CONTROL THEORY PROJECT WORK REPORT

ACTIVITY'S INTRODUCTION

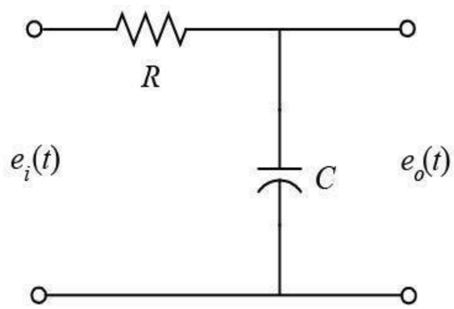
This activity aims to show how to simulate a basic electrical system. In particular, a black box method based on recorded data and a first principles approach based on the fundamental physics of the circuit will be used. Both the black box approach and the accuracy of the derived models are demonstrated through the use of the related experiment. Additionally, this activity gives a concrete illustration of the typical class of first-order systems.

TIME RESPONSE:

Components:

- Arduino board.
- Breadboard.
- Resistor
- Capacitor
- Connecting wires

Circuit Diagram



DESCRIPTION

The RC circuit's input will be created and its output will be measured using an Arduino board. The circuit's layout and design will be grounded by the breadboard, and connections between circuit locations will be made by jumper wires. The resistor will initially add resistance to the circuit. The capacitor in the circuit constantly charges and discharges as current is applied to the circuit. By efficiently generating a graph from the capacitor's voltage output, we will be able to determine several characteristics of our response.

SYSTEM MODELLING:

An Arduino board will be used to create the input for the RC circuit and measure its output. The breadboard will serve as the circuit's ground, and jumper wires will be used to connect circuit locations to one another. The resistor will first increase the circuit's resistance. As current is applied to the circuit, the capacitor in the circuit is constantly charged and discharged. We may identify numerous aspects of our response by effectively creating a graph from the capacitor's voltage output.

$$e_i - iR - \frac{1}{C}\int i\,dt = 0$$

Converting the following from variables in time domain to Laplace, we get:

$$E_i(s) - I(s)R - \frac{1}{Cs}I(s) = 0$$

$$I(s) = \frac{E_{i(s)}}{R + \frac{1}{Cs}}$$

$$I(s) = \frac{E_o(s)}{\frac{1}{Cs}}$$

Expressing this in terms of output over input we get:

$$\frac{E_{i(s)}}{R + \frac{1}{Cs}} = \frac{E_o(s)}{\frac{1}{Cs}}$$

$$G(s) = \frac{E_o(s)}{E_i(s)} = \frac{\frac{1}{Cs}}{R + \frac{1}{Cs}}$$

In the end, we arrive at the transfer function:

$$G(s) = \frac{1}{RCs + 1}$$

Considering our transfer function shows us that our system is a first order system, we can manipulate the transfer function so that it has the standard form shown below:

$$\frac{K}{\tau s + 1}$$

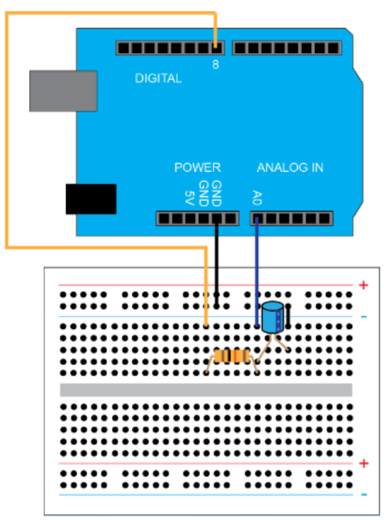
We may determine our time constant by multiplying the values of the resistor and capacitor because they have an impact on the circuit's response time. Because the circuit's time response is slow enough for the Arduino/Simulink combo to sample the circuit at a fast enough rate to provide a fair image of the circuit's output, we picked a resistance value of 10k Ohms and a capacitance of 100 micro-Farad. This gives us a sample time of one second.

We may determine our time constant by multiplying the resistor and capacitor values as their values affect how quickly the circuit responds. The circuit's time response is slow enough for the Arduino/Simulink combo to sample the circuit at a fast enough rate to produce an accurate representation of the circuit's output, therefore we chose a resistance value of 10k Ohms and a capacitance of 100 micro-Farad. As a result, we have a sampling time of one second.

In this experiment, we'll record the output voltage of the RC circuit for a change in input voltage. Based on the ensuing time response of the output voltage, we will fit a model to the data. We'll compare the model we eventually find to the one we already know, which we got via our transfer function.

