Temple University

College of Engineering

Department of Electrical and Computer Engineering (ECE)

Student Report Cover Page

Course Number: ECE 3412
Lab 2: Designing a bi-directional open-loop position controller

Student Name: Devin Trejo; Robert Irwin
TUid: 914924557; 914980083
Due Date: 2/10/2015

TA Name: Michael Korostelev

Grade: / 100

I. Introduction

MatLab is a versatile toolbox that allows us to build digital control systems quickly. In this lab we will use an open loop system for the control of a stepper motor. We create the hypothetical situation of using a photo resistor to sense the ambient light in a room. If the light is too low then we tell the motor to open the blinds, and if the light is too high then we close the blinds. The stepper motor is crucial for this operation in order to precisely control the position of the blinds.

MatLab has built in functionality to use stepper motors via an Arduino board. The stepper block has three inputs so we can control the speed, direction, and step count directly. Using switching, sum, difference, memory blocks we can construct our open loop control system that controls the inputs to our stepper motor.

II. Procedure

To begin the experiment, we first had to construct the following circuit.

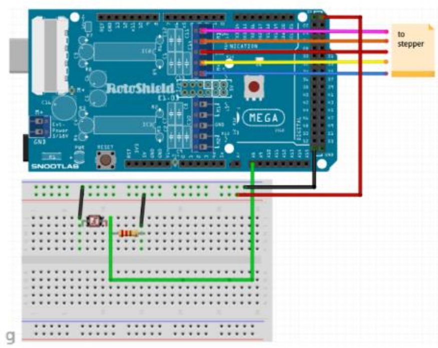

Figure 1: The Hardware Configuration for the Experiment

Once we had the circuit set up, we had to connect it to the stepper motor. The pin-out for the motor is shown below.

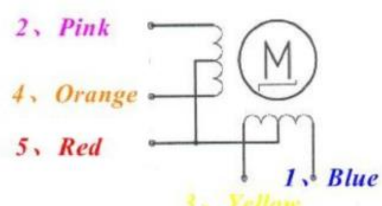

Figure 2: Pin-out for the Stepper-Motor

Now that all of the hardware is connected we can build the control system in Simulink that will gather our experimental results. The block diagram for the control system is shown below.

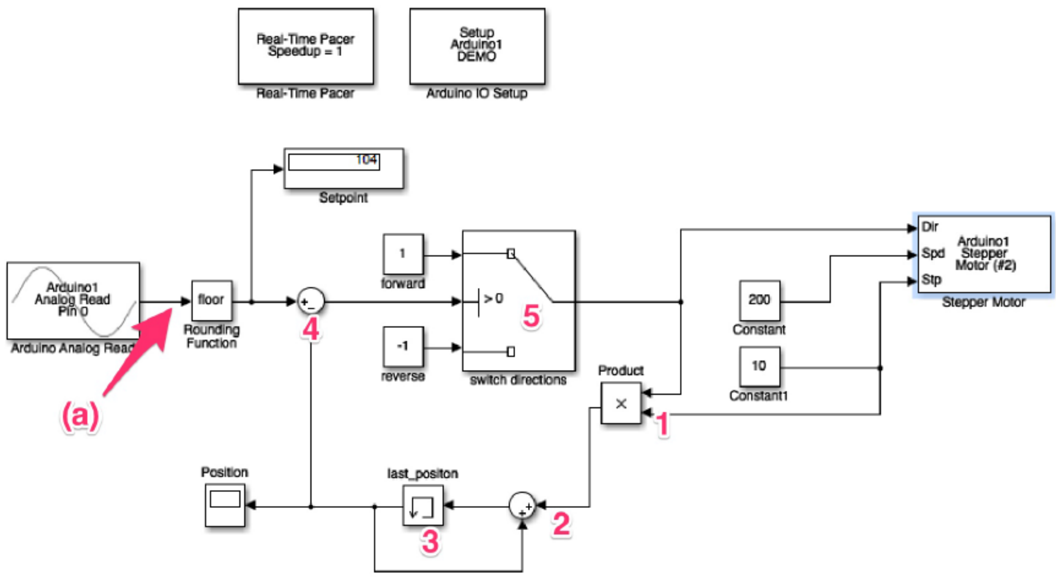

Figure 3: Block Diagram

Now we are ready to begin collecting data. The first order of business was checking the values displayed on the set point block. This block digitally displays the reading from the photo-resistor. If the range of values from darkness to full exposure to light was not wide enough, we would have to add a scaling factor so that the logic gate could register two different states. In our case, this was not an issue. Before we began the experiment however, we had to examine and research how each block in the system functions.

Next we had to run the simulation and watch how the stepper rotated when the photo cell was exposed to different amounts of light. The position from the motor was stored in the MatLab workspace, and we plotted the data.

III. Results

Below we see the position of the stepper motor plotted against a time scale.

