Inverted Pendulum

ECEN 5638 - Controls Lab

Riley Kenyon

04/16/2020

1. Explain How the Inverted Pendulum was Modeled and Identified

The inverted pendulum was modeled via euler langrange equations. The lagrangian was determined with the reference of $\alpha = 0$ as the inverted upright position. The components of the lagrangian include the potential energy from the upright beam, with mass-moment of inertia with respect to the horizontal arm's vertical position. The expression for potential energy is described as $P = mgLcos(\alpha)$.

The kinetic energy of the lagrangian includes the kinetic energy of the horizontal arm rotating, the Kinetic energy of the vertical bars' rotation, and the linear kinetic energy of the bar. In total, the expression for kinetic energy is

$$K = \frac{1}{2}J_m\dot{\theta}^2 + \frac{1}{2}J_L\dot{\alpha}^2 + \frac{1}{2}m_{bar}v^2$$

Converting to the euler-lagrange equations, the differential equation can be written in state space. The matrices that describe the state-space model can be condensed into generalized matrices to represent Mass, damping and coriolis effects. This results in the non-linear form of the model.

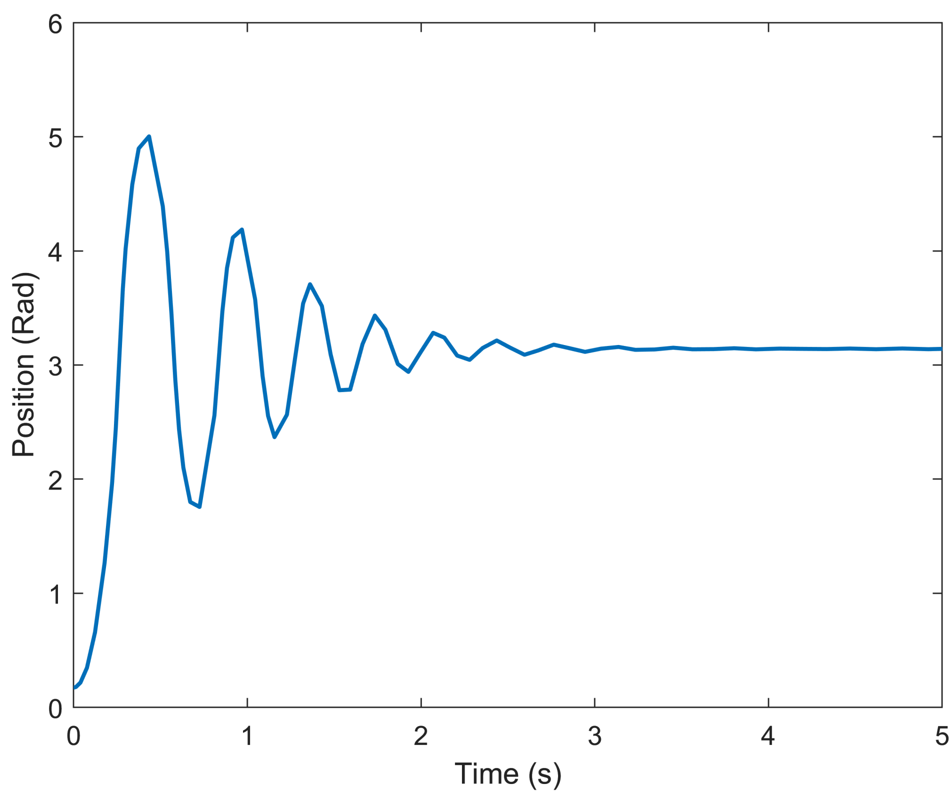
$$\begin{bmatrix} J_m + md^2 & -mld\cos\alpha \\ -mld\cos\alpha & 4/3ml^2 \end{bmatrix} \begin{bmatrix} \ddot{\theta} \\ \ddot{\alpha} \end{bmatrix} = \begin{bmatrix} -mld\sin\alpha\dot{\alpha}^2 - b_\theta\dot{\theta} \\ mgl\sin\alpha - b_\alpha \end{bmatrix} + \begin{bmatrix} \mathrm{k}_m \\ 0 \end{bmatrix} u$$

In order to identify the parameters of the system, the system is linearized about $\theta = 0, \alpha = \pi$ (stable system), and their subsequent derivatives equal to zero. The information gained from the identification will be applied to the unstable system, linearized about the position $\alpha = 0$. The linearized form of the transfer functions allow the parameters to be determined from the experimental bode-plot for each transfer function: $\frac{\alpha}{\theta}$, $\frac{\theta}{U}$ and therefore $\frac{\alpha}{U}$ which is the desired relation. $\frac{\alpha}{\theta}$ yeilds the parameter b_α from the experimental data and measured parameters m, l, d & g. For the function $\frac{\theta}{U}$, the vertical arm is removed and simplified to determine the parameters k_m/J_m, b/J_m. The experiment is continued with a stationary vetical arm to get relations for two more combinations of parameters $k_m/(J_m + md^2)$ and $(J_m + md^2)/J_m$. At this point, the vaules of each of the parameters is determined by using measurable known quantites of m and d.

