ECE486 Lab 1: Analog Simulation

Scope and Objective

1. Explore 2nd order systems through simulation of analog computer
   - Use Simulink to simulate operation amplifiers in an analog computer
   - Log data and import into Matlab workspace for analysis
2. Familiarizing with the hardware and software for data acquisition and analysis
   - Use Simulink and Arduino for Time-Response Analysis
   - Circuit prototyping with NI Labview and Elvis III board prototyping

Introduction

Analog computers are machines that computes solution of problem modeled based on changing physical properties including that of electrical, mechanical or hydraulic. In our context, analog computers can be constructed based on operational amplifiers and various circuit components. Taking the example of solving 2nd order differential equations based on operational amplifiers with negative feedback, in this lab, we will look at the 2nd order differential equation of the form:

$$\ddot{x} + 2\zeta\omega_n\dot{x} + {\omega_n}^2 x = f(t) \tag{1}$$

From here, we look at how an analog computer could solve such differential equations using simulation examples in Simulink.

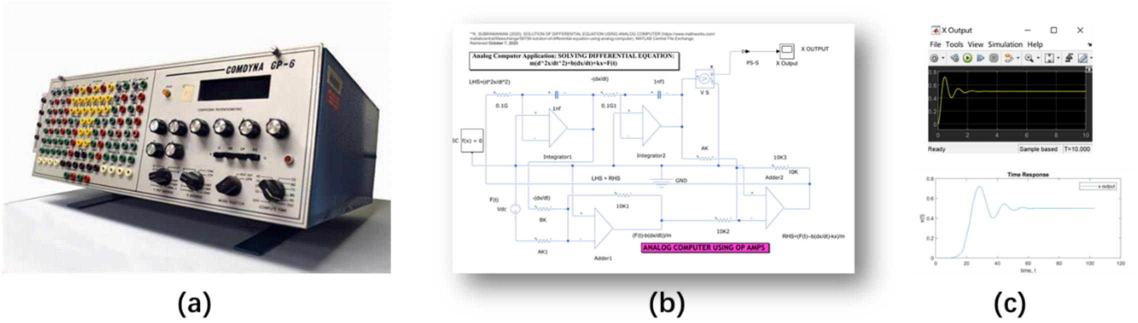

COMDYNA GP-6
Analog Computer Application: SOLVING DIFFERENTIAL EQUATION: m(d^2x/dt^2)+b(dx/dt)+kx=F(t)
Integrator1
Integrator2
Adder1
Adder2
X OUTPUT
ANALOG COMPUTER USING OP AMPS
X Output
File Tools View Simulation Help
Ready
Sample based
T=10.000
Time Response
x output
time, t
(a)
(b)
(c)

Figure 1: Analog Computer: (a) Comdyna GP-6 (b) Op Amp Simulation in Simulink, (c) Computer Analysis in Matlab Workspace

Finally, we shall also familiarize ourselves with the hardware and software for data acquisition and analysis for modeling of physical systems like those of fluidic or electrical. In particular, we will look at first order dynamic systems.

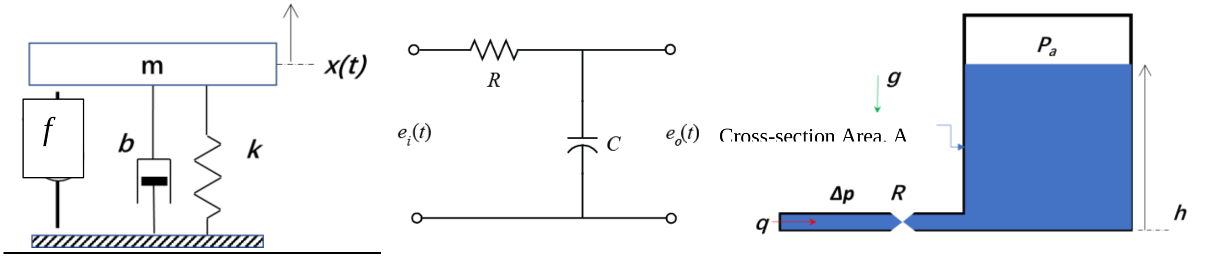


Figure 2: System dynamics that can be represented with first order ode: (a) Mass-Spring Damper System (b) RC circuit, (c) Water tank