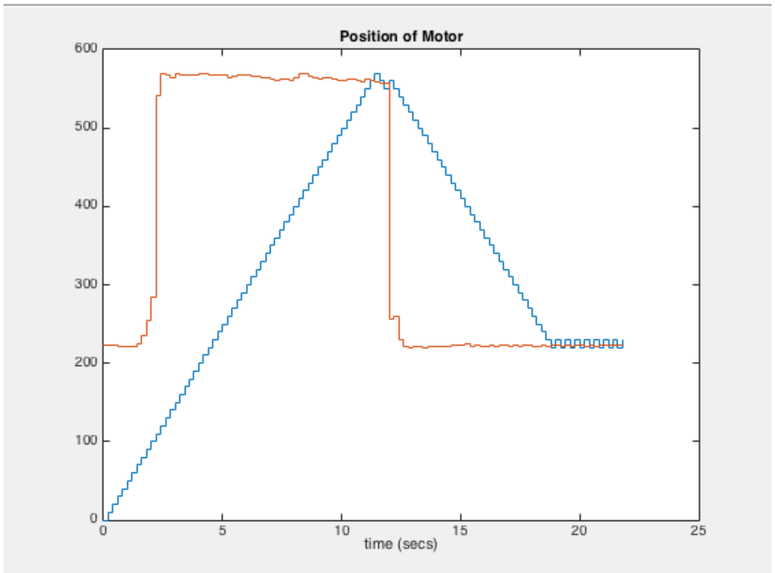

Figure 4: Position Vs. Time Plot, Step Size 10

This figure shows the value outputted by the photo-resistor (in orange), and the position of the servo (in blue). It is important to note that the step size for this plot was 10. From this output we can see that the motor takes about ten seconds to reach its final position. This final position represents the photo-cell with maximum light exposure.

We also notice that once the desired position is reached, it oscillates about the final position. We can also see this oscillation around 20 seconds, when the stepper motor achieved the position that corresponds to minimum light exposure.

The ten seconds it takes the stepper to go from one position to the other may be undesirable. To make this amount of time less significant, we can increase the speed of the stepper, or simply increase the size of each step. The figure below demonstrates the effect of a doubled step size.

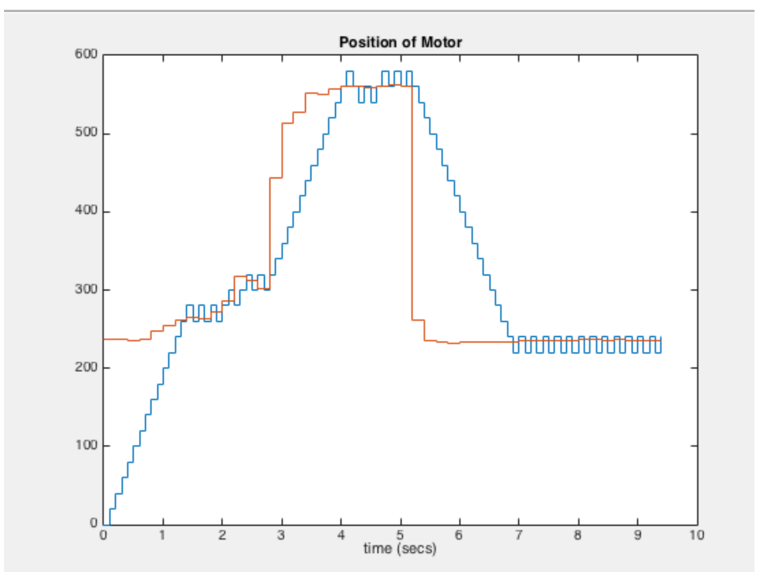

Figure 5: Position Vs. Time Plot, Step Size 20

From the plot above, we can see that doubling the step size halved the amount of time it took the stepper to go from one position to the other. With the dramatic decrease in time, we see a dramatic increase in the oscillation about the final positions of the stepper. Because of this effect, we must sacrifice speed to achieve a steadier final position.

Below shows the change in the block diagram that resulted in the increased step size. It is also important to note that we decreased the sample time of the position for this plot.