COMPONENT SETUP

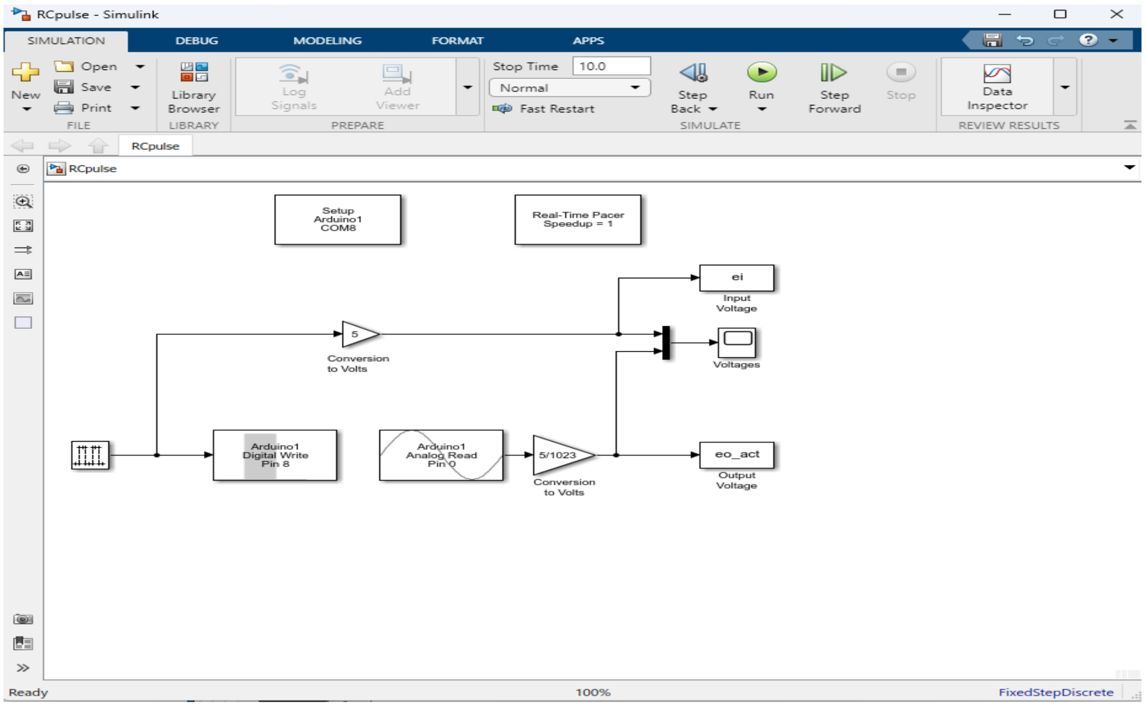
The Arduino board was attached to the breadboard on which our circuit was constructed. The board's input was then digital pin 8, and the circuit's negative side was linked to ground. The analogue pin (A0) was also linked to the output. As a result, the Arduino board is able to apply the input voltage to the circuit (through a Digital Output) and receive the input command via Simulink. Additionally, the board takes information from the output voltage of the circuit (through an analogue input) and transmits it to Simulink.



Simulink was used to read data from the board and plot it in real time. Simulink was connected to an Arduino. We had to include the Arduino IO library and a few more packages to enable serial communication between the Arduino board and the Simulink.

After working on the hardware setup, software configuration, and creating a strong connection between the board and Simulink, we plotted our graph. After it has been generated, we run the Simulink model to get the input and output voltage information. Because the plot would barely last five seconds, the capacitor discharging graph was skipped.

Our code was executed, and we got the graph shown below:



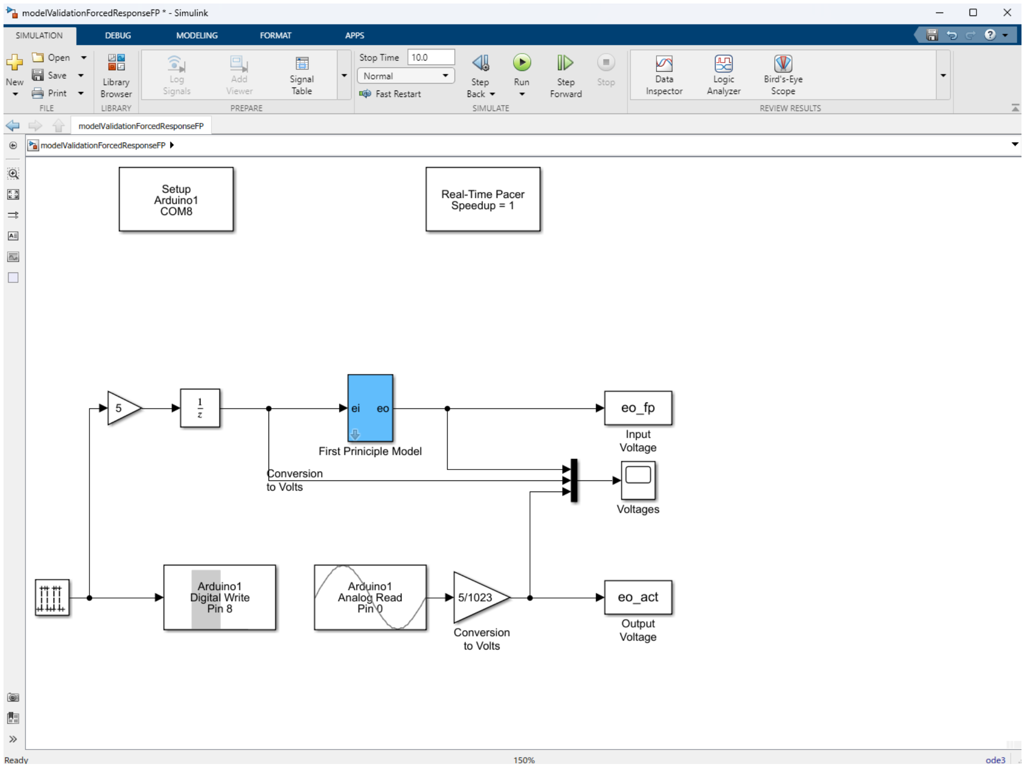
IDENTIFICATION OF THE PARAMETER

After plotting the graph, we saw that it has the appearance of a first order response. We can now write an equation in first order standard form as a result. Given as: Using the recorded response data, we can determine the values of K and our time constant (). The steady-state magnitude of the response specifically shows that K is the DC gain. Please take note that we used a Voltmeter to confirm that the output voltage produced by the Digital Output was very near to 5 Volts. Remembering that the time constant () denotes the length of time it takes the system's response to reach roughly 63 percent of its overall change, the following equation can be used to determine it: where 63.2 percent of 5 is roughly 3.16. Since the step input occurs at approximately 0.10 seconds and the output reaches 3.16volts at approximately 1.16seconds, the time constant can be identified as the time taken to 3.16 volts minus the time the step input occurs. That is 1.16-0.10 = 1.06 seconds.

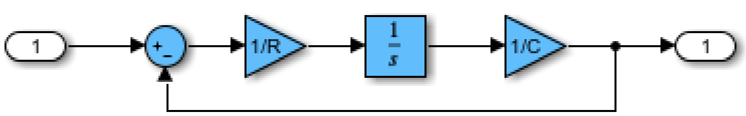
Writing the equation now as:

$$G(s) = \frac{1}{RCs + 1} = \frac{K}{1.06s + 1}$$

We next contrasted the two circuit models we came up with, the first principles model and the black box model (experimental modelling, sometimes known as black box modeling, is a method for creating models based on process data). For its highest order derivative, we created a Simulink equation using the equation from the model created before using the first principles method. Then, in order to directly compare the simulated output voltage to the actual recorded output voltage, we added the blocks to the already-existing Simulink model. Additionally, we added delay to make the simulated input voltage's timing more consistent with the input voltage produced by the Arduino board. Graph seen below.