Starting with a first-principles approach based on the underlying physics of the circuit, we move on to determine the accuracy of the resulting model through electrical circuit prototyping and demonstrate how the individual circuit components affect the response experimentally through computer analysis.

Prelab Exercise

a. Express the dynamics of the system of Figure 2(a) in the form of Equation 1 given *m*=2, *b*=0.7, *k*=1 and *f*=0.5 with zero-initial conditions.

b. Draw a block diagram to represent the system described by Equation 1.

*** In the Lab, you will be modeling this dynamic system using the 2nd order ODE and solve the model using simulated analog computer*

c. Show that for the systems in Figure 2 b and c can be expressed as follows.

$$e_i - iR - \frac{1}{c}\int i\,dt = 0 \qquad (2)$$

$$A\dot{h} - \frac{g}{R}h = 0 \qquad (3)$$

*** In the Lab, you will be analyzing the first order systems via I) data acquisition, II) circuit prototyping and III) computer analysis*

Prelab Readings (Optional)

Analog Computing Technique by Robert Paz: Uploaded on Course Website

Simulink Basic Tutorial (for Control Application):
http://ctms.engin.umich.edu/CTMS/index.php?aux=Basics_Simulink

Simulink ArduinoIO Package (a good option in connecting Simulink for hardware projects):
http://ctms.engin.umich.edu/CTMS/index.php?aux=Activities_IOpack

Lab1 Exercise 1: Solving Differential Equations using Analog Computer

**model from: K. SUBRAMANIAN (2020). SOLUTION OF DIFFERENTIAL EQUATION USING ANALOG COMPUTER (https://www.mathworks.com/matlabcentral/fileexchange/56756-solution-of-differential-equation-using-analog-computer), MATLAB Central File Exchange. Retrieved October 7, 2020.

1. Launch the Simulink model *ECE486Lab01_Differential_eqn.slx*.

You should see the following model and be able to draw relationship with the dynamic system represented by a 2nd order ODE in prelab

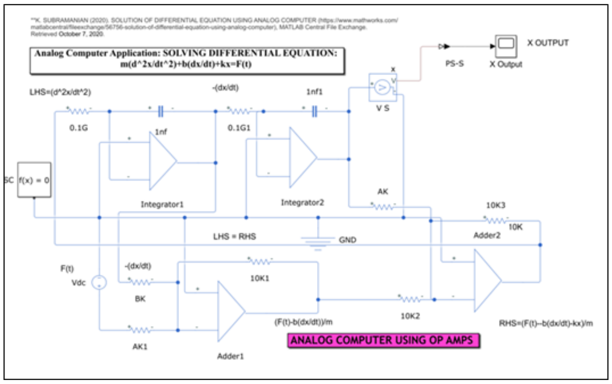

**K. SUBRAMANIAN (2020). SOLUTION OF DIFFERENTIAL EQUATION USING ANALOG COMPUTER (https://www.mathworks.com/matlabcentral/fileexchange/56756-solution-of-differential-equation-using-analog-computer), MATLAB Central File Exchange. Retrieved October 7, 2020.
Analog Computer Application: SOLVING DIFFERENTIAL EQUATION:
m(d^2x/dt^2)+b(dx/dt)+kx=F(t)
X OUTPUT
PS-S
X Output
x
V S
LHS=(d^2x/dt^2)
-(dx/dt)
1nf1
0.1G
1nf
0.1G1
SC
f(x) = 0
Integrator1
Integrator2
AK
10K3
10K
Adder2
LHS = RHS
GND
F(t)
Vdc
-(dx/dt)
BK
10K1
AK1
Adder1
(F(t)-b(dx/dt))/m
10K2
RHS=(F(t)--b(dx/dt)-kx)/m
ANALOG COMPUTER USING OP AMPS

