd Order Residuals

The residuals look like random error, which indicates that the system order is correct.

Next, we try lowering the system to 1st order, finding the parameters, and plotting the residuals:

Chart

Description automatically generated

Figure 2.3: 1st Order Residuals

The residual or error plot has a pattern that resembles the sine sweep input, indicating there are dynamics that are still at play in the system, not captured by the 1st order system.

Chart

Description automatically generated

Figure 2.4: 3rd Order Residuals

Plotting the residuals from a third order model look fairly random. Sense both the 3rd and 2nd order systems residuals look similar, we will go with the 2nd order system model.

## 2.2 Step 3

To find the frequency response of our system, we first performed the Fast Fourier Transform (FFT) on our input. The result is shown in the plot below.

Chart, histogram

Description automatically generated

Figure 2.5: Measured Input Frequency Response

In a similar fashion, we performed the FFT on our output and the result is portrayed in the figure below.

Chart, histogram

Description automatically generated

Figure 2.6: Measured Output Frequency Response

Ensuring that our input and output signals were sampled at the same rate and for the same number of times, we divided the FFT of the measured output by the FFT of the input (measured frequency response) and plotted that against the FFT of the modeled output divided by the FFT of the input (modeled frequency response). The comparison can be seen below.

Chart, histogram

Description automatically generated Figure 2.7: Measured vs Modeled System Frequency Response

It is important to note that the measured and modeled frequency responses line up well within the range of frequencies excited by the input (0.1Hz - 10Hz), as this is the frequency range we are interested in. Between 0Hz and 10Hz, the two do not match perfectly, but the trends are the same; every time there is a peak in the measured response, there is a peak in the modeled response. However, beyond 10Hz, the responses just look like noise. This is because we do not excite the dynamics of the system at the higher frequencies, so the model has no way of knowing how the system would behave at those frequencies. These spikes in frequency magnitude beyond 10Hz could be a result of the square windowing during data measurements. The system model is explaining the sudden drop in response by assuming there are higher frequency dynamics in the system, which are shown here. These dynamics are not accurate and can be dismissed as windowing noise.

## 2.3 Step 4

Using the System Identification Toolbox in MATLAB, we modeled different order systems by adjusting the number of zeros and poles in the transfer function. We compare 1st, 2nd and 3rd order models of our system and look at the time response and their residual plots.

Below you can see 4 different system outputs compared to the actual output data. The system fit increases dramatically between first and second order systems yet stays about the same between second and third order systems. This leads us to believe the 2nd order system is the best to represent our system because an increase in order doesn’t increase the fit. We also noticed a large increase in fit between a 2nd order 2 pole, 1 zero system and a 2nd order 2 pole, 2 zero system. Our least squares ID used only 1 zero, yet the fit may be increased by using a 2 zero model, according to the system ID toolbox.

Graphical user interface

Description automatically generated with medium confidence Figure 2.8: 1st order, 2nd order & 3rd order systems – MATLAB

Reviewing the residual plots for the 4 system models, we can see the blue 1st order system seems to be oscillating with the system output, leading us to believe the 1st order doesn’t represent all the system dynamics. The red plot, showing the 2nd order 2 pole 2 zero model seems to minimize the residuals and hold at zero, which shows us it best models the actual system response.

Chart, histogram

Description automatically generated Figure 2.9: System ID Residuals Plot

For a weighted least squares analysis, we changed R1 from 220Ω to ~0Ω by shorting the resistor while the system was receiving a sine sweep input.

Diagram, schematic

Description automatically generated

Figure 2.10: Circuit Diagram

Here you can see the input in blue and the output response in orange. Note the output response change when the resistance was changed.

Chart

Description automatically generated with medium confidence

Figure 2.11: Plot showing response change in system dynamics

Looking at just the output, a clear change in system dynamics is visible slightly after 2.5 seconds.

Graphical user interface

Description automatically generated with low confidence

Figure 2.12: Weighted Least Squares Simulated Response

Implementing a weighted least squares analysis technique, we analyzed the transfer functions pole parameters A1 and A2 as we scanned through the system output values. The MATLAB code conducts a for loop, to look at the output values from time 0 to N, as N increases by 1. The values for the A1 and A2 parameters are plotted below. A high value of gamma = 0.99 weights the later measurements slightly more than the previous measurement. This gamma value gave a good indication of when the systems dynamics changed.

Below you can see the systems dynamics changed slightly after 2.5 seconds, as the transfer functions parameters adjusted to a new value.

Chart, line chart

Description automatically generated

Figure 2.13: Transfer Function pole parameter values changing over time