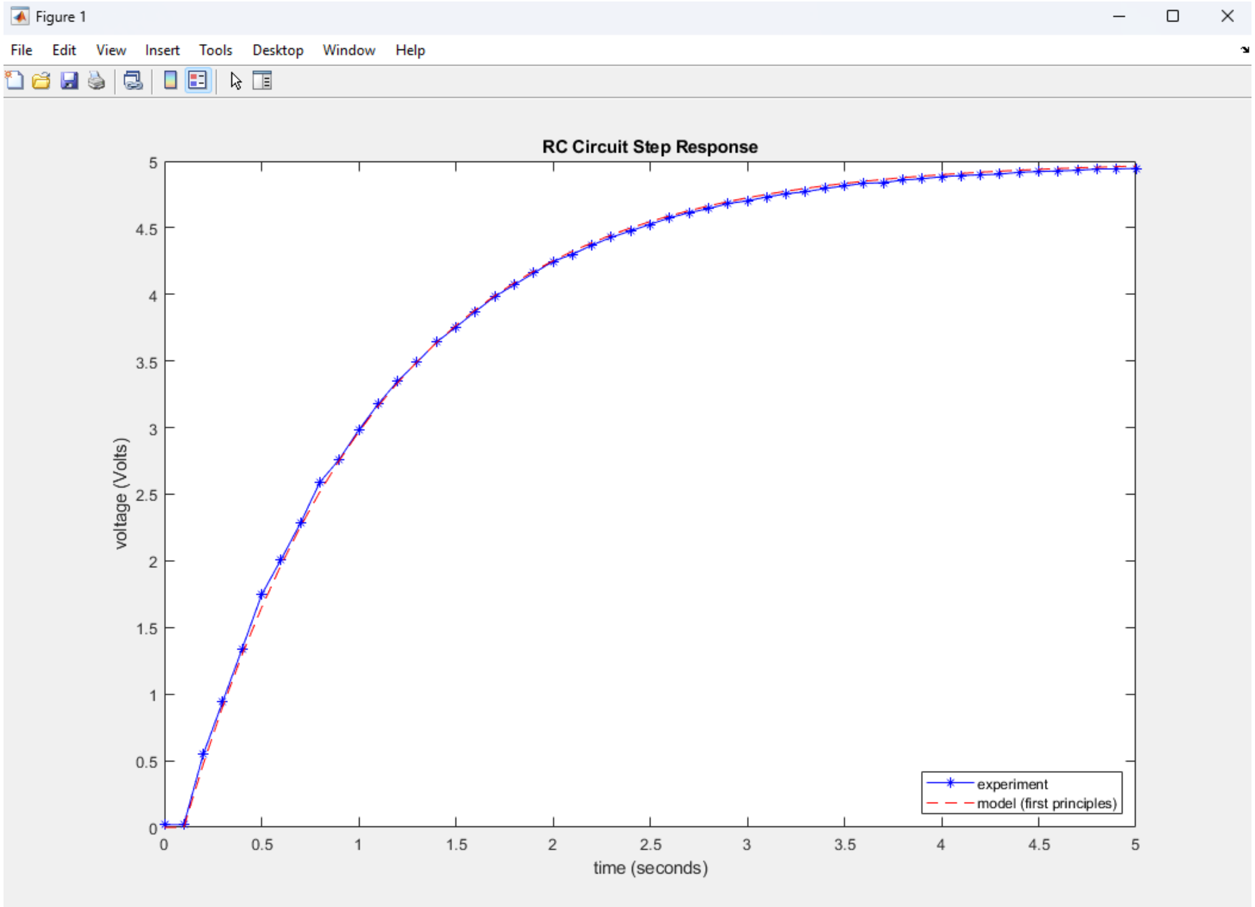


The Diagram below shows the break down of the first principle model.



$$e_i - iR - \frac{1}{C}\int i\,dt = 0$$

After running this model, we may contrast the output voltages produced by simulation (eo_fp) and actual experimentation (eo_act). Recall that the capacitor must be fully discharged for the first five seconds of the run before the Pulse Generator block steps to output 1; this is done to prevent damage to the capacitor.