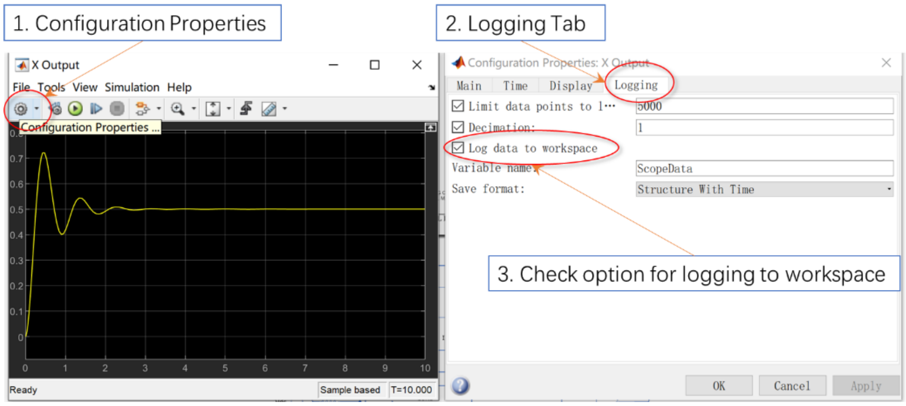
Figure 3: System dynamics that can be represented with first order ode: (a) Mass-Spring Damper System (b) RC circuit, (c) Water tank

2. Study the Simulink model and input the appropriate parameters to model the dynamics of the system in Prelab Exercise (a)

3. Run the model and display the output of *x* over time

4. Display also the signal of the velocity by connecting the necessary blocks

5. Log the data on MATLAB workspace *x(t)* and *x'(t)*. Plot both functions in MATLAB


1. Configuration Properties

2. Logging Tab

3. Check option for logging to workspace

☑ Limit data points to l… 5000

☑ Decimation: 1

☑ Log data to workspace

Variable name: ScopeData

Save format: Structure With Time

Figure 4: Activate logging to workspace option

6. Change the appropriate parameter such that the drag coefficient *b*=1

7. Play around with different inputs and parameters to look at their responses. In your report, discuss the changes in response based on graphical observation from the simulation experiment.

Lab1 Exercise 2: System identification experiment

In this experiment, we will investigate dynamic systems modeled with first order ODEs, via I) Data Acquisition, II) Circuit Prototyping and III) Computer Analysis in this experiment.

This is a concise version of a series of control experiments from: http://ctms.engin.umich.edu/CTMS/index.php?aux=Activities_RCcircuitA. Interested learner may go through them optionally at their own pace.

Hardware setup

A RC circuit can be implemented on a breadboard and connected to the Arduino board as shown. One thing to note is that if you employ an electrolytic capacitor, its orientation matters. Specifically, if your capacitor has legs of different lengths and one leg is marked by a negative sign, then you have an electrolytic capacitor. Orient an electrolytic capacitor so that the leg marked by the negative sign connects to the lower potential part of the circuit (ground in this case). The Arduino board is employed to receive the input command from Simulink and to apply the input voltage to the circuit (via a Digital Output). The board also acquires the output voltage data from the circuit (via an Analog Input) and communicates the data to Simulink.