The final experimental bode plot is used as a confirmation to validate the parameters were identifed correctly for the relationship between θ/U.

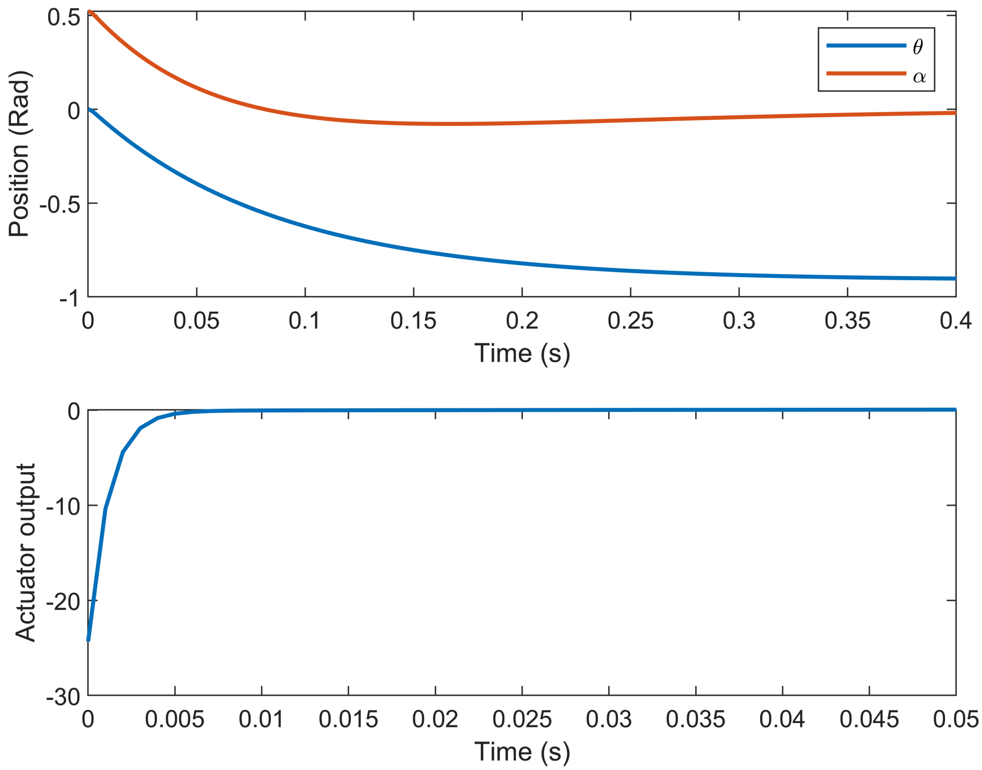
2. Generate a Non-Linear Simulink Model for the Inverted Pendulum

```
clear all; close all; clc;

% Base parameters
m = 0.03;
L = 0.053;
d = 0.085;
g = 9.8;

% Parameters from the system identification
J_m = 5.42e-5;
k_m = 0.0413;
b_theta = 5.36e-4;
b_alpha = 4.9734e-5;

% Construct non-linear model in simulink
d_alpha = deg2rad(10);
data = sim('invertedPendulum_nonlinear.slx');
figure(1);
plot(data.test.Time,data.test.Data(:,2),'LineWidth',1.5)
xlim([0 5])
xlabel('Time (s)')
ylabel('Position (Rad)')
```