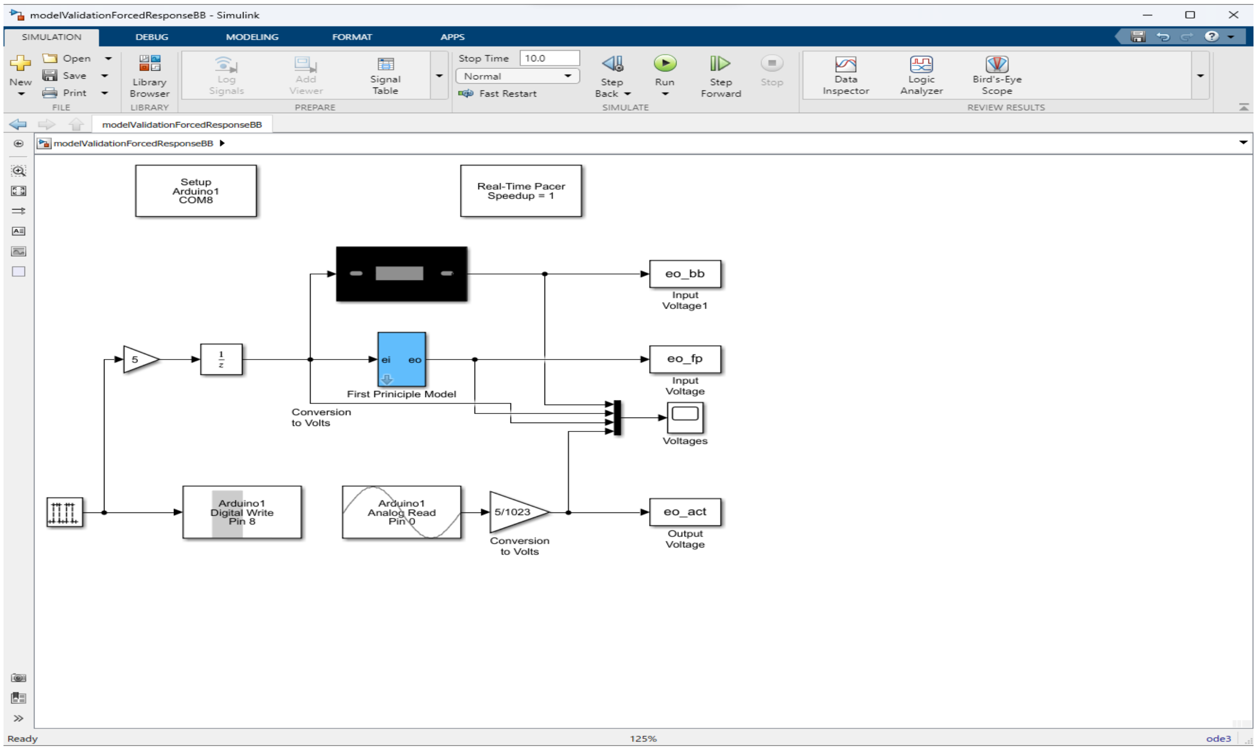


Here, the above figure shows great agreement between the output voltages predicted by our first principles model and the physical circuit. It can sometimes differ due to components values not been exact.

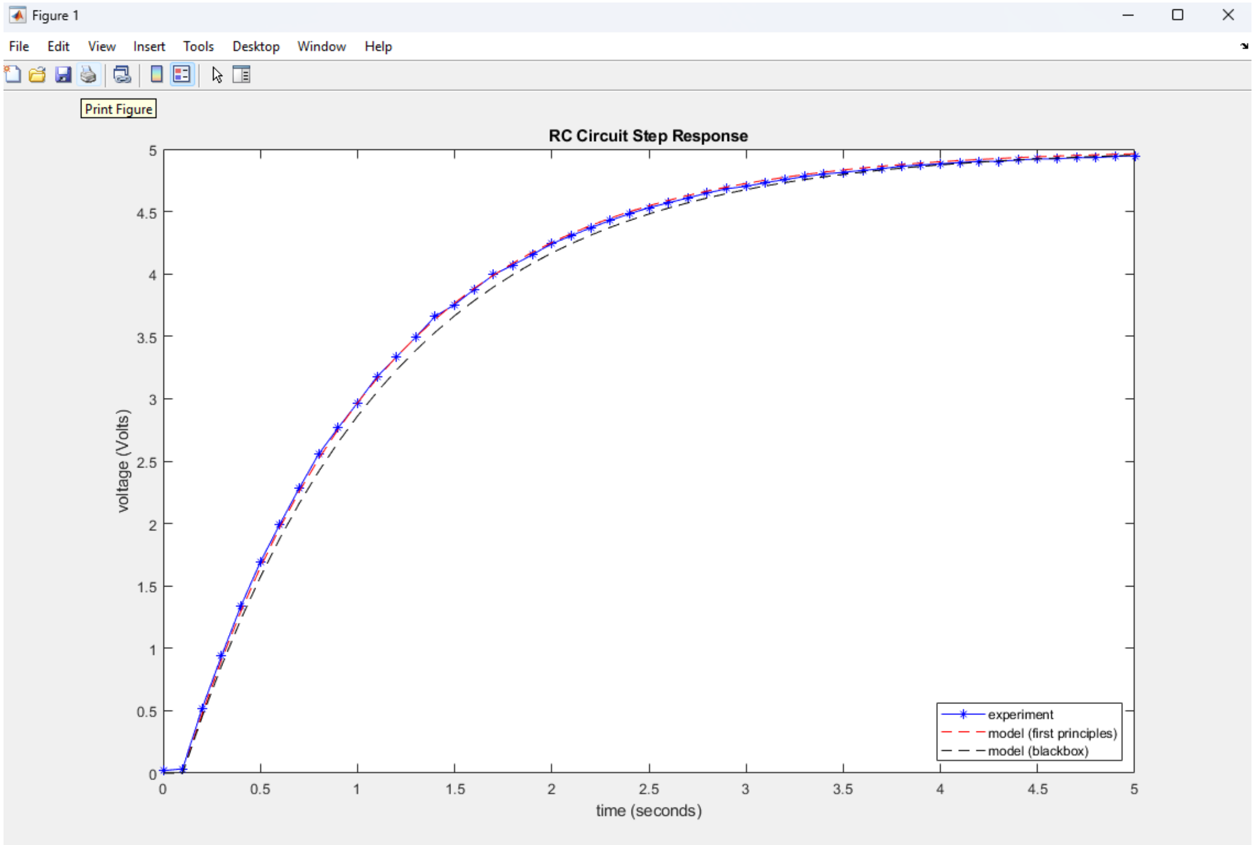
Recall that in the **System identification experiment** section we estimated the system parameters as $K = 1$ and $\tau = 1.06$ leading to the following black box model.

$$G(s) = \frac{E_o(s)}{E_i(s)} = \frac{1}{1.06s + 1}$$

We can generate the black box model in Simulink using the Transfer Function block. Double-clicking on the Transfer Function block we can set **Numerator coefficients** equal to the variable "K" and the **Denominator coefficients** equal to "[tau 1]" where *tau* is also a variable, the time constant.