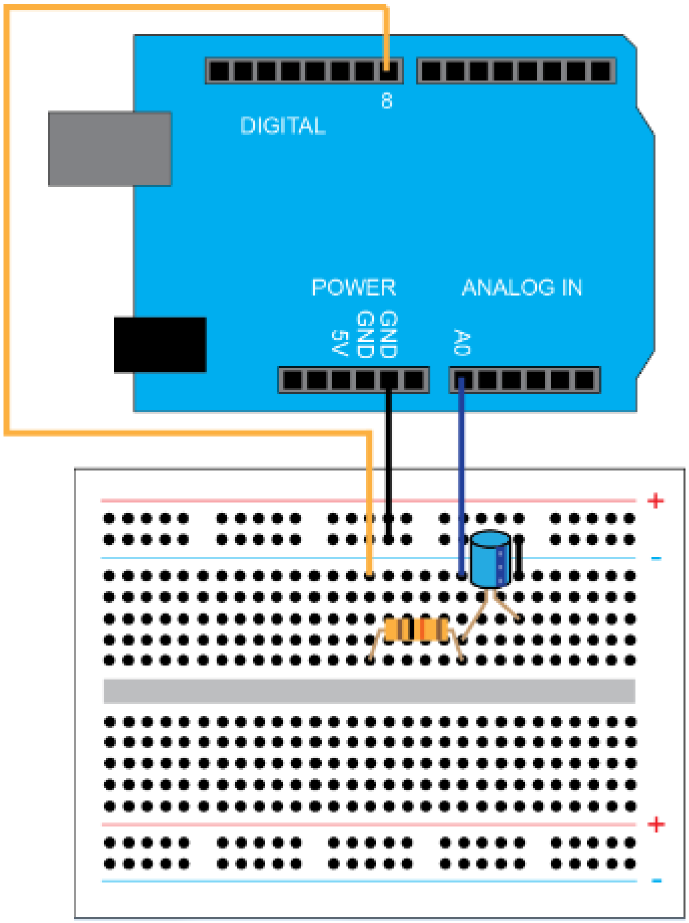


Figure 5: Circuit Prototyping

In this experiment, the values of the resistor and capacitor are chosen such that the circuit's time response is slow enough that the Arduino/Simulink setup can sample the circuit at a fast-enough rate to give a clear picture of the circuit's output. If the sampling rate is at least 10 times faster than the circuit's time constant, we will be able to get a clear picture of the response. In this experiment, we chose $R = 10\ k\Omega$ and $C = 100\ \mu F$ to produce a circuit with time constant $\tau = RC = 1$ second. This is a low power application so most resistors and capacitors available will work fine, just make sure that the capacitor is rated for at least 5 Volts.

Software setup

In this experiment, we will employ Simulink to read the data from the board and to plot the data in real time. In particular, we will employ the IO package from the MathWorks. You may need to change the port to which the Arduino board is connected.

As shown below, the input voltage command is generated by a Pulse Generator block. This block is chosen because of its generality, though for this experiment we could have used a Step block. The Pulse Generator block generates values of 0 or 1 which are then fed to an Arduino Digital Write block. Since we are using channel 8 for the digital output, we double-click on the Arduino Digital Write block to set the Pin to 8 from the drop-down menu. An input of 0 to the Digital Write block causes an output of 0 Volts to be generated at the corresponding pin, while an input of 1 to the Digital Write block generates an output of 5 Volts. This scaling is captured by the Gain block that is included prior to the Scope block. In this model the Pulse Generator block is set to output 0 for the first 5 seconds of the run in order to discharge the capacitor completely (in case it had built up charge) before stepping to 1 for the last 5 seconds of the run.

