Simulation 4: Slider-crank with Air Spring

Brandon Reddish

November 12th 2018

1 Introduction

This lab looked at the dynamics of a piston in an air spring connected to a flywheel. The system that was modeled can be seen in Figure 1.

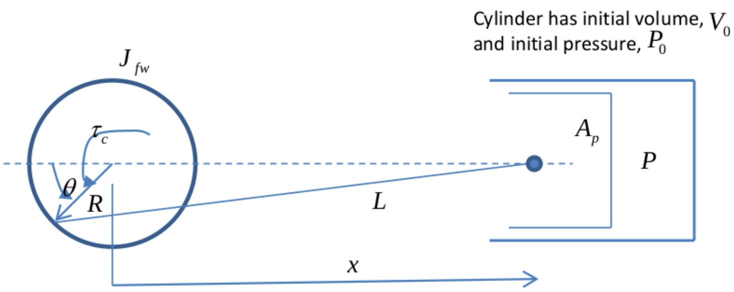

Figure 1: Model of Mass Striking a Barrier

The goal of this lab was to investigate the motion of the piston in relation to the flywheel and design a proportional controller to keep the flywheel mowing at a constant velocity.

1.1 Bond Graph Model

The first step in understanding the system was to model it with a bond graph. Figure 2 shows the simplified model and the associated bond graph.

The bond graph has two state variables $\dot{p}_{J_{fw}}$ and $\dot{q}_{air}$. The value of I for the inertial element will be J_{fw}. The compliance for the air spring will be modeled with Equation 1.

$$P = P_0[\frac{1}{[1 - \frac{A_p x}{V_0}]^\gamma} - 1] \tag{1}$$

It is assumed that the rate of change of θ is equal to the rotational velocity ω_{fw} as shown in Equation 2.

$$\dot{\theta} = \omega_{fw} \tag{2}$$

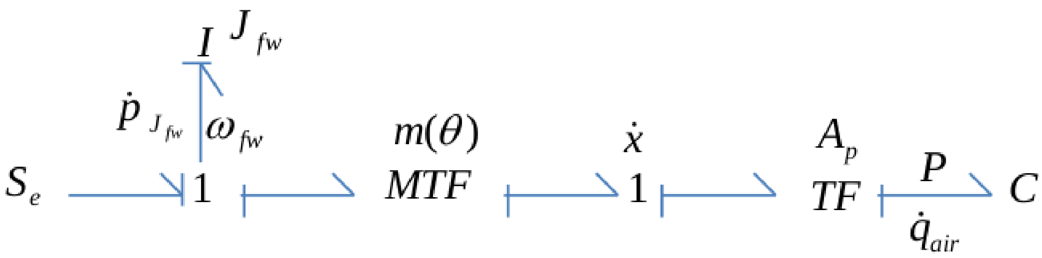

Figure 2: Simplified Model and Bond Graph

The modulated transformer in the middle of the bond graph relates the values of $\dot{\theta}$ to $\dot{x}$ with a function that depends on the variable θ. This relationship comes from the geometry of the piston location and the angle of the attachment point of the piston to the flywheel. Figure 3 shows the simplified geometry of the system.

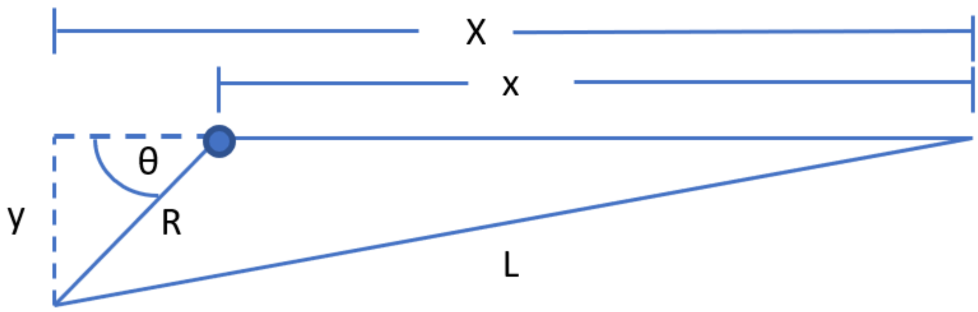

Figure 3: Simplified Geometry

The distance in the x direction form the piston pin to the attachment point on the flywheel can be calculated with the Pythagorean theorem shown in Equation 3.

$$L^2 = X^2 + y^2 \tag{3}$$

The variable of importance is x, but we will solve for X first for simplicity and then input the relationship between X and x. The value of y can be written in terms of know variables R and θ.

$$y = r * sin(\theta) \tag{4}$$

Plugging in this relationship yields the following equation.