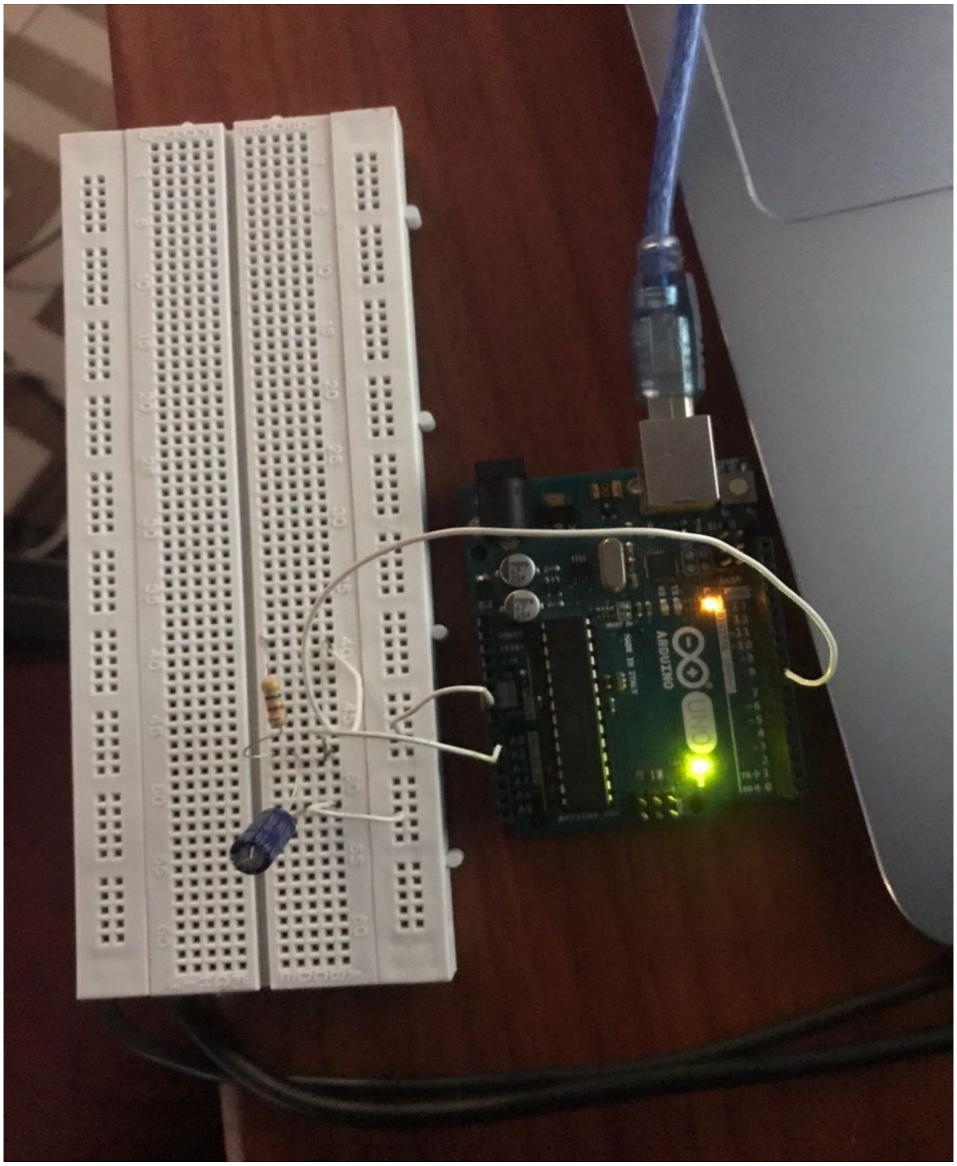
OBSERVATION



Frequency-Response Identification of a Resistor–Capacitor (RC) Circuit

Equipment Used

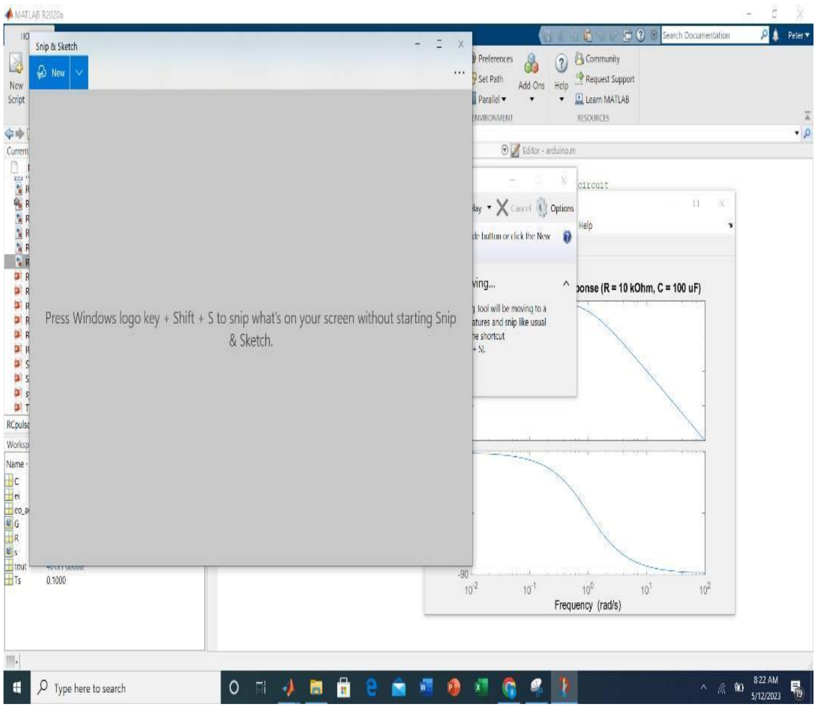
- Arduino board Breadboard
- Electronic components (resistor and capacitor)
- Ohmmeter, Capacitance meter (optional)
- Jumper wires



Purpose

This activity's goal is to better comprehend the same circuit's frequency response. For the RC circuit, we will specifically experimentally build the magnitude plot section of the Bode plot. We will sweep through a variety of frequencies in this activity, however square wave inputs rather than sine wave inputs will be used. This means that the magnitude response we produce won't quite fit the concept of frequency response. Given our knowledge of the circuit's step response from Activity 1a, we will use square wave inputs in order to develop intuition about the meaning of the frequency response.

Executing commands at the MATLAB command line will generate the theoretical Bode plot for our RC circuit (with $R = 10 k\Omega$, $C = 100 \mu F$).



