t-ches-csuc / MECA482-Furuta_Pendulum Private

0 stars 0 forks

Star Watch

Code Issues Pull requests Actions Projects Security Insights

main

QuincyatChico Update README.md ... 3 minutes ago 84

View code

Furuta Pendulum

MECA 482: *Keegan Penso, Tyler Chesney, Mohit Bhardwaj, Quincy Owyang, Jack Russo*

Introduction

1. Background

A Futura Pendulum is a simple Control Theroy learning tool. It is comprised of a rotating arm and a pendulum attached to the end. The model neglects friction, when considering the mass relative to the fricton of the bearings in the pendulum it is a good assumtion that the coefficient of friction is too small to significantly impact the model. A real system would include encoders for position feedback of the arms, this is achieved programmatically in the simulation.

README.md

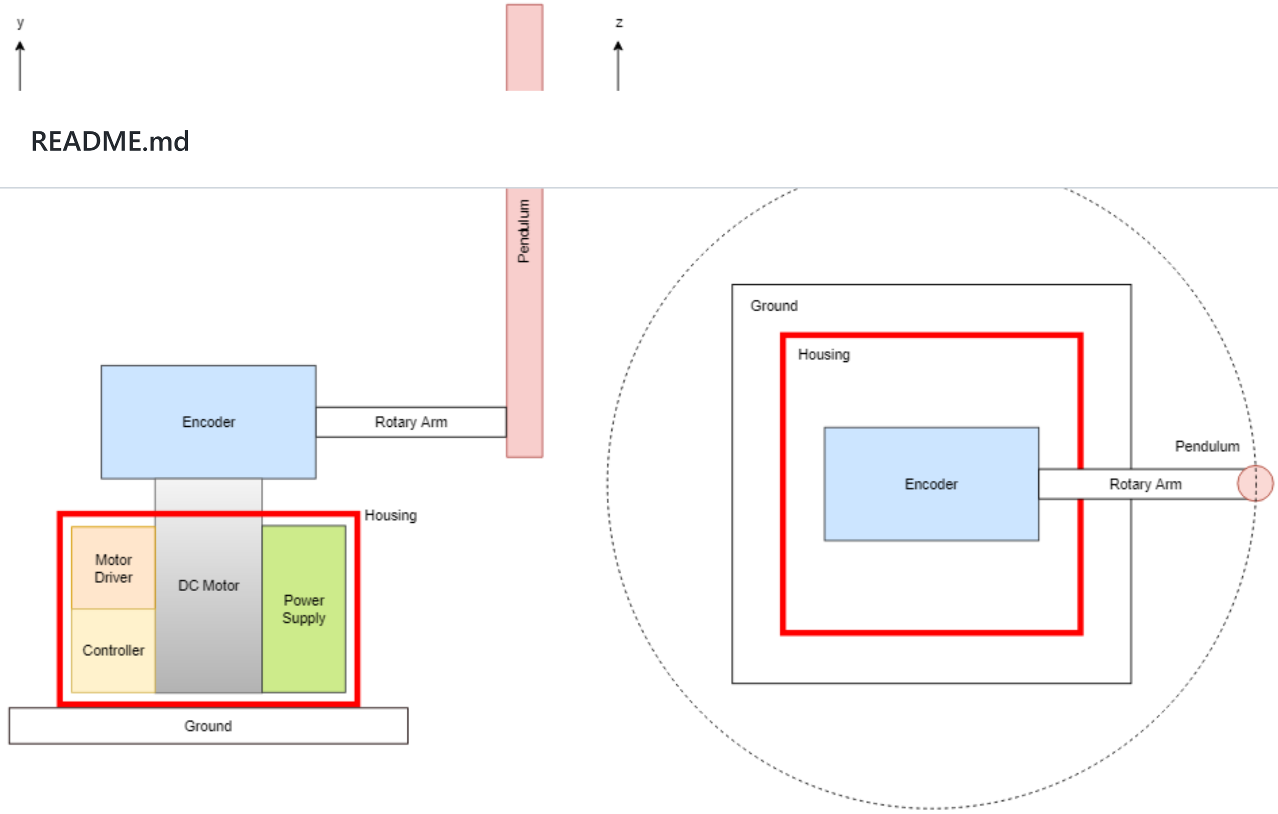

Figure 1: Operational viewpoint for the pendulum.

The Operational Viewpoint serves as a physical baseline for development of the overall control system. For a clearer understanding of the control hardware required for the system and their functional configuration, a functional viewpoint has been developed. Seen below in Figure 2 is the logical/functional viewpoint.

