

ECEN315 - Compensating a Propeller Driven Pendulum

Joshua Benfell - 300433229

Abstract—

1 INTRODUCTION

In the previous report the open loop transfer function was derived for a propeller driven pendulum. The response of the system was stable with oscillations making it unideal for practical use. To improve the system, a compensator will be designed for it to speed up the systems response and make it less oscillatory. To do this, a variety of compensators will be modelled using the simulink package to gain a better understanding of the function and benefits of them. Finally a PID compensator will be implemented with that reduces the settling time of the system at least threefold.

2 BACKGROUND

3 SIMULINK MODEL

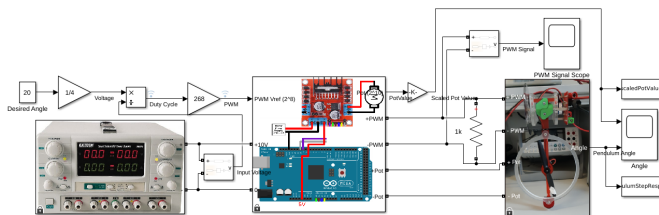


Fig. 1. Calibrated Open Loop System

In the last report, the transfer function of our system was derived, however, the derived system is not always an accurate representation of how the actual system will behave. To imitate this dissonant behavior, a slightly different system will be used in the simulink models. The system used in this report (shown in ??, where it is fully calibrated.), takes a PWM reference and then outputs the resultant angular displacement. However, it is desirable that the input units are the same as the output units, so the system needs to be calibrated.

There are 3 main calibration stages on the input sides:

- Converting the desired angular displacement into the required applied voltage.
- Converting that voltage into a duty cycle
- Converting that duty cycle into the PWM value the arduino should output to the motor

To calibrate these values first start with the angle. By applying a voltage over the pendulum motor an angular displacement can be generated. By applying different voltages, a relationship can be determined between the voltage and the angle. For this system it was found that the voltage required is 4 times less than the desired angle (??).

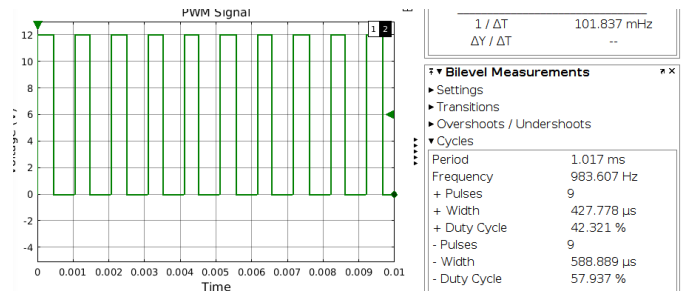


Fig. 2. Signal Generated by the Arduino

The next constant to determine is the conversion between voltage and a duty cycle. To do this, the system incorporates the arduino to control the effective voltage applied to the motor. A series of PWM values are input into the arduino and the resultant signal will be read across a $1k\Omega$ load, where the duty cycle can be measured (??). Plotting the relationship between duty cycle and PWM and voltage and duty cycle result in the constants 2.68 and 1/12 respectively (??); this second constant is 1 over the source voltage. A factor of 100 needs to be added in somewhere along this cascaded chain

as the duty cycle is shown as a percentage not a proportion.

With the above constants computed, the pendulum will now rise to the desired angle. There is one final constant to compute, and that is the constant to convert the analog potentiometer value into an angle so that the angle can be read for later application; practically the arduino is unable to read the angular displacement without a component such as a potentiometer. To calibrate this constant, the potentiometers value is plotted against the resulting angle; the constant that converts the ADC value to an angle is $1/25$ (??). Finally, the

such we are able to simulate an error ???. The relation between the error and the steady state value for the system with simulated error is that the error is 0.2

