**Problem Description:**

This is a group project, so you will work on the code in a group. You will report which part/ percentage of the code was written by which person. Your uploads will include your code and a YouTube link to your 2-minute project video.

Your project goal is to create a Graphical User Interface (GUI) using MATLAB, Object-Oriented Programming and the Appdesigner system. The GUI should have a central area (one or more) where graphing is done and menu items, buttons and LEDs will be located as needed. The system should connect Matlab with an Arduino microcontroller with a sonar distance sensor hooked up to it. The system should take distance data and time data from the computer (see: tic/toc) from a swinging pendulum. You will clean and smooth this data and show the distance vs. time plot of your pendulum. You should also allow for continuous plotting of sensor data.

A picture containing text, electronics

Description automatically generatedA picture containing electronics, projector

Description automatically generated



Note: Power and ground are on the left (red, power) and (blue, ground)

In this example, the trigger is pin seven and, the echo is pin six.

The following features should be part of your system.

1. It should have a menu or button to initialize the Arduino and the distance sensor. Use the given example program, but make sure that the system can have the variables of echo and transmit pins, port etc., to be input as part of the GUI.
2. Create a simple pendulum from string and cardboard or a small plate. Hook the end to a appropriate height. Then swing (use a small controlled movement) and collect pendulum data with your system.
3. The data captured should be cleaned and smoothed using techniques discussed in class (median filter) or your developed techniques.
4. To help you determine good positioning, create LEDs (on the breadboard and the GUI) that indicate when good data has been captured. If data is in the range of “good” data, a green LED lights up; a red LED lights up if it is in a “bad” range. Make sure to use the appropriate resistors to ensure the LEDs don’t burn out. You can then carefully position your pendulum in a range with mostly good data collection.
5. You should have a menu or GUI features that allow the data to be captured from actual hardware or a file you have saved from the GUI and can reload. This will allow you to debug code and get it working without the hardware hooked up.
6. In the center of the GUI, create a plotting system of your design. Make sure to be able to plot the original data and the smoothed data.

**Experiments:**

* 2D Continuous plotting: Create a button that continuously plots your sensor values for a selectable amount of time.
* Collect data from pendulums of at least three different lengths.
* Using the mouse and clicking on the peaks of your graph, determine the period of the pendulum swings for the three different trials.
* Determine how the model of the pendulum compares with the [theory of pendulums](https://en.wikipedia.org/wiki/Pendulum) and compute the error of your model compared with pendulum theory. Compute the gravitational constant from your data. Creatively (within the time limit) show this in your video. Remember a shorter video will take longer to make. Make this a video that you can show during an interview.

**Project Requirements:**

A critical exercise in this project is organizing these codes in a structured manner (objected-oriented structured programming). **Your program codes must be well documented**. Reuse code whenever possible, creat mini functions that can be used between methods (do not duplicate code that can be written as functions). Indicate in the commenting what portions of the code were done by which group member. Indicate also what % was done by which member.

Create a 2-minute video of your project. Show the software running, the hardware setup, the data collection, and how your system compared with the model of the pendulum. All members of the team should narrate some portion of the video.

Make sure your code is well organized, neat, clear. Make the code and video something you could show at an interview.

**Make sure the code does not contain cut and pastes from other student code or the internet. This will result in a zero for the project for both.**

**Programming Tips:**

In order to load Arduino with the ultrasound library installed, you will have to issue a command like this: Note that ‘Com7’ should be replaced with your com and ‘D7’ ‘D6’ should be replaced with your pins on the ultrasound.

app.Arduino = arduino('COM7','Uno','Libraries','Ultrasonic');

% app.Arduino is a private variable of your class, you can

% name it anything you want.

% connect to the Ultrasound sensor the class variable

% is UltraSound. Same as above.

app.UltraSound = app.Arduino.ultrasonic('D7','D6');

app.ClearAllData();

To Read data from the Ultrasound sensor:

Array(i) =app.UltraSound.readDistance();

Here create an array private varialbe that you read data into.

Use the tic/toc features to get the time values.