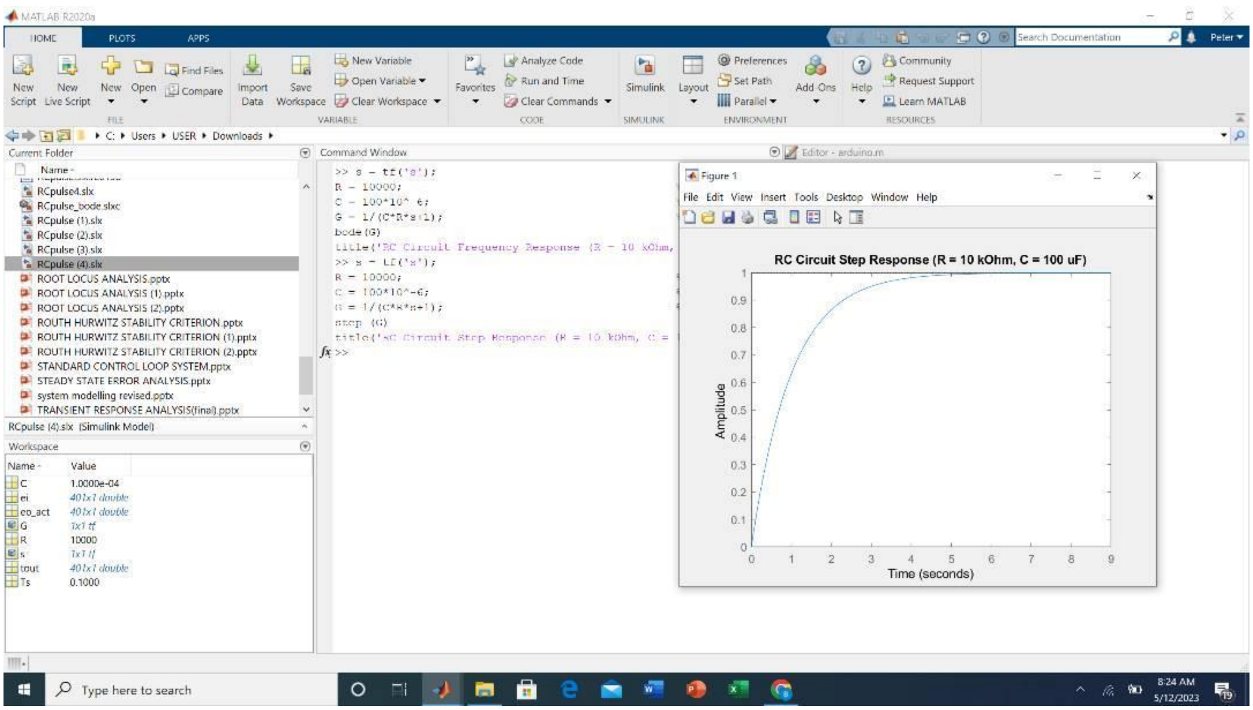
Frequency Response

When we consider a system's frequency response, we are specifically interested in how the amplitude and phase of the steady-state output compare to the sinusoidal input. One way to represent this amplitude (magnitude) data and this phase data is as a Bode plot. A Bode plot consists of two graphs, one being the magnitude of the response (the ratio of the output amplitude to the input amplitude, Y/R) versus frequency, and the other being the phase of the response versus frequency.

This circuit's break frequency is governed by the position of its pole, hence it is equal to $1/tau = 1/RC$. The break frequency for this circuit is somewhere around 1 rad/sec, and $RC = 1$. We can predict how the circuit will react to a square wave input with different frequencies by remembering the shape of the RC circuit's step response. In the following section, we'll use an experiment based on hardware to confirm our intuition. The main goal of this exercise is to have a better understanding of the significance of a system's frequency response.

The theoretical step response plot for our RC circuit (with $R = 10 kOmega$, $C = 100 mu F$) can be created by running the commands below at the MATLAB command line.

```
s = tf('s');

R = 10000;                          % resistance of resistor in RC circuit

C = 100*10^-6;                      % capacitance of capacitor in RC circuit

G = 1/(C*R*s+1);                    % RC circuit transfer function

step (G)

title('RC Circuit Step Response (R = 10 kOhm, C = 100 uF)')
```