3.1 Proportional

3.2 PI

3.3 PD

3.4 PID

4 DISCUSSION

5 CONCLUSION

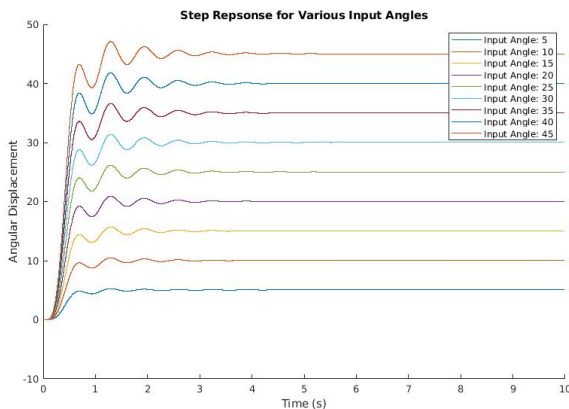


Fig. 3. System Response to various desired angles

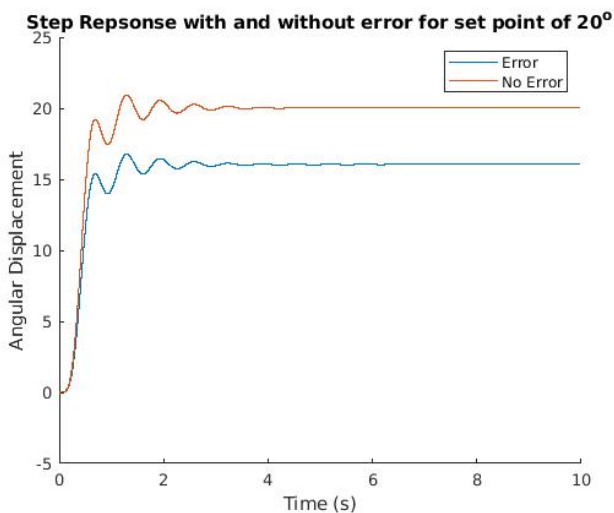


Fig. 4. System Response with and without a simulated Error

calibrated system needs to be tested to confirm that it reaches the desired input. ?? shows the output to various desired inputs, and it can be seen that the system has zero steady state error for all the inputs. However, not all systems are perfect and as

APPENDIX A

LAB 4 CALIBRATION

Effective Voltage (From PWM) vs Pendulum Angle

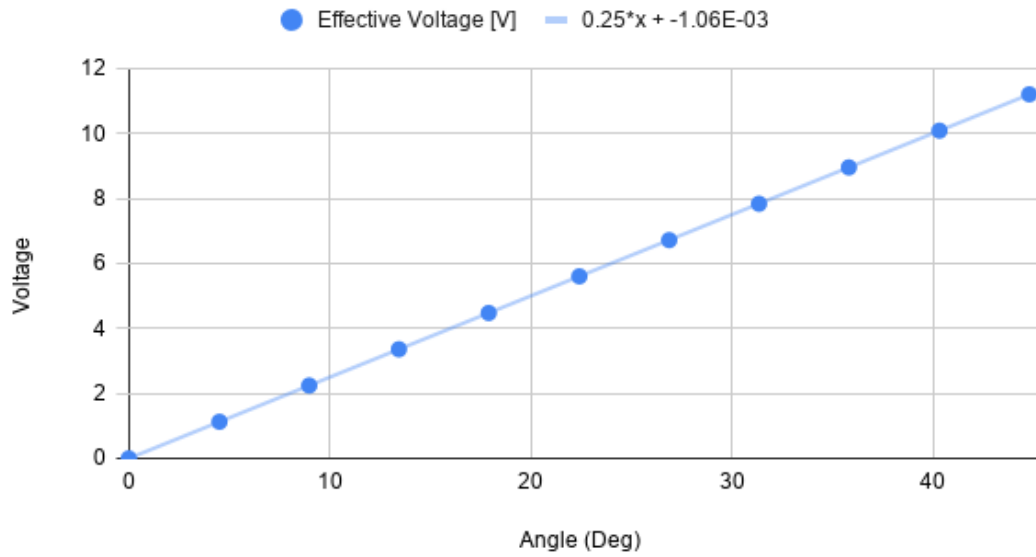


Fig. 5. Plot of Voltage applied to motor and the angular displacement produced

Duty Cycle (%) vs Voltage (V)