$$L^2 = R^2 sin^2(\theta) + X^2 \tag{5}$$

By rearranging the equation, the value of X can be calculated as such.

$$X = \sqrt{L^2 - R^2 sin^2(\theta)} \tag{6}$$

The relationship between X an x is the value of x plus the value of $Rcos(\theta)$ as shown in Equation 8.

$$x + Rcos(\theta) = \sqrt{L^2 - R^2 sin^2(\theta)} \tag{7}$$

$$x = -Rcos(\theta) + \sqrt{L^2 - R^2 sin^2(\theta)} \tag{8}$$

Equation 8 provides a geometric relationship, but we are interested in the dynamics of the system. To get the relationship between the rotation velocity of the flywheel and the linear velocity of the piston, we must take the derivative of the geometric relationship. By doing this we get the final solution shown in Figure 9.

$$\frac{dx}{d\theta} = Rsin(\theta) - \frac{R^2 cos(\theta) sin(\theta)}{\sqrt{L^2 - R^2 sin^2(\theta)}} \tag{9}$$

With this equation, we can say that the first derivative of position $\dot{x}$ represents the velocity of the piston and the derivative of θ represents the angular velocity ω. As a result the expression shown in Eqaution 9 states that for some small change in $theta$ there will be a resulting change in x governed by the right hand side of the equation. Implementing the relationships of $V = dx$ and $\omega = d\theta$ the following equation shows the relationship between V and ω that will be used as the expression for the modulated transformer.

$$V = [Rsin(\theta) - \frac{R^2 cos(\theta) sin(\theta)}{\sqrt{L^2 - R^2 sin^2(\theta)}}] * \omega \tag{10}$$

The relationship in Equation 12 relates the flow variables. To get the relationship between show and effort, we must look at equation 11

$$FV = \tau\omega \tag{11}$$

Combining equations 12 and 11 provides us the a relationship between the torque of the flywheel and the force on the piston.

$$\tau = F * [Rsin(\theta) - \frac{R^2 cos(\theta) sin(\theta)}{\sqrt{L^2 - R^2 sin^2(\theta)}}] \tag{12}$$

1.2 State Equations

The state variables of the model are p_J and q_{air}. The rate of change of momentum of the flywheel p_J is equal to the value of the effort variable on that bond. That effort can be calculated with Equation 13.

$$\dot{p}_{J_{fw}} = S_e - m(\theta) * A_p * P_0[\frac{1}{[1 - \frac{A_p x}{V_0}]^\gamma} - 1] \tag{13}$$

The change in momnetum of the flywheel is dependent on theta and x which are not yet included in the calculations. To get these values, we must expand the state space. This can be done by using Equation 2 and Equation 14.

$$\dot{x} = m(\theta)\dot{\theta} \tag{14}$$

The value for ω_{fw} is calculated from the state variable $p_{J_{fw}}$ with the following relation.

$$\omega_{fw} = p_{J_{fw}}/J_{fw} \tag{15}$$

Lastly, the value for the air spring is modeled with Equation 16.

$$\dot{q}_{air} = A_p * m(\theta) * \frac{p_J}{J_f w} \tag{16}$$

1.3 Proportional Gain Controller

The controller for this system will control an external torque applied to the flywheel. The amount of torque applied will be some constant times the difference between the desired angular velocity and actual angular velocity of the flywheel. Equation 17 shows the equation used for the simulation.

$$\tau_c = K_p(\omega_{fw_{des}} - \omega_{fw}) \tag{17}$$

2 Running the Simulation

The simulation was run with the parameters shown in Table 1.

Table 1: Simulation Parameters

Parameter	Value	Units
m_{fw}	4.545	kg
R	0.1016	m
L	0.4064	m
J_{fw}	0.0235	kgm^2
D_p	0.1016	m
A_p	0.0081	m^2
$V_s t$	0.0016	m^3
V_{TDC}	$2.4e-04$	m^3
V_0	0.0019	m^3
P_0	1e5	N/m^2
$\omega_{fw_{des}}$	1500	RPM
γ_{air}	1.4	

Figures 4 through 7 show the response of the system with a controller gain of $K_p = 0.1$. For this simulation, the momentum of the flywheel was set as $2kg \cdot m^2/sec$. The relationship of theta and x is governed by the geometric equation shown in Equation 8. This relationship is shown in Figure 4.