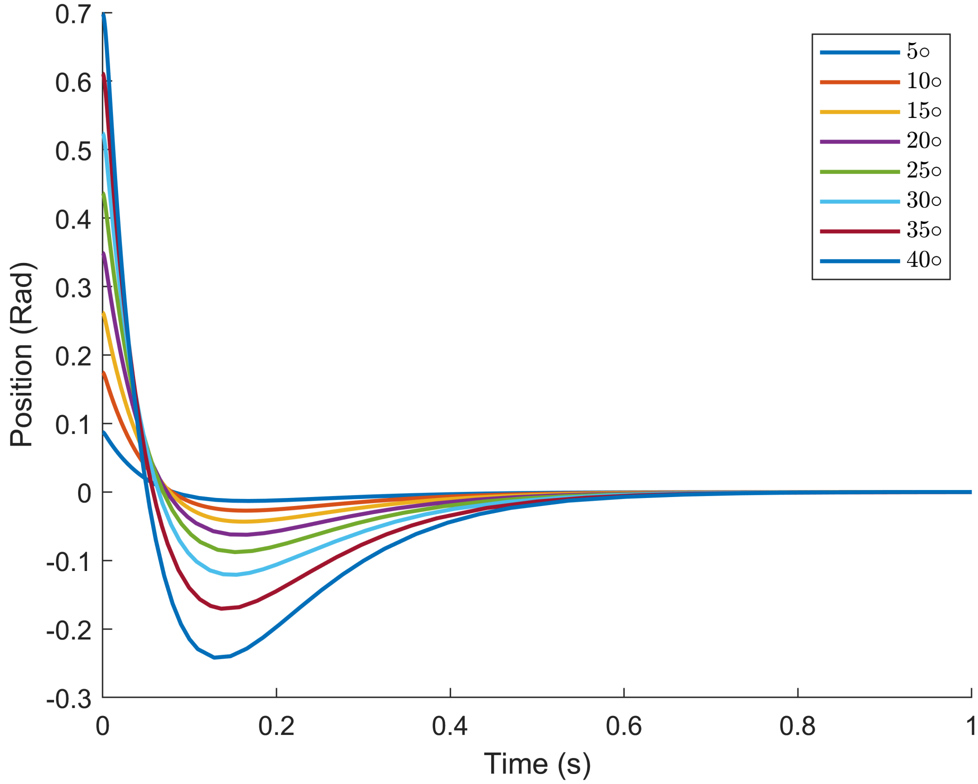
3. Design a Controller Linearized about $\alpha = 0$

```matlab
% Create linearized model
M = [J_m+m*d^2, -m*L*d; -m*L*d, 4/3*m*L^2];
V = [0 0 -b_theta 0; 0 m*g*L 0 -b_alpha];
A = [0 0 1 0; 0 0 0 1; M\V];
B = [0;0;M\[k_m;0]];
C = eye(4);
D = zeros(4,1);
model = ss(A,B,C,D);

% Choose LQR
Q11 = 100;          % error for theta
Q22 = 0.01;         % error for alpha
Q33 = 0.1;          % error for alpha_dot
Q44 = 10;           % error for theta_dot
Q = diag([1/Q11,1/Q22,1/Q33,1/Q44]);         % should be a 4x4 matrix x'Qx

R11 = 0.5;                % should be a scalar u'Ru
R = 1/R11;
[K,~,~] = lqr(A,B,Q,R); % get controller gains

% Construct closed loop
A_CL = (A-B*K);
B_CL = zeros(4,1);
C_CL = eye(4);
D_CL = zeros(4,1);
model_CL = ss(A_CL,B_CL,C_CL,D_CL);

% Command line simulation on linearized model
d_alpha = deg2rad(30);
t = 0:0.001:2;
u = zeros(1,length(t));
[y,t,x] = lsim(model_CL,u,t,[0 d_alpha 0 0]'); % Displace from equilibrium
u = -K*x';
u = u';

figure(2), hold on
subplot(2,1,1)
plot(t,y(:,[1,2]),'LineWidth',1.5);
legend('\theta','\alpha');
xlim([0,0.4]);
xlabel('Time (s)')
ylabel('Position (Rad)')

subplot(2,1,2)
xlabel('Time (s)')
plot(t,u,'LineWidth',1.5);
ylabel('Actuator output');
xlabel('Time (s)')
xlim([0,0.05]);
```