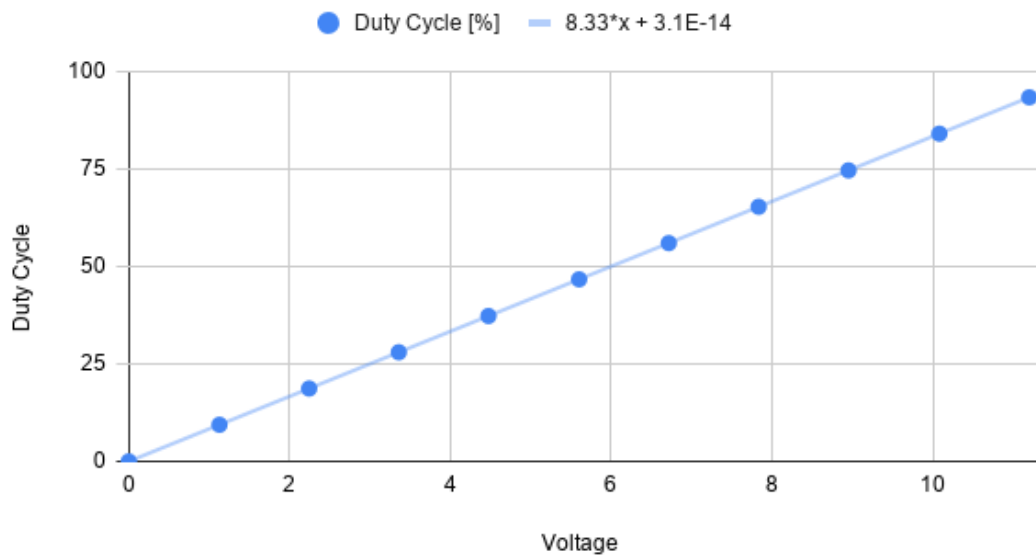


Fig. 6. Plot of effective voltage for a given duty cycle

Duty Cycle vs. PWM Size

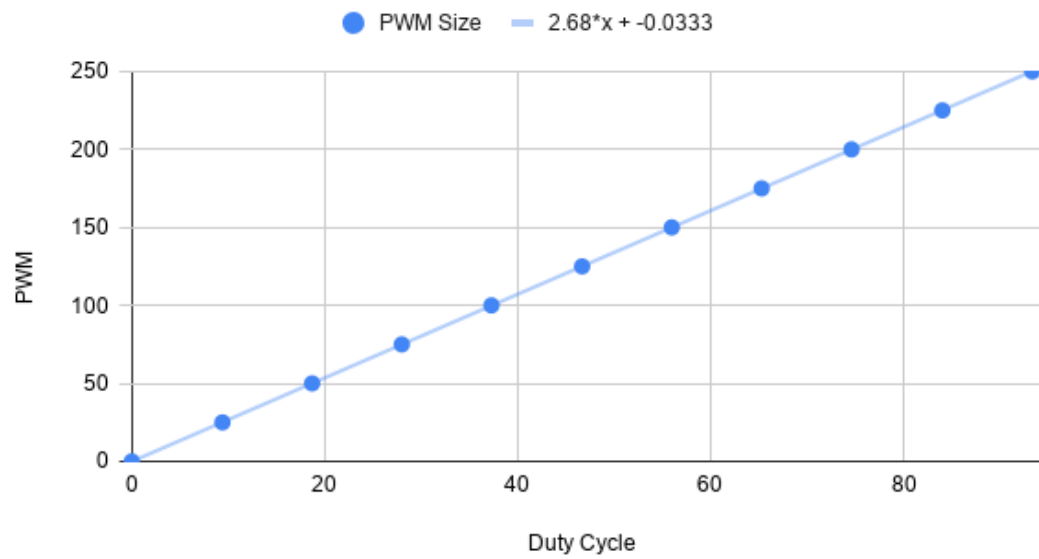


Fig. 7. Plot of the Duty Cycle that a PWM value creates

PotValue ADC vs. Pendulum Angle