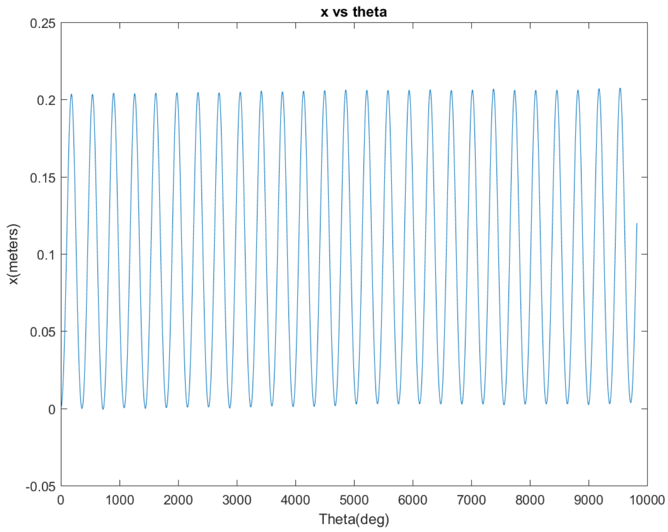

Figure 4: x Location vs Theta

The plot shown in Figure 5 shows how the velocity of the piston relates to the angular velocity of the flywheel. As the angular velocity of the flywheel increases, the maximum velocity of the

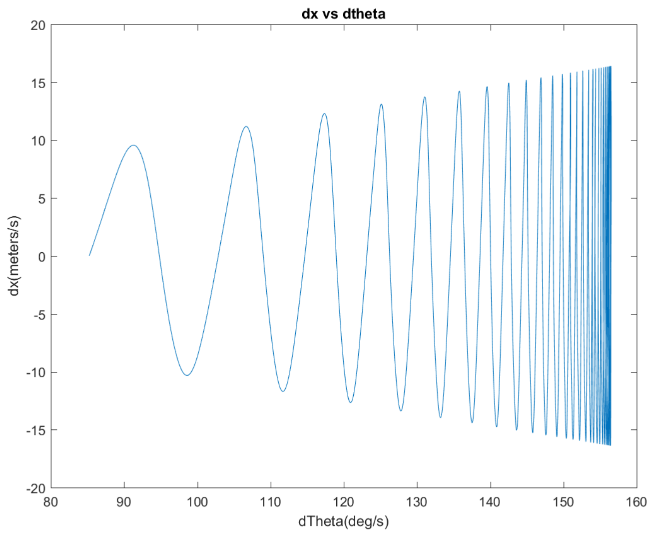

Figure 5: x Velocity vs Angular Velocity

piston increases.

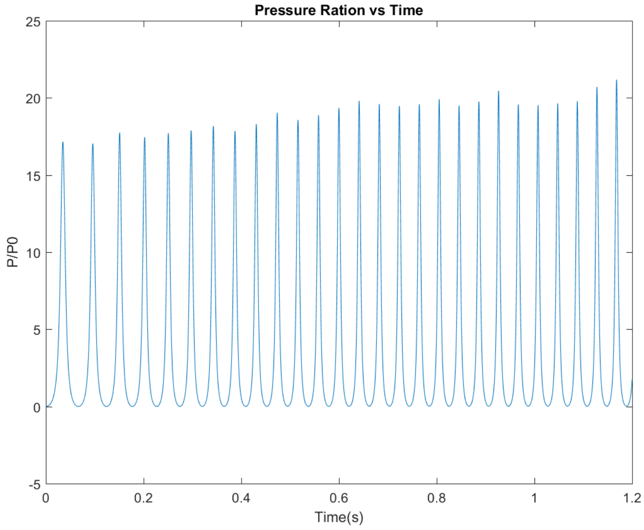

Figure 6: Pressure Ratio

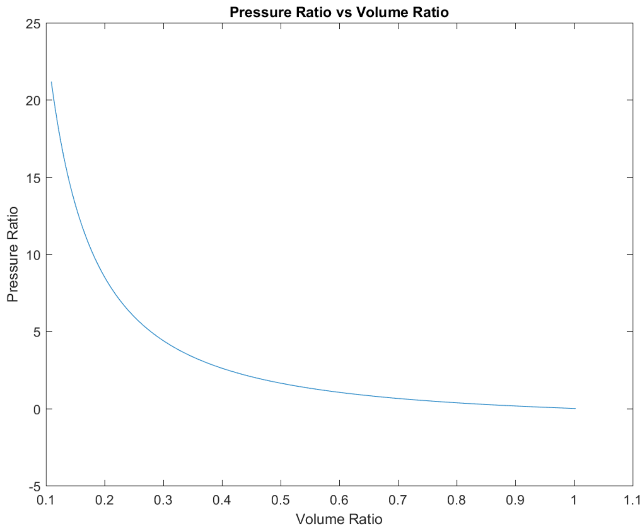

Figure 7: Pressure Ratio vs Volume Ratio

The gain of the controller is most apparent in Figure 8. This simulation was run with $K_p = 0.1$.

