George Mason University

Inverted Pendulum Experiment
Controller Design and Implementation

Report from laboratory experiment E conducted on 26 April 2016
As part of ECE 429 Control Systems Lab
Course Instructor: Dr. Daniel M. Lofaro

Michael Kepler
G00804828

28 April 2016

The Volgenau School of Engineering

Objective

The objective of this lab is to obtain the transfer function representation of the system so that a PD controller can be designed for it. Once the controller is designed, it shall be implemented using SIMULINK and QUARC hardware setup.

Theoretical Background

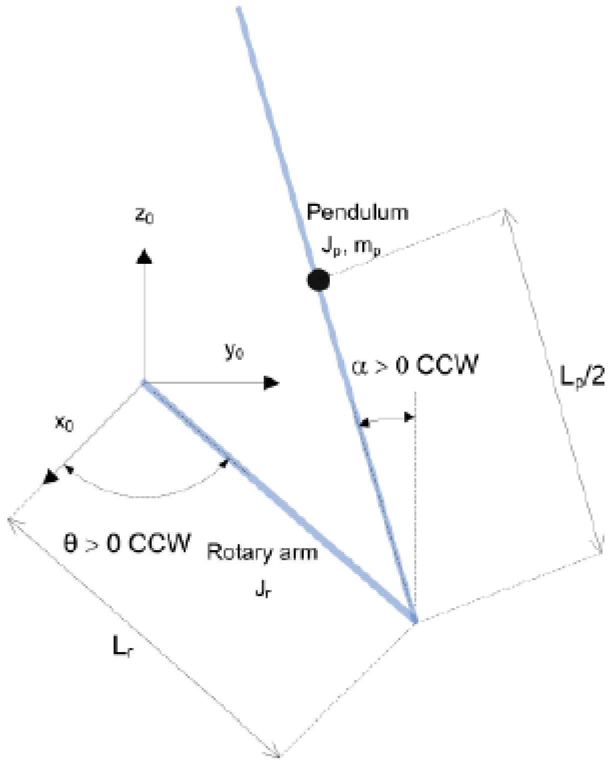


Figure 6: Inverted Pendulum Setup

There are two angles that will be fed back for the controller, the angle of the pendulum α(t) and the angle of the servo motor angle θ(t).

$$\ddot{\theta}(t) = 80.3\alpha(t) - 10.2\dot{\theta}(t) - 0.93\dot{\alpha}(t) + 83.2u(t) \quad (1)$$
$$\ddot{\alpha}(t) = 122\alpha(t) - 10.3\dot{\theta}(t) - 1.4\dot{\alpha}(t) + 80.1u(t) \quad (2)$$

Task 1

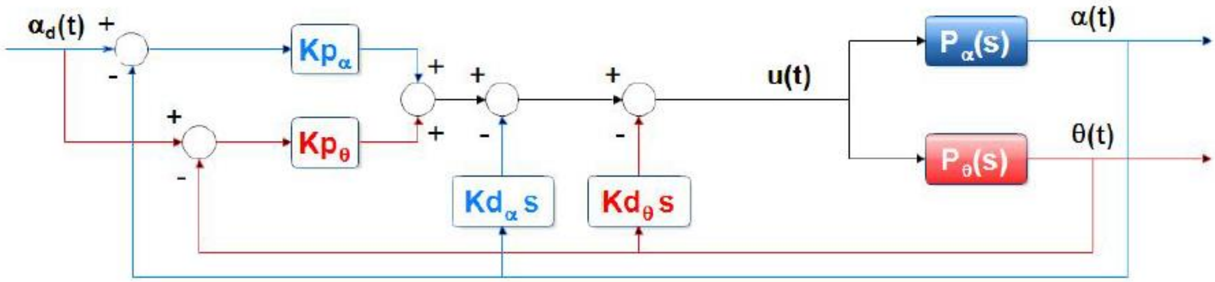
Applying the Laplace transform to eqns. (1) and (2) then solving for the transfer functions $P_\theta(s) = \frac{\theta(s)}{U(s)}$ and $P_a(s) = \frac{a(s)}{U(s)}$:

$$P_a(s) = \frac{80.1s^2 - 39.94s}{s^4 + 11.6s^3 - 117.3s^2 - 417.3s} \quad (3)$$

$$P_\theta(s) = \frac{83.2s^2 + 42s - 3718}{s^4 + 11.6s^3 - 117.3s^2 - 417.3s} \tag{4}$$