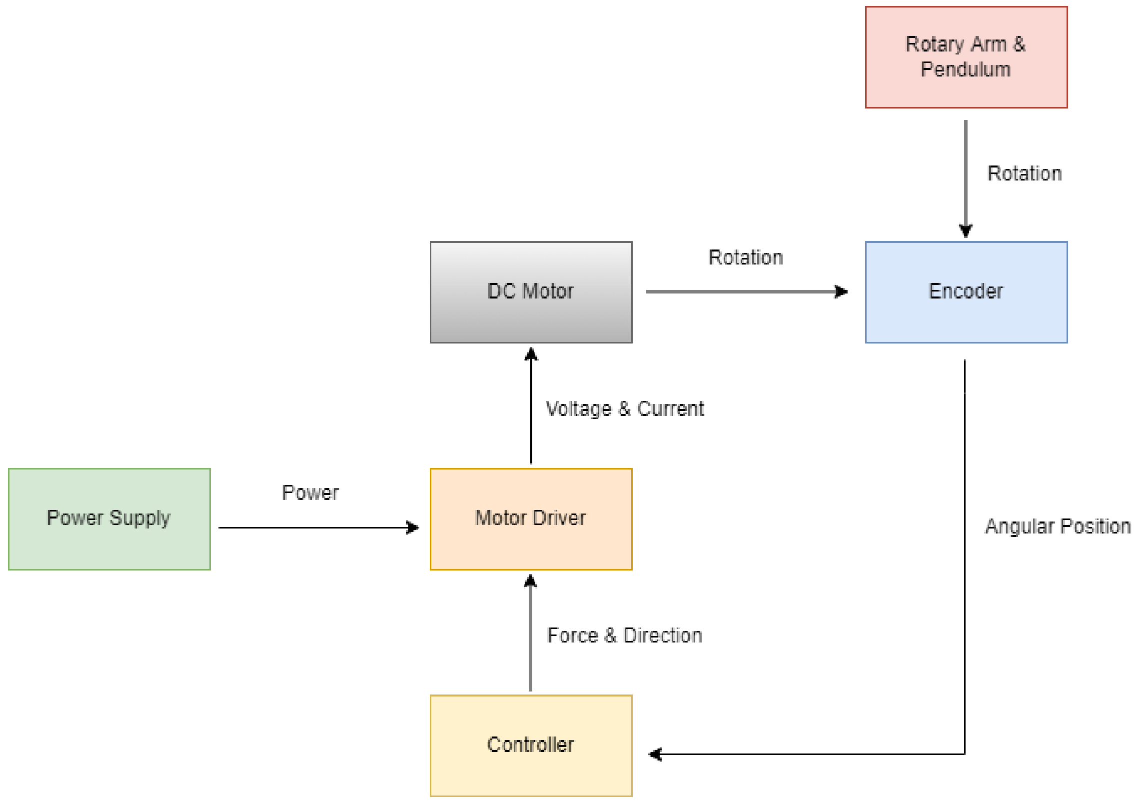

Figure 2: Logical/functional viewpoint for the pendulum.

When used in tandem, the two viewpoints allow for a full understanding of the control system.

The modeling presented in this paper couples the two systems. As will be discussed, one degree of freedom in each system can easily be reduced to a single constant dramatically simplifying the system. Vikash Gupta puts forward a well modeled Simulink model with full state feedback covering all necessary state variables.

The following report consists of documentation pertaining to the solution obtained by the team. In order to acheive the desired functionality a mathematic model wast obtained for the system, from there a controller architechture was devised to simulate and implement the derived mathematical model. The following report consists of documentation of the control system design process.

2. Resources

Listed below are several key resources utilized by the team throughout the control system desing process (for full citation please see "References"):

- *Control System Engineering*: 7th Edition; Norman S. Nice
- *Mathworks MATLAB Central*: Full State Feedback of Furuta Pendulum; Vikash Gupta

Modeling

1. Schematic

To appropriately model the system, a schematic diagram of the pendulum was used from Quanser[2]. This schematic defines the variables to be used in this system.