The simulation begins with the flywheel spinning at a significantly lower speed that the desired 1500 RPM. As a result, an additional torque is applied at a rate proportional to the difference in desired and actual angular velocities. One can see that the desired RPM is reached at a time of about 1.2 seconds. For comparison, the same simulation was run with a gain of 0.2 and the response is shown in Figure 9.

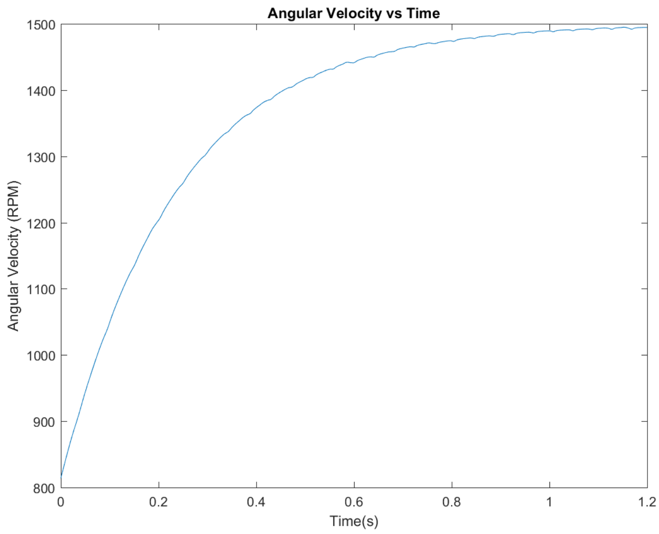

Figure 8: Angular Velocity Over Time ($K_p = 0.1$)

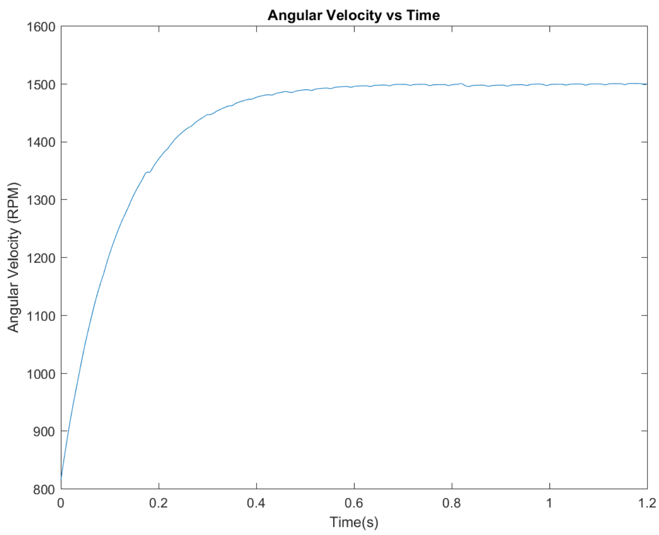

Figure 9: Angular Velocity Over Time ($K_p = 0.2$)

By doubling the gain, the system reached the desired rpm much faster. One thing to note is that once the flywheel reaches the desired RPM, the values fluctuate around 1500 RPM. Figure 10 shows a section of the plot in Figure 9.The fluctuation is within 5-6 RPM, but this plot shows that the proportional gain controller reacts to the system after it has deviated from the desired steady state.

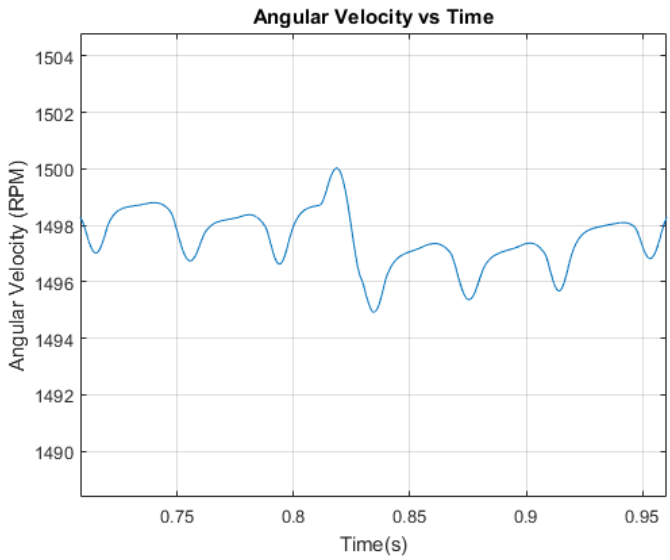

Figure 10: Angular Velocity Over Time ($K_p = 0.2$) Zoomed In

3 Source Code

The following code is broken into three sections. The first section is the main.m script that will set the variable, call the ode45 function, and plot the results. The second section contains the state equations that govern the system. The last section is a small function that is used to calculate the value of $m(\theta)$. This function also takes in the values for R and L although they do not change during the simulation.

