Mechatronic Systems Engineering
School of Engineering Science
SIMON FRASER UNIVERSITY

MSE 312

Design Project – Robotic Arm Integration

Professor: Dr. Arzanpour

Zheng Wang 301188444
Sohail Sangha 301186636
Rame Putris 301047157
Ane Tendo 301186203

Lab section: Thursday morning
Due Date: Aug 7th 2015

Introduction

The robotic arm system introduced in this project has a cantilever arm mounted to a motor shaft to create a simple cantilever arm system conducting a pick and place operation. During the operations, a metal object is attracted by an electromagnet to the robot arm and moved from position 0 to a specified angular position with a specified overshoot and minimum overall time. The integration section of this project mainly involves Simulink block designs of feedback loop, motor driver, magnet control and parameter measuring meters, and PID tuning for better performance. This document will cover the integration portion of our design.

Integration designs

1. Interfacing with Analog I/O and Encoder Module

The interface with the "Encoder" block is shown below:

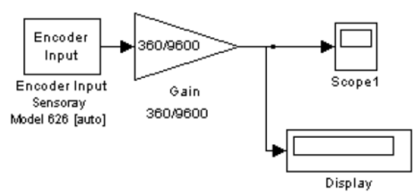

Figure 1: Encoder Module

The shown blocks can read the value of the encoder, through the gain block, readings are interpreted as the position of the motor. By having the output of the block, position and speed of the motor can also be measured and controlled.

2. Motor Driver Block

The H-bridge on electronic drive for the robotic arm project is controlled by two input lines A, and B. Controlling a pulse width modulated digital signal on one line, while keeping the other low, exposes the terminals of the motor to the power supply. Reversing the signal and low line switches the location of power supply and GND on the motor terminals thus allowing speed and direction control.

The H-Bridge is sent digital signal from a computer controlled embedded system linked to the SimuLink software. The control for the motor was based on PID control which outputs a singular analog value. The singular analog value needed to be converted into the line A, B signals mentioned above, which was achieved using the motor driver block.

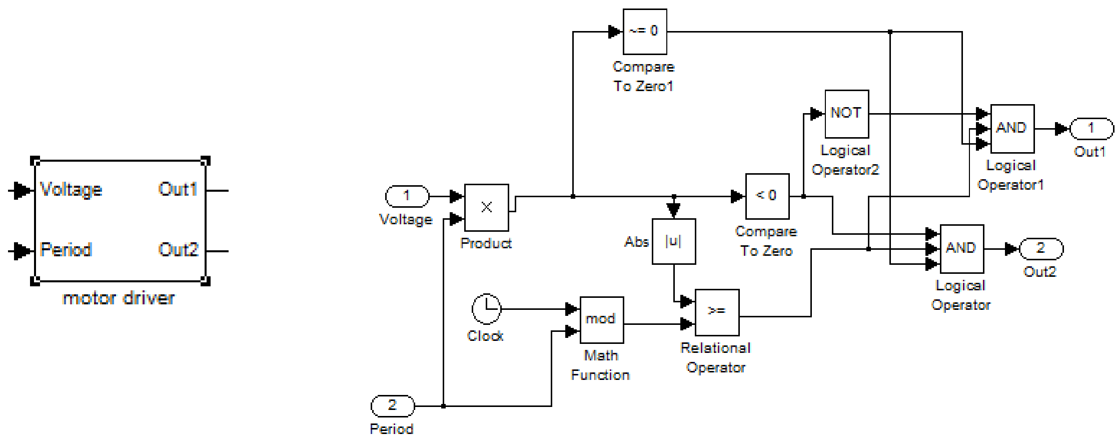

Figure 2: Motor Drive Block

The motor driver block takes a voltage and period input. The voltage input ranges between -1 to 1 and corresponds to a voltage of -V Supply to +V Supply on the motor terminals. The period input to the block is used to determine the time period of the pulse width modulated wave sent to the line A or B, while the Out1 and Out2 correspond to A and B. The block was tested using simulated signals and verified with an oscilloscope before integration.

3. PID Controller Integration

The main objective of the robotic arm project was to be able to control a single degree of freedom arm using PID control loop. Details regarding the closed loop and its requirements can be found in the simulation section of this project. This report focuses solely on the integration of the PID control into an actual system, the problems encountered and fixes designed to overcome the problems.