Task 2

Solve for the denominator of the closed loop system $G(s) = \frac{a(s)}{a_d(s)}$



$$\begin{aligned} &\textit{Denominator of } G(s) \\ &= s^4 + (11.6 + 80.1K_{da} + 83.2K_{d\theta})s^3 \\ &+ \left(-117.3 + 80.1K_{pa} - 39.94K_{da} + 83.2K_{p\theta} + 42K_{d\theta}\right)s^2 \\ &+ \left(-417.3 - 39.94K_{pa} + 42K_{p\theta} - 3718K_{d\theta}\right)s - (3718K_{p\theta}) \end{aligned} \tag{5}$$

The general form of a second order system with two poles:

$$(s^2 + 2\zeta w_n s + w_n^2)(s + p1)(s + p2) \tag{6}$$

$$\begin{aligned} &= s^4 + \left((p1 + p2) + 2\zeta w_n\right)s^3 + (p1p2 + 2\zeta w_n(p1 + p2) + w_n^2)s^2 + (p1p2)2\zeta w_n \\ &\quad + (p1 + p2)w_n^2)s + (w_n^2 p1p2) \end{aligned}$$

Comparing the coefficients of eqn. (5) and (6) yields four equations with four unknowns:

```
%              kda     kdt     kpa     kpt      contstants
sys_eqn = [ 80.1    83.2    0       0       (-11.6+s3);
            -39.94  42      80.1    83.2    (117.3+s2);
            0       -3718   -39.94  42      (417.3+s1);
            0       0       0       -3718    s0];
```

Where s0-s3 are the coefficients of eqn. (6). The row reduced echelon form of the equation above yields the values for the gains.

The gains calculations were done with the following MATLAB code:

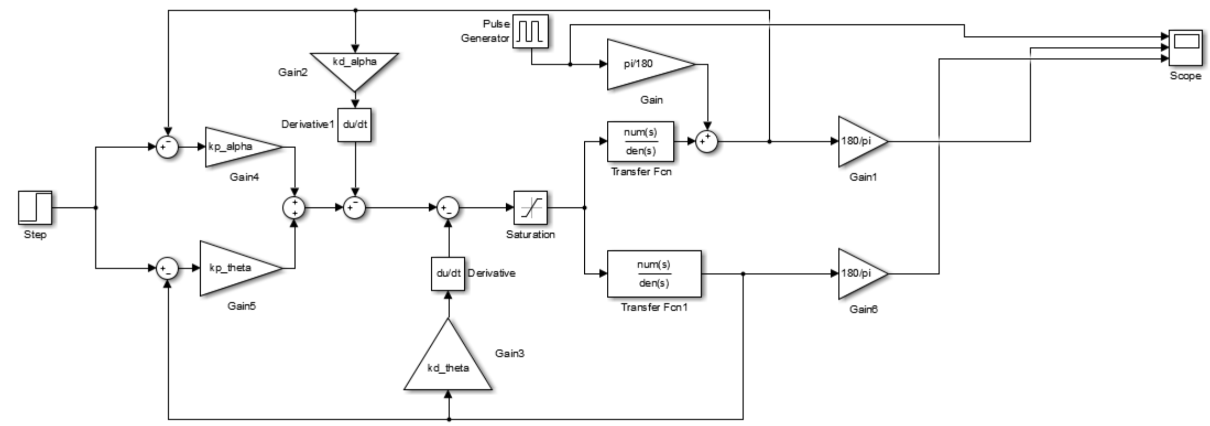
```
clear all

s = tf('s')

%alpha and theta transfer functions
alpha_num = [80.1 -39.94 0];
alpha_den = [1 11.6 -117.3 -417.3 0];
```