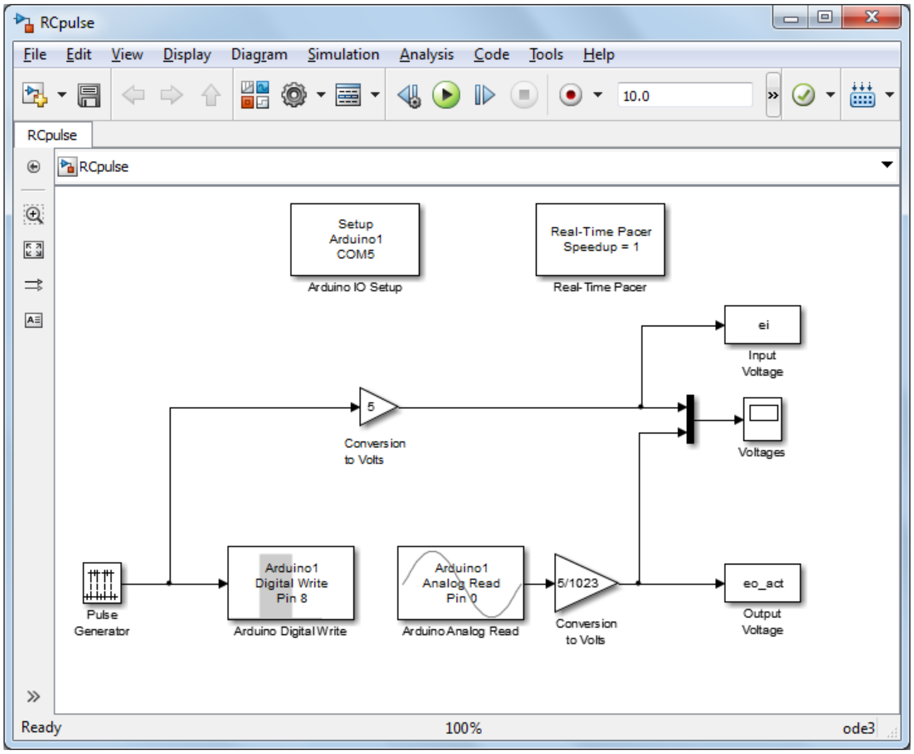


Figure 6: Simulink Model

The Arduino Analog Read block reads the output voltage data via the Analog Input A0 on the board. Double-clicking on the block allows us to set the Pin to 0 from the drop-down menu. We also will set the Sample Time to "0.1". This is 10 times faster than the circuit's time constant and hence is sufficiently fast. The other blocks in the model can also be set to have a Sample Time of "0.1" (or left as "-1"). In the downloadable model, the sample time is set to the variable Ts which needs to be defined in the MATLAB workspace by typing Ts = 0.1 before the model can be run. The Gain block on the Analog Input is included to convert the data into units of Volts (by multiplying the data by 5/1023). This conversion can be understood by recognizing that the Arduino Board employs a 10-bit analog-to-digital converter, which means (for the default) that an Analog Input channel reads a voltage between 0 and 5 V and slices that range into $2^{10} = 1024$ pieces. Therefore, 0 corresponds to 0 V and 1023 corresponds to 5 V.

The given Simulink model then plots the commanded input voltage and recorded output voltage on a scope and also writes the output data, as an array, to the MATLAB workspace for further analysis. The Arduino Digital Write block, the Arduino Analog Read block, the Arduino IO Setup block, and the Real-Time Pacer block are all part of the IO package. The remaining blocks are part of the standard Simulink library, specifically, they can be found under the Math, Sources, and Sinks libraries.

Once the Simulink model has been created, it can then be run to collect the input voltage and output voltage data. Executing the following code at the MATLAB command line will generate the graph shown below. Note that we plot only the last five seconds of the run, thereby excluding any discharging of the capacitor that may have taken place.

```
plot(0:0.1:5,ei(51:101),'r--');
hold on
plot(0:0.1:5,eo_act(51:101),'b*-');
xlabel('time (seconds)')
ylabel('voltage (Volts)')
title('RC Circuit Step Response')
legend('input','experimental output','Location','SouthEast')
axis([0 5 0 5.1])
```