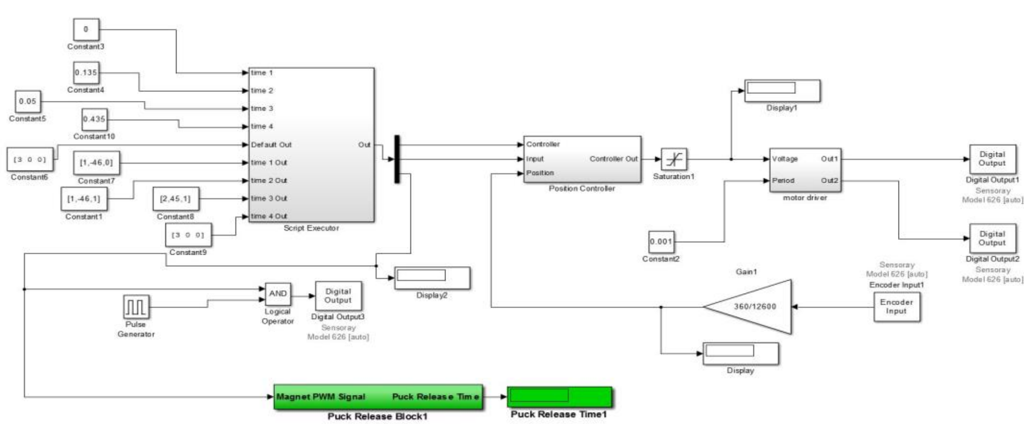

Figure 3: Close Loop Controller

The closed loop PID control is achieved by obtaining the incremental position from the encoders and converted into degrees, which is subtracted from the reference value to generate error signal and sent through the PID. The analog value produced by the PID is sent through the motor driver to produce H-Bridge signals.

Issues

- During the initial integration of the PID control the team was attempting to obtain comprehensive data using the scopes in SimuLink. The software was unable to handle the visual processing required and thus resulted in a slower running software. While the issue was unobservable at the program level, the frequency of the signal going to the H-Bridge was drastically reduced, resulting in a very noisy motor.
- The movement of the motor was very noisy, although relatively less than with the scope.
- The incremental encoder produced a small amount of noise, which resulted in a noisy PID even after settling.

Fixes

Most of the problems faced at PID integration stage were regarding the noise from the motor, although they would not be considered performance hindering, they generally adversely affect the lifetime of the motor, while obviously impacting the students through the high frequency noise. The fixes put in place to combat issues listed above were:

- All the scopes were removed from the SimuLink controller, leaving only displays for data collection. Further precautions were taken by testing blocks in simulation to ensure desired behavior.
- The simulation was run at 1 us fixed time step. Although the fixed step time was impossible to meet, this ensured that the Simulink PID controller was sampling and running at maximum frequency possible. The noise was drastically reduced by setting the period of motor driver block to 1 ms.
- A dead zone was placed on the error fed into the PID controller, this ensured that the voltage output from PID was 0 resulting in the motor completely electronically shutdown when the error was too small to operate on.

4. Overshoot and Settling Meter

As listed under the issues for the PID integration a major setback to SimuLink controller design was the unavailability of scope in order to the run the system as fast as possible. To be able to quantitatively tune the PID controller, two blocks measuring the overshoot and the settling time were designed.

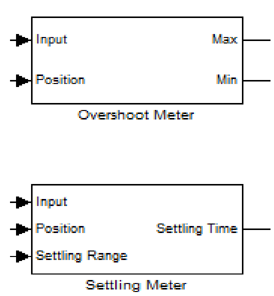

Figure 4: Measuring Meter Block

The overshoot meter takes in the reference input to the PID controller, and the actual position of the robot. The output max is useful when the reference input changes to larger number, for example the robot was settled at 0 degrees for equivalent reference input, but the input now changes to 45 degrees. The max would give a quantitative values in degrees by which the robot has moved further than the input. The minimum output is useful in the reverse condition where input changes to smaller value.

The settling meter taken the first two inputs similar to as defined for overshoot meter above. The third input settling range is the values in degrees in which the system has to remain to be considered settled.