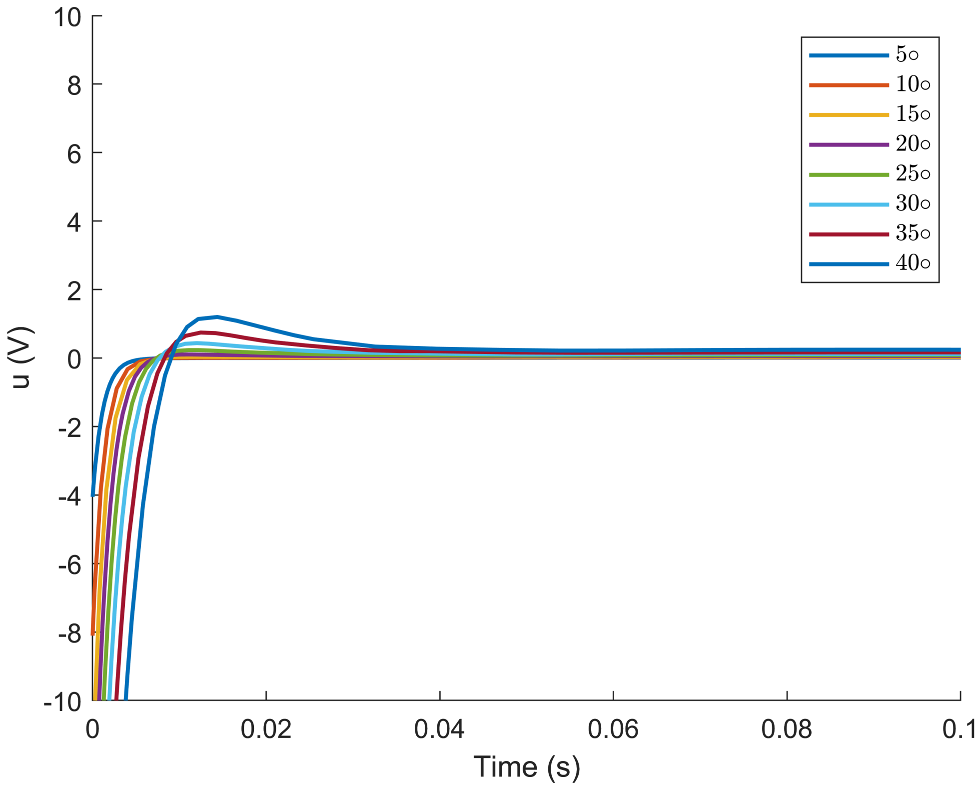


4. Implement the Control Law on the System Initialized Close to the Origin

```
clear t;

% Simulation parameters
T = 5;                                                  % Simulation time
figure(3); hold on
figure(4); hold on
for ii = 1:8
    d_alpha = deg2rad(ii*5);
    data = sim('invertedPendulum_nonlinear_controller');   % Simulink
    position{ii}=data.states;
    t{ii} = data.time;
    figure(3)
    plot(data.time,data.states(:,2),'Linewidth',1.5)
    figure(4)
    plot(data.time,data.u,'Linewidth',1.5)
end
figure(3)
xlim([0 1])
xlabel('Time (s)')
ylabel('Position (Rad)')
legend('$5\circ$','$10\circ$','$15\circ$','$20\circ$','$25\circ$','$30\circ$','$35\circ$','$40\
```




```
figure(4)
xlim([0 0.1])
ylim([-10 10])
xlabel('Time (s)')
ylabel('u (V)')
legend('$5\circ$','$10\circ$','$15\circ$','$20\circ$','$25\circ$','$30\circ$','$35\circ$','$40\
```



Initializing alpha at greater values increases the output required by the controller. After about 40 degrees, the results become undesirable and unstable.

5. Design a Nonlinear Control Law on the System that Swings the Pendulum to the Upright Position

Define the total energy of the pendulum only

$$P = mglcos(\alpha)$$

$$K = \frac{1}{2}J_L\dot{\alpha}^2$$

$$E = mglcos(\alpha) + \frac{1}{2}J_L\dot{\alpha}^2$$

The maximum total energy of the pendulum is undefined because the potential and kinetic energy are uncoupled, and with $\dot{\alpha}$ being able to become arbitrarily high the limit is infinite. However, due to the friction with the bearing of thependulum, there is some limit. The maximum potential energy occurs at the position $\alpha = 0$ with a value of $P = mgl$.

The time derivative of the total energy of the pendulum is therefore

$$\dot{E} = -mgl\sin\alpha\dot{\alpha} + J_L\dot{\alpha}\ddot{\alpha}$$

In order to simplify the model and make designing a control for the swing up procedure. The relation of the original dynamics

$$-mld\cos\alpha\ddot{\theta} + J_L\ddot{\alpha} = mgL\sin\alpha - b_\alpha\dot{\alpha}$$

is simplified using the assumption $(J_m + md^2) >> ml$, thereby allowing

$$(J_m + md^2)\ddot{\theta} = k_m u$$

or more usable

$$\ddot{\theta} = \frac{k_m u}{J_m + md^2}$$

assuming $b_\alpha \approx 0$ and backsolving for the derivative of energy yields

$$\dot{E} = \frac{ml\cos\alpha\dot{\alpha}k_m}{J_m + md^2}u$$

From this relation, in order for the energy of the pendulum to be monotonically increasing, the sign of $\cos(\alpha)\dot{\alpha}u > 0$. The sign function is then used to determine the if the argument is greater than zero. In addition, the controller is designed to be proportional to the error of the current energy with respect to the maximum potential energy, mgl. The gain *K* is then used to increase the control output *u* until the actuator overcomes friction and sinusoidally increases to mgl. In other words,