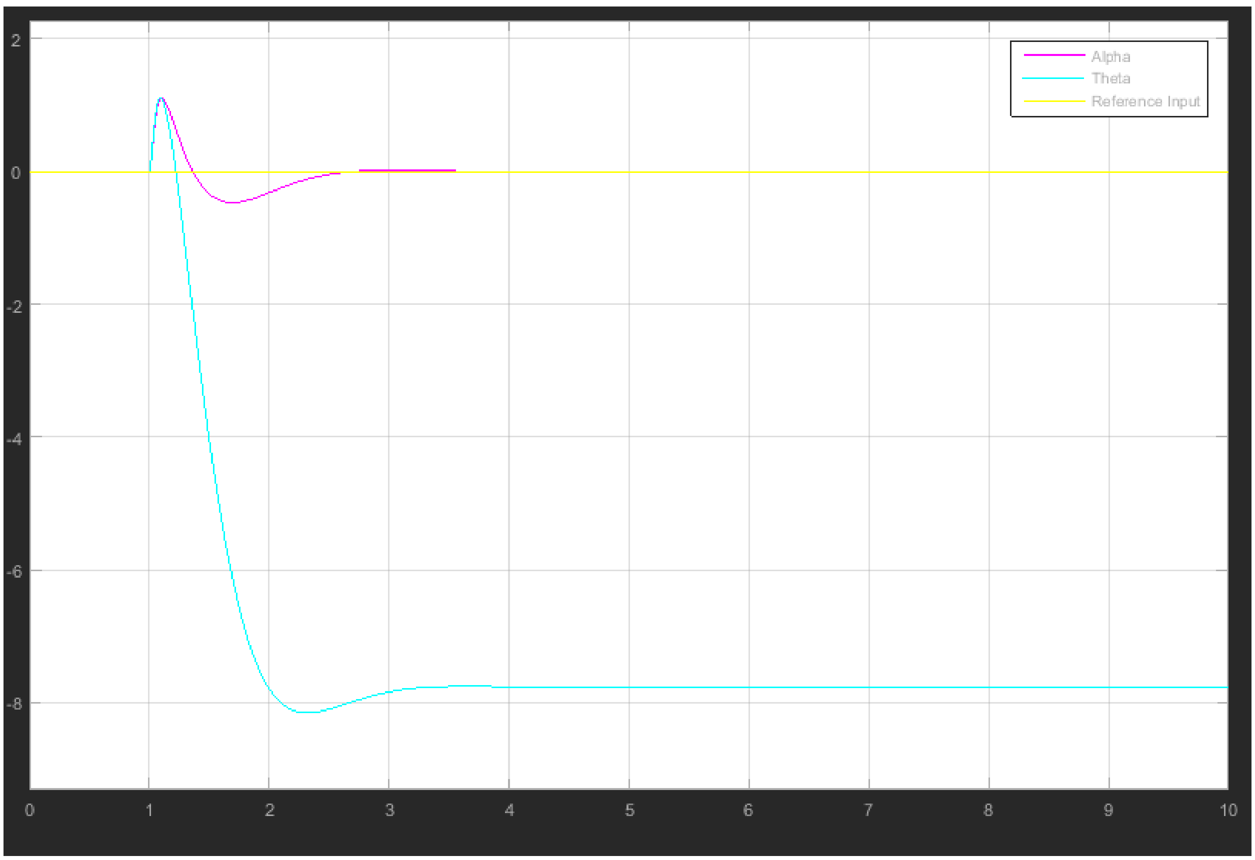
```
p_alpha = tf(alpha_num, alpha_den);

theta_num = [83.2 41.987 -3718.37];
theta_den = [1 11.6 -117.3 -417.3 0];
p_theta = tf(theta_num, theta_den);

%specifications
PO = 4.5;
ts = 1.75;

zeta = sqrt(log(PO/100)^2/(pi^2+log(PO/100)^2));
wn = 4/(ts*zeta);

%Coefficients of s to the power s# for (s+p1)(s+p2)(s^2+2zeta*wn+wn^2)
p1 = 30;
p2 = 40;

s3 = (p1+p2+2*zeta*wn);
s2 = p1*p2+2*zeta*wn*(p1+p2);
s1 = p1*p2*2*zeta*wn + (p1+p2)*wn^2;
s0 = wn^2*p1*p2;

%            kda    kdt    kpa    kpt     contstants
sys_eqn = [ 80.1    83.2   0      0      (-11.6+s3);
            -39.94  42     80.1   83.2   (117.3+s2);
            0       -3718  -39.94 42     (417.3+s1);
            0       0      0      -3718   s0];

sys_eqn_reduced = rref(sys_eqn)
kd_alpha = sys_eqn_reduced(1,5);
kd_theta = sys_eqn_reduced(2,5);
kp_alpha = sys_eqn_reduced(3,5);
kp_theta = sys_eqn_reduced(4,5);
```

Task 3

Having obtained the gain values, the system can now be implemented in SIMULINK. The reference input into the system should be 0 degrees to keep the pendulum position vertical.