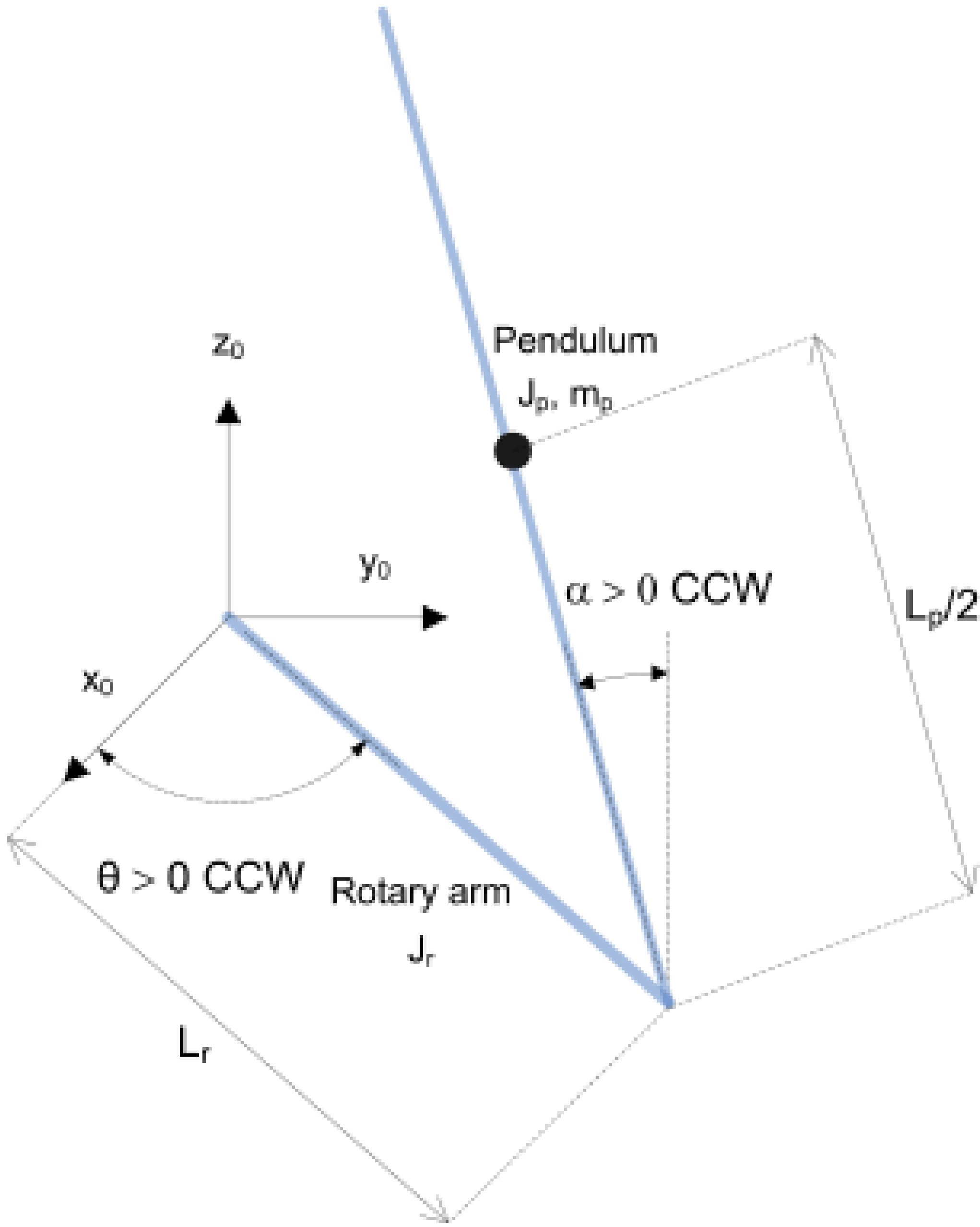

Figure 3: System schematic diagram.

The following table depicts definitions based on the above schematic for system parameters used in the derivation of the mathematical model.

Table 1: Parameter definitions relevant to preparing the mathematical model for the inertia wheel pendulum.

Symbol	Description
θ	Rotational angle about the z-axis which corresponds to the DC motor.
$\dot{\theta}$	Angular velocity about the z-axis.
$\ddot{\theta}$	Angular acceleration about the z-axis.
α	Angular position of the pendulum normal to the z-axis.
$\dot{\alpha}$	Angular velocity of the pendulum normal to the z-axis.
$\ddot{\alpha}$	Angular acceleration of the pendulum normal to the z-axis.
J_P	Mass moment of inertia of the pendulum.
J_r	Mass moment of inertia of the radial arm.
L_r	Length of the radial arm.
L_p	Length of the pendulum.
m_p	Mass of the pendulum
m_r	Mass of the radial arm.
B_p	Viscous damping coefficient of the pendulum.
B_r	Viscous damping of the radial arm.

Table 2: Parameter definitions relevant to preparing the mathematical model for motor torque.

Symbol	Description	Matlab Variable	Value	Variation
V_{nom}	Motor nominal input voltage		6.0 V	
R_m	Motor armature resistance	Rm	2.6 Ω	$\pm$ 12%
L_m	Motor armature inductance	Lm	0.18 mH	
k_t	Motor current-torque constant	kt	7.68×10^{-3} N m/A	$\pm$ 12%
k_m	Motor back-emf constant	km	7.68×10^{-3} V/(rad/s)	$\pm$ 12%
K_g	High-gear total gear ratio	Kg	70	
	Low-gear total gear ratio	Kg	14	
η_m	Motor efficiency	eta_m	0.