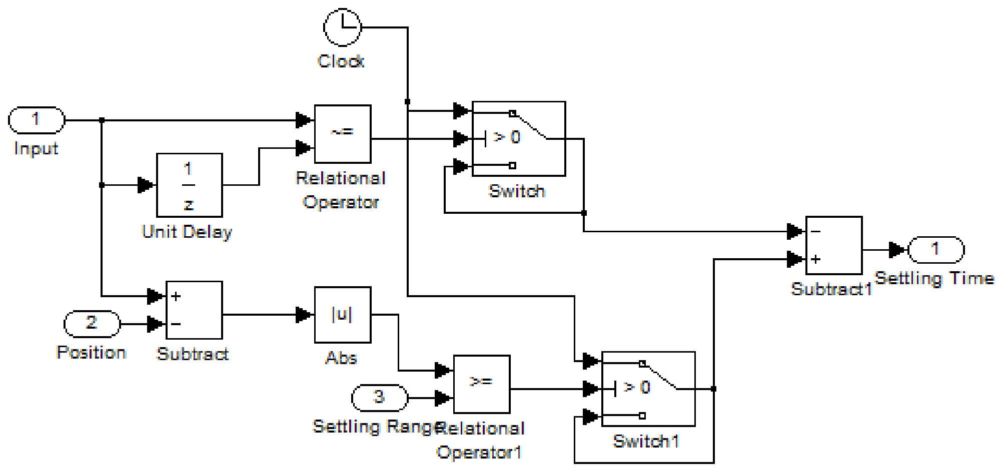

Figure 5: Settling Meter Block Design

Both the settling meter and overshoot meter are based on the idea of triggering and capturing using associative loops constructed using switches. Exploded block diagram of settling meter is shown above, for sake of conciseness of the report, overshoot meter will not be displayed.

5. PID Tuning

The tuning of PID values for achieving high pick and place operation performance involves two individual tuning parts. When the cantilever arm moves towards -45 degree position, the arm will pick up the magnet, which is extra load attached to the cantilever arm. Hence, a new set of PID tuning values is tested and adopted for smooth operation during both operation periods. During PID tuning, we first increase the proportional gain to increase the arm operation speed. While testing different proportional gain values between a certain range, we change the integral gain and derivative gain to limit the overshoot and control the settling time, which are monitored in real time by settling meter and overshoot meter mentioned above. The finely tuned sets of data for our application are shown below:

Table 1: PID, Settling Time and Overshoot Values

	PID1(toward -45 degree)	PID2(toward 45 degree)
Proportional gain (P) :	1/4	1/7.5

Integral (I)	0	1/200
Derivative (D)	1/75	1/70
settling time	0.184s	0.4263s
overshoot	0.01 degree	0.257 degree

6. Magnet Testing

When we initially apply set voltage to the magnet circuit loop, the magnet is not working as expected. After testing the applied voltages on circuit components, we found that the MOSFET was burned and we need to replace it by soldering a new circuit loop on the bus board. After replacing MOSFET and applying 12V, the magnet successfully picked up the metal in the designated position.

7. Script Executor and Position Controller Block

In order to increase the speed of operation and to ensure that the desired sequence of events occurs we created a script execution block that controls which controller is active, the position input and the electromagnet input (on/off). In short, at certain times which we found through tuning the system the system will execute the programmed actions.

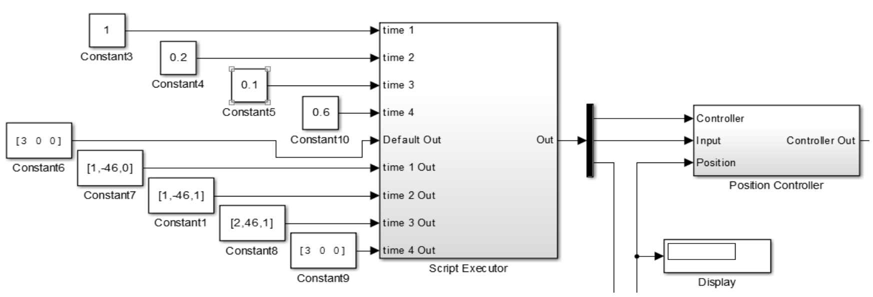

Figure 6: Script Executer and Position Controller Blocks

Conclusion & Recommendations

In conclusion, we were able to achieve a time of 0.62 seconds and have the system run smoothly without any loud motor noises. Also, we were able to achieve our time consistently with a number of trials. The controller which we have developed is stable and operates as intended. We believe that our mechanical design has also contributed to the speed of our system due to its compactness and rigidity. All in all, our group was successful at achieving the objective with a consistent competitive time and with the required overshoot criteria.