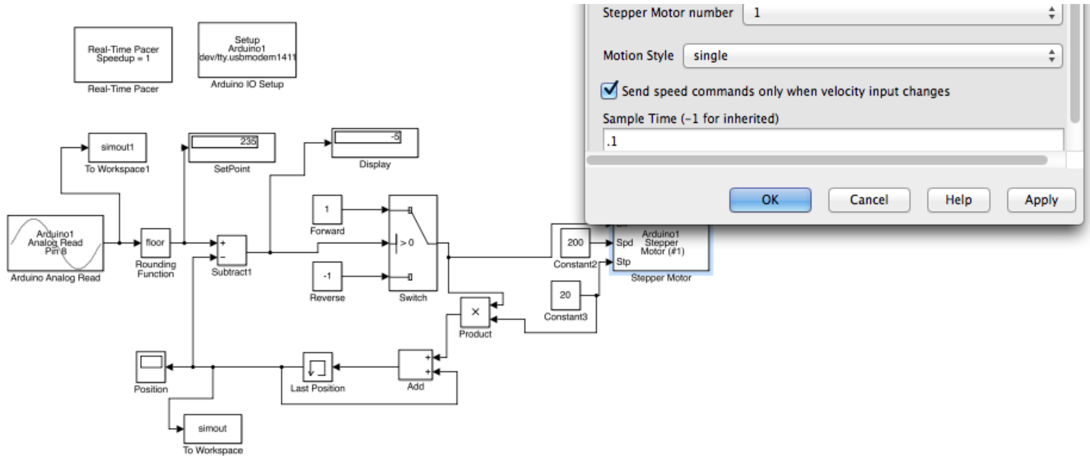

Figure 6: Shows the Altered Block Diagram that resulted in an Increased Sample Time as well as an Increased Step Size.

In the figure above, we can see that that to achieve an increased step size, we simply had to change the constant block that fed the step input of the stepper-motor block. To increase the sample size, we had to double click the stepper block, and alter the sample time.

IV. Conclusion/Discussion

Let us begin this conclusion by answer question proposed during the lab period:

1. **In the lab, demonstrate a functioning controller to the instructor.**
   Lab demonstrated
2. **How was the compensating value determined for this controller?**
   The compensating value was determined by identifying the maximum and minimum output of the photo-resistor. If these values were to close, a scaling factor had to be chosen that would widen the gap between these values. A good way to do this would be to subtract a value that made the minimum output negative, while keeping the maximum output positive, and multiplying by 100. This would yield values roughly between -100 and 100, which would provide enough of a difference to get the stepper rotating.
3. **Output the position value to the MATLAB workspace and plot it using MATLAB's plotting function.**
   See Figure 4 and Figure 5
4. **After constructing the system and circuit, examine and research the numbered blocks in Figure 2.3. Describe how each functions and what purpose it performs in the controller.**
   Each individual block constructs a loop to tell the motor what direction to move in. The first block we start with is the step count multiplied by the direction we want the motor to move in. Block 2 takes the product of the previous block and adds it with the previous position of

the motor. The previous position of the motor is held by the "last_position" block or block 3. We save the position of the block position so we later can plot the position of the block in comparison to our desired position. We take the difference (block 4) between the current stepper motor position and the desired position set by the photo resistor. The 5th block completes the loop by taking the difference value found by block 4. If block 4 is positive then our direction chooser (block 5) will tell the motor to close the blinds. If negative it tells the block to open the blinds.

5. **In this simulated scenario, the position of the motor does not effect the value of the photocell. How might this system act differently in real world implementation?**
   In a real world scenario, the position of the stepper would definitely effect the value of the photo cell. If this was a system that controlled blinds based on light exposure, the position of the servo would reduce, or increase the exposure of the photo cell. Therefore, the in the same type of plot shown previously, the output of the photo-resistor and the position of the stepper would converge toward each other, as opposed to the position of the stepper converging to the output of the photo-resistor.
6. **The motor's previous position is fed back into the system to calculate the next position, yet this is still open loop control. Why?**
   We do not have a true closed loop system since we do not know far the stepper motor actually moves. The stepper motor has three inputs and no inputs. The input are as follows: an input telling the motor what direction to move in, the speed we wish it to move, and the number of steps to take. The only input into the stepper motor that we manipulate with our system is the direction. If the photo resistor is higher than the last position then our stepper motor will work to close the blinds. Vice versa if the photo resistor is lower than the last position. An estimate of the last position is used to determine the proportion of the photo resistor value to the last position seen. Since no direct measurements are being taken on how far the stepper motor has moved, we say it is open looped.
7. **What are some advantages and disadvantages of the two types of motors we have used?**
   The other motor we looked at was a DC motor. A DC motor does not have as much precise control as a stepper motor. To get the same precision with a DC motor would require other control techniques to sense where the DC motor is. Also, stepper motors have full toqure no matter what speed it operating at, while in contrast DC motors build up torque depending on its speed. DC motors can operate at faster speeds when compared to stepper motors which make them advantageous in certain situations. In our application we wanted to control blinds to a precise level which is why we choose stepper motors.

Looking back at the lab we know now how to quickly construct a control system in MatLab. For this lab we had an open loop control which is easier, cheaper, and generally more stable ton to construct. We could combine the techniques we learned in the first lab to construct a close loop control system, which would provide us with better accuracy in our motor control. The closed loop control would help alleviate the problem we saw where our motor would oscillate back and forth between two positions. Generally, close loop control is a better implementation of a control system to ensure your system is outputting what it should be.