Below is the SIMULINK model and simulation results:





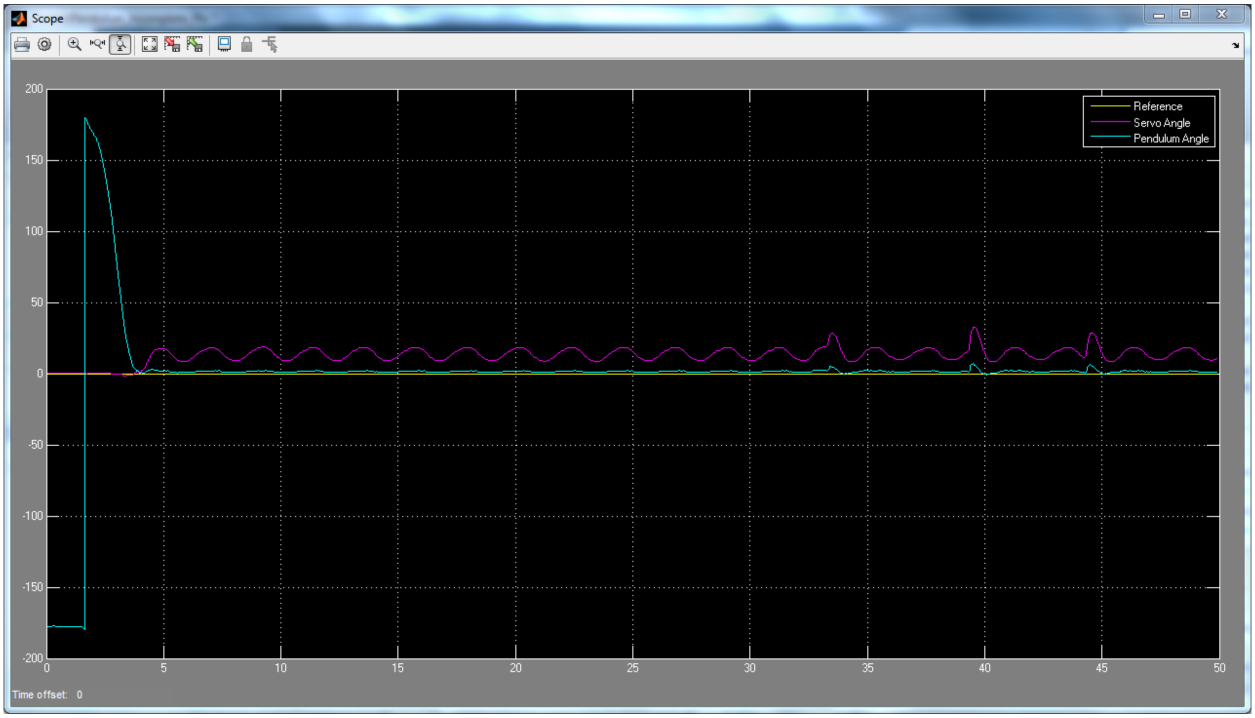
The controller meets the specifications given in the lab manual:

$$|a(t)| < 10\ degrees \qquad |\theta(t)| < 40\ degrees \qquad t_s \leq 2\ s.$$

Note from the simulation results the response settles back to zero at t = 2.5s. This is because the disturbance begins at t = 1s. Therefore the settling time is about 1.5s.

Task 4

Given that the system works in simulation, it can now be implemented in hardware. The first step was to run the student provided MATLAB script *InvertedPendulum_Parammeters.m.* Then the student provided SIMULINK model *ControlPendulum_Incomplete.slx* was completed with the gains calculated in task 2. The model was built and connected to the hardware. The pendulum was raised from the downward position to its vertical position. The results are displayed below:



The inverted pendulum achieves a settling time of nearly 1.75s with negligible overshoot, thus meeting the system requirements as stated in the lab. Notice the spikes at t = 33.5s, 39s, and 44s. These spikes are due to disturbances entering the system in the form of the pendulum being tapped. Despite the disturbances the system is still able to maintain suitable response and return 0 within 1.75s.

Conclusion

Given the equations for the dynamics of the pendulum-servo system, the gains were able to be obtained by deriving the closed loop equation of the controller and comparing the coefficients of 's' in the denominator of the closed loop equation with that of the equation of a second order system with two additional poles. Specifying the poles, settling time, and percent overshoot allows for the comparison of coefficients to comprise a system of equations to solve for the gains. Once having the gains the system can be simulated in hardware and software.