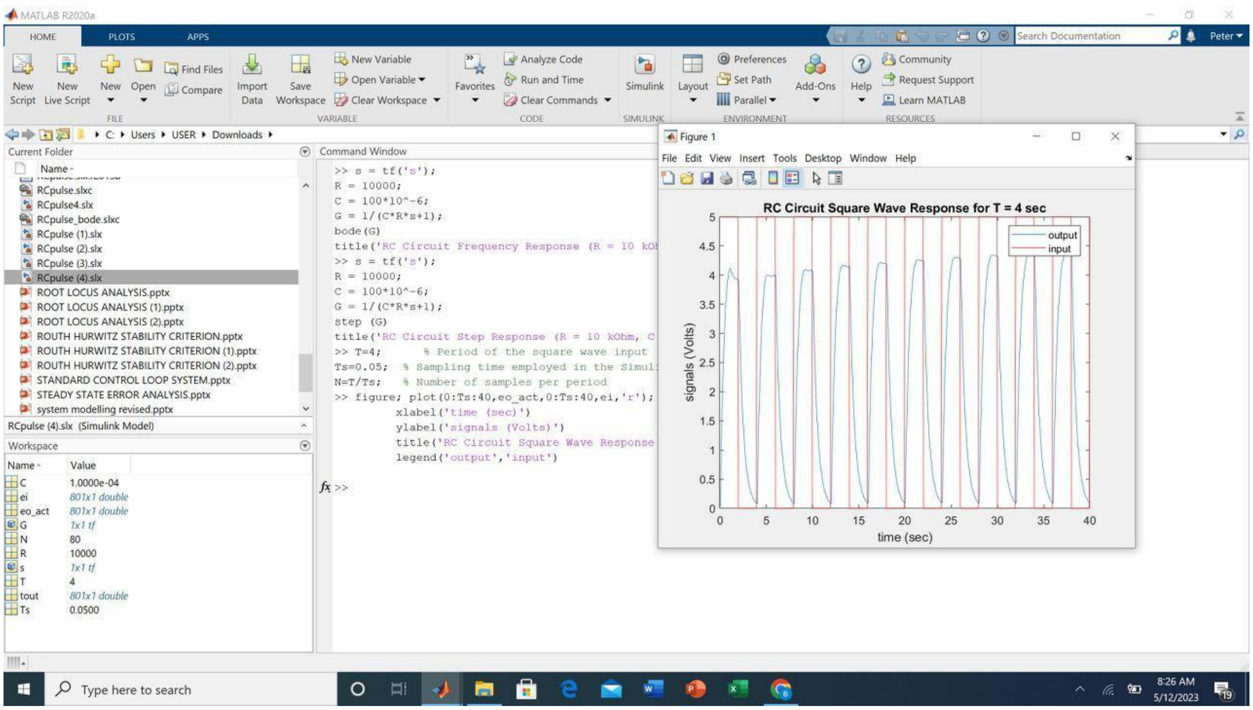
Data Gathering

For demonstration purposes, we will look at the circuit's response to a square wave input of period $T = 4$ seconds. This corresponds to a frequency of approximately 1.571 rad/sec, which is slightly faster than the circuit's 1 rad/sec break frequency. To get started, enter the following commands in the command window:

```
T=4;      % Period of the square wave input

Ts=0.05;  % Sampling time employed in the Simulink model

N=T/Ts;   % Number of samples per period
```

Entering the following code at the MATLAB command line will generate a plot like the one shown below which includes the square wave input and the circuit's output response.

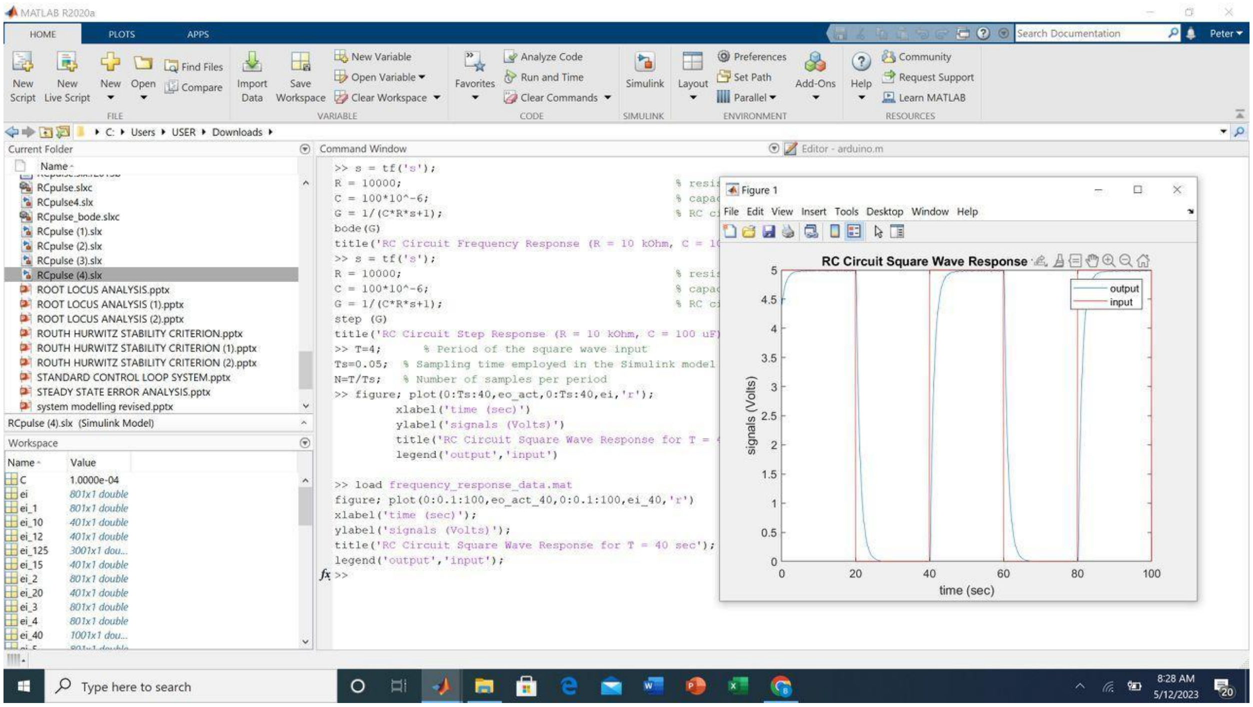
```
figure; plot(0:Ts:40,eo_act,0:Ts:40,ei,'r');
xlabel('time (sec)')
ylabel('signals (Volts)')
title('RC Circuit Square Wave Response for T = 4 sec')
legend('output','input')
```