3.1 Main Script

```
function main()
%% Set parameters
global Jfw L V0 P0 omegaDes Kp Ap R gamma
close all
%% Parameters and initial conditions
mfw = 10/2.2; % kg
R = 4*(0.0254); % m
L = 4*R; % m
Jfw = mfw * (R^2) /2; %kg m^2
Dp = 4*(0.0254); %m
Ap = pi * (Dp^2) /4;%m^2
Vdisp = Ap*2*R; %m^3
Vtdc = 240*1e-6; %m
V0 = Vdisp + Vtdc;
P0 = 155;%N/m^2
gamma = 1.4;
%% Run setup
```

```
initials = [0 2 0 0]; % [airSpring flywheelMomentum x theta]
timeStep = 0.0001;
tfinal = 1.2;
tspan = 0:timeStep:tfinal;
% Desired angular velocity
omegaDes = 1500*0.104719;%rpm to raidians/sec
%% Controller gain
Kp = 0.1; % gain of controller
%% Run ODE
[t,x] = ode45(@equations,tspan,initials);
% Extract derivatives and other values
for i = 1:length(t)
    [dx(i,:),oth(i,:)] = equations(t(i),x(i,:));
end
%% Post Proc
% Plot momentum
time = t(:,1);
spring = x(:,1);
momentum =  x(:,2);
angularVel = momentum/Jfw;
xLoc = x(:,3);
% limit theta for range 0<x<360 and convert to deg
for i = 1:length(x(:,4))
    x(i,4) = x(i,4)*180/pi;
end
theta= x(:,4);
dxLoc = dx(:,3);
dtheta = dx(:,4);
%% Plotting
% plot x vs theta
figure(1);
plot(theta,xLoc)
title('x vs theta')
xlabel('Theta(deg)')
ylabel('x(meters)')
print('xvstheta','-dpng')
grid on
% plot dx vs dtheta
figure(2);
plot(dtheta,dxLoc)
title('dx vs dtheta')
xlabel('dTheta(deg/s)')
ylabel('dx(meters/s)')
print('dxvsdtheta','-dpng')
grid on
%plot pressure over p0 vs time
figure(3)
plot(time,oth(:,1)/P0)
title('Pressure Ration vs Time')
```

```
xlabel('Time(s)')
ylabel('P/P0')
print('poverp0','-dpng')
grid on
%plot angular velocity over time
figure(4)
plot(time,angularVel*9.5492965964254)
title('Angular Velocity vs Time')
xlabel('Time(s)')
ylabel('Angular Velocity (RPM)')
print('angvelovertime','-dpng')
grid on
%plot pressure ratio over volume ratio
figure(5)
plot(oth(:,2)/V0,oth(:,1)/P0)
title('Pressure Ratio vs Volume Ratio')
xlabel('Volume Ratio')
ylabel('Pressure Ratio')
print('volratpresrat','-dpng')
grid on
end
```

3.2 Equation Script

```
    function [dx,oth] = equations(t,x)
% Global variables
global Jfw L V0 P0 omegaDes Kp Ap R gamma
% Unpack variables
q     = x(1); % spring state
p     = x(2); % momentum
xLoc = x(3); % mass location
theta= x(4); % angle of flywheel
% Expaned state space to cacluate dx and dtheta
dtheta = p/Jfw; % dtheta = omegafw = p/J
dxLoc = MTF(theta,L,R)*dtheta;
%Proportional controller
torque = Kp*(omegaDes - dtheta);
%State equations
dp = torque - MTF(theta,L,R)*Ap*P0*((1/(1-(Ap*xLoc/V0))^gamma)-1);
dq  = Ap * MTF(theta,L,R) * p/Jfw;
%Additional values for analysis
p =  P0*((1/(1-(Ap*xLoc/V0))^gamma)-1);
V = (V0-Ap*xLoc);
dx =[dq;dp;dxLoc;dtheta];
oth = [p V]; % pressure and volume
end
```

3.3 Modulated Transformer

```
    function mtf = MTF(theta,L,R)
%Modulated transformer function
mtf = R*sin(theta) -...
    ( (R^2 *cos(theta)*sin(theta))/(sqrt(L^2-R^2 * (sin(theta))^2)) );
end
```