$$u = K(E_{max} - E)sign(\cos\alpha\dot{\alpha})$$

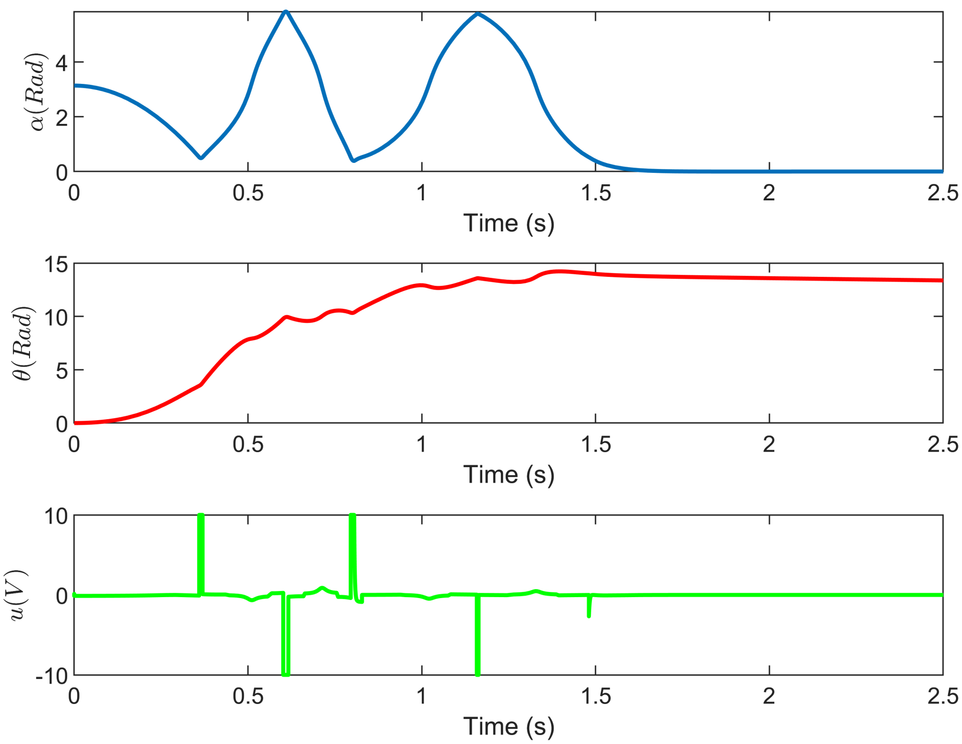
The benefits to having the energy of the system monotonically increasing is similar to that of a playground swing. At each instant, energy in being imparted to the system accelerating the pendulum while the system is simultaneously exchangin potential and kinetic energy. At some point the energy of the system is great enough to be within a margin of the vertical position of $\alpha = 0$. An additional benefit of energy of the system monotonically increasing is that does not waste energy fighting the dynamics of the system, and can reach the vertical position more efficiently.

6. Desgin and Simulate a Controller that Switches from the Swing-Up Procedure to the Linearized Feedback

In order to implement a switching controller, a switch block was used to use either the swing-up procedure or the LQR controller depending on the angle of the pendulum. To accomplish this, the angle was mapped between [0,2*π*] and if the angle was greater than *π*, it was mapped to [0,*π*]. The switching could also been done based off the energy of the pendulum. The output of the switch was fed through a saturation block to eliminate the requests for large actuator authority, it was restricted to operate between [-10,10] volts.

```
% start pendulum at equilibrium (alpha = pi)
data = sim('invertedPendulum_nonlinear_controller_swingup_switch.slx');
figure,
```

```
subplot(3,1,1)
plot(data.time,data.states(:,2),'LineWidth',1.5)
xlim([0 2.5])
xlabel('Time (s)')
ylabel('$\alpha (Rad)$','interpreter','latex')

subplot(3,1,2)
plot(data.time,data.states(:,1),'r','LineWidth',1.5)
xlim([0 2.5])
xlabel('Time (s)')
ylabel('$\theta (Rad)$','interpreter','latex')

subplot(3,1,3)
plot(data.time,data.u,'g','LineWidth',1.5)
xlim([0 2.5])
xlabel('Time (s)')
ylabel('$u (V)$','interpreter','latex')
```



The results show that the controller stabilizes the inverted pendulum in the upright position, with the value of θ drifting slightly with time. The controller is saturated to output the max amplitude voltages of the power supply, which occurs during two of the swing up oscillations. The gain of the swing up controller $K = 4$ and the LQR controller takes over when the pendulum is within 30 degrees of the upright position.