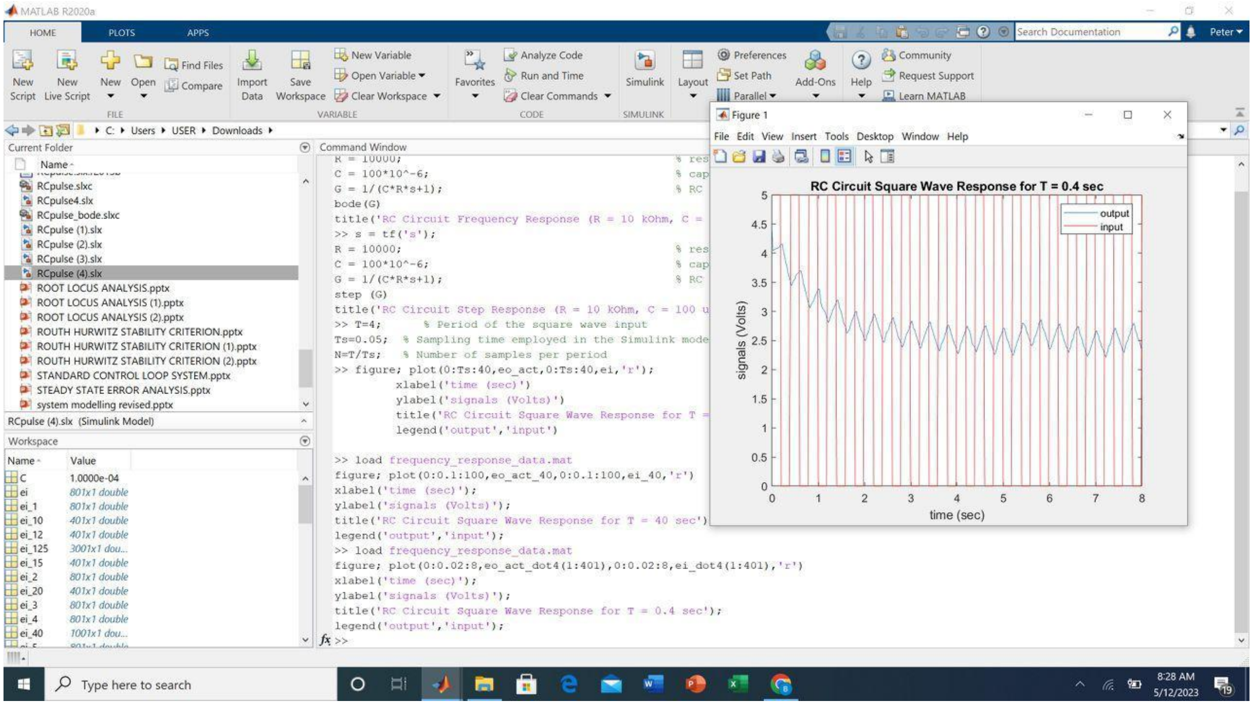
Entering the following code will allow us to examine the system's response for a slower frequency, specifically, for $T = 40$ seconds which corresponds to a frequency of approximately 0.157 rad/sec.

```
load frequency_response_data.mat
figure; plot(0:0.1:100,eo_act_40,0:0.1:100,ei_40,'r')
```

```
xlabel('time (sec)');

ylabel('signals (Volts)');

title('RC Circuit Square Wave Response for T = 40 sec');

legend('output','input');
```



The output response has plenty of time to achieve steady state before the input switches because the square wave input's period is so much greater than the RC circuit's time constant. As a result, the magnitude response of the circuit at this frequency is roughly 1 (0 dB). The output was somewhat attenuated when the period was equal to 4 seconds because the circuit did not have enough time to reach steady-state. The circuit will have even less time to react and the output will be muted more if we further reduce the period of the input (raise the frequency). The circuit's response to an input square wave with a period of $T = 0.4$ seconds and a frequency of around 15.71 will be displayed by executing the following commands.


MATLAB R2020a
HOME
PLOTS
APPS
Search Documentation
Peter
New Script
New Live Script
New
Open
Find Files
Compare
Import Data
Save Workspace
New Variable
Open Variable
Clear Workspace
Favorites
Analyze Code
Run and Time
Clear Commands
Simulink
Layout
Preferences
Set Path
Parallel
Add-Ons
Help
Community
Request Support
Learn MATLAB
FILE
VARIABLE
CODE
SIMULINK
C: ▸ Users ▸ USER ▸ Downloads ▸
Current Folder
Name
RCpulse.slxc
RCpulse4.slx
RCpulse_bode.slxc
RCpulse (1).slx
RCpulse (2).slx
RCpulse (3).slx
RCpulse (4).slx
ROOT LOCUS ANALYSIS.pptx
ROOT LOCUS ANALYSIS (1).pptx
ROOT LOCUS ANALYSIS (2).pptx
ROUTH HURWITZ STABILITY CRITERION.pptx
ROUTH HURWITZ STABILITY CRITERION (1).pptx
ROUTH HURWITZ STABILITY CRITERION (2).pptx
STANDARD CONTROL LOOP SYSTEM.pptx
STEADY STATE ERROR ANALYSIS.pptx
system modelling revised.pptx
RCpulse (4).slx (Simulink Model)
Workspace
Name Value
C 1.0000e-04
ei 801x1 double
ei_1 801x1 double
ei_10 401x1 double
ei_12 401x1 double
ei_125 3001x1 dou...
ei_15 401x1 double
ei_2 801x1 double
ei_20 401x1 double
ei_3 801x1 double
ei_4 801x1 double
ei_40 1001x1 dou...
Command Window
C = 100*10^-6;
G = 1/(C*R*s+1);
bode(G)
title('RC Circuit Frequency Response (R = 10 kOhm, C =
>> s = tf('s');
R = 10000;
C = 100*10^-6;
G = 1/(C*R*s+1);
step (G)
title('RC Circuit Step Response (R = 10 kOhm, C = 100 u
>> T=4; % Period of the square wave input
Ts=0.05; % Sampling time employed in the Simulink mode
N=T/Ts; % Number of samples per period
>> figure; plot(0:Ts:40,eo_act,0:Ts:40,ei,'r');
xlabel('time (sec)')
ylabel('signals (Volts)')
title('RC Circuit Square Wave Response for T =
legend('output','input')
>> load frequency_response_data.mat
figure; plot(0:0.1:100,eo_act_40,0:0.1:100,ei_40,'r')
xlabel('time (sec)');
ylabel('signals (Volts)');
title('RC Circuit Square Wave Response for T = 40 sec')
legend('output','input');
>> load frequency_response_data.mat
figure; plot(0:0.02:8,eo_act_dot4(1:401),0:0.02:8,ei_dot4(1:401),'r')
xlabel('time (sec)');
ylabel('signals (Volts)');
title('RC Circuit Square Wave Response for T = 0.4 sec');
legend('output','input');
fx >>
Figure 1
File Edit View Insert Tools Desktop Window Help
RC Circuit Square Wave Response for T = 0.4 sec
output
input
signals (Volts)
time (sec)
Type here to search
8:28 AM
5/12/2023