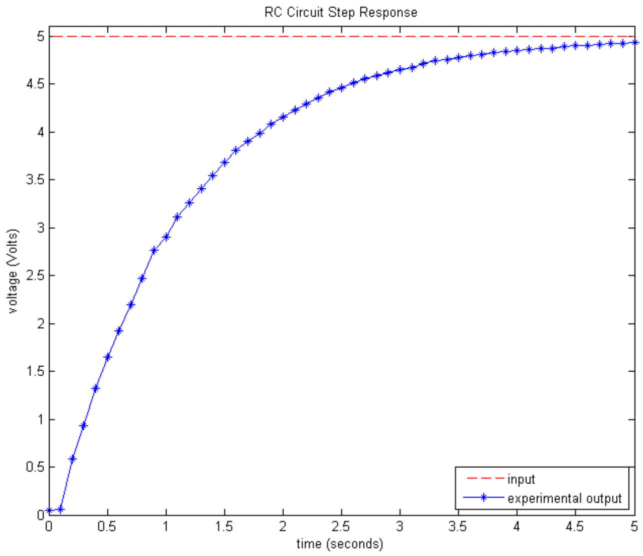


Figure 7: RC Circuit Step Response

Parameter identification

Based on the on the above figure, we can fit a model to the recorded data. Recognizing that the observed data has the shape of a first-order step response, we will assume the following standard first-order model for the circuit (happens to match our first-principles model).

$$G(s) = \frac{E_o(s)}{E_i(s)} = \frac{K}{\tau s + 1} \tag{8}$$

We can then identify the system parameters K and τ from the recorded response data. Specifically, the steady-state value of the response indicates that the DC gain is $K = e_{o,ss}/e_{i,ss} \approx 1$. Note, we verified with a Voltmeter that the output voltage generated via the Digital Output was very close to 5 Volts. Recalling that by definition the time constant τ represents the time it takes the system's response to reach $1 - e^{-1} \approx 63.2\%$ of its total change, τ can be calculated from the following where 63.2 percent of 5 is approximately 3.16.

In your report, show how you obtain the value of τ empirically.

Suggested guide for report:

Exercise 1

6. Change the appropriate parameter such that the drag coefficient *b*=1

→ Change resistor value of BK from 28.6 to 20 kOhm.

7. Play around with different inputs and parameters to look at their responses. In your report, discuss the changes in response based on graphical observation from the simulation experiment.

→ With a different few experimental values, Show that at steady state the response matches the input F(t), b creates damping effect and k oscillating effect.

Exercise 2:

Parameter identification

```
plot(0:0.1:5,eo_act(51:101),'b*-');
xlabel('time (seconds)')
ylabel('voltage (Volts)')
title('RC Circuit Step Response')
legend('experimental output','Location','SouthEast')
axis([0 5 0 5])
```

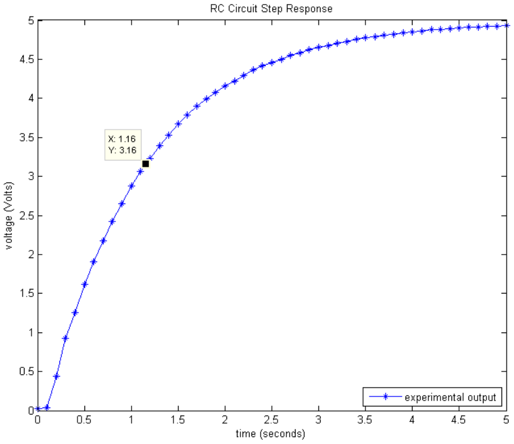


Figure 8: Parameter Identification using Step Response Experimentally

Since the step input occurs at approximately 0.10 seconds and the output reaches 3.16 Volts at approximately 1.16 seconds, the time constant can be identified as $\tau \approx 1.16 - 0.10 = 1.06$ seconds. In reality, we only have data at intervals of 0.1 seconds, but using a datatip in the MATLAB plot allows us to interpolate between the samples. Using a smaller sample period would allow us to improve the accuracy of our identification of the system parameters.

In conclusion, our resulting blackbox model is the following.

$$G(s) = \frac{E_o(s)}{E_i(s)} = \frac{1}{1.06s+1} \tag{9}$$