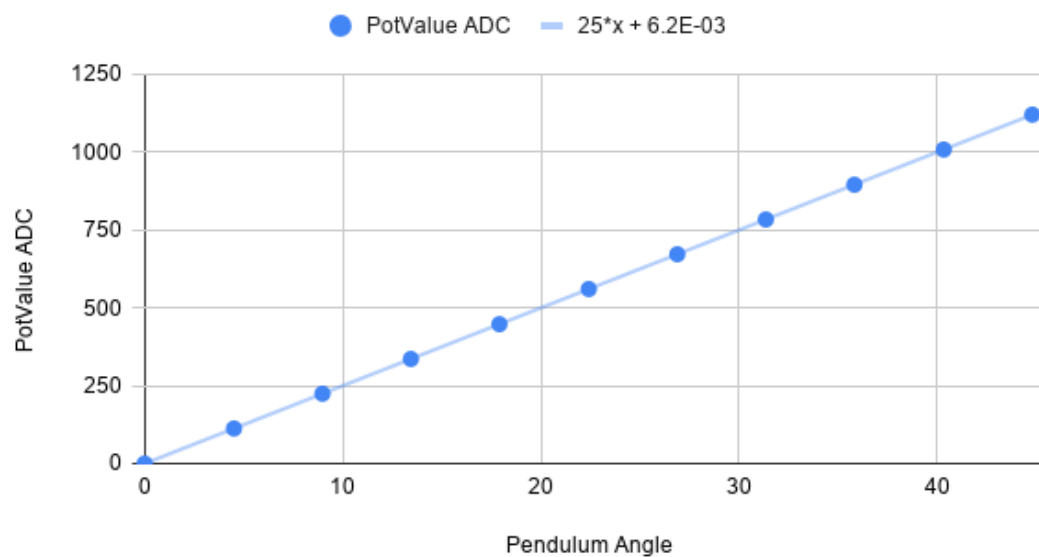


Fig. 8. Plot of the potentiometer ADC value that an angular displacement causes.

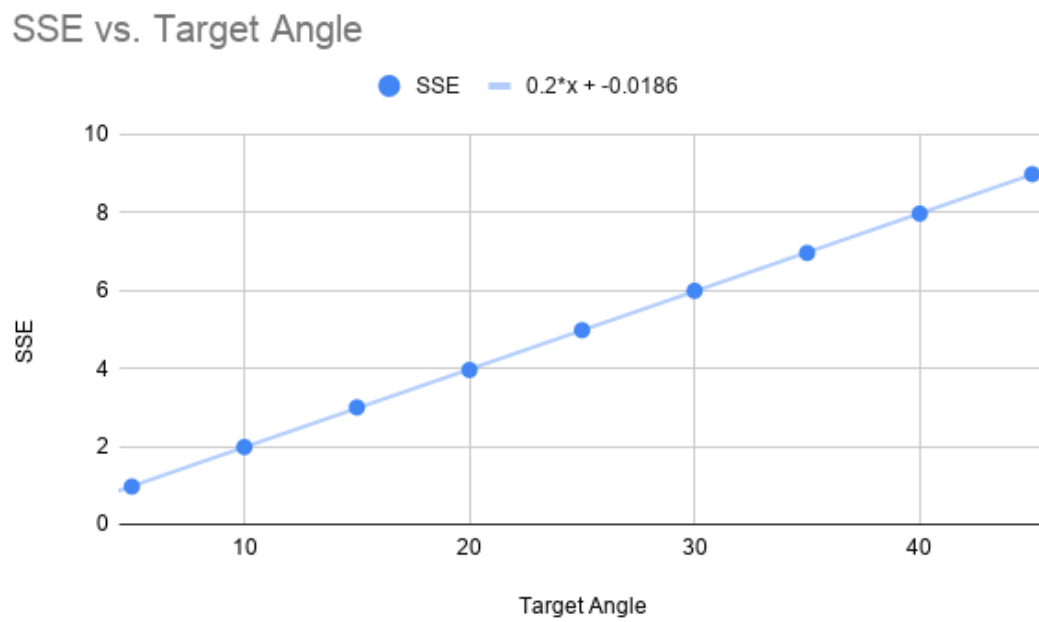


Fig. 9. SSE for a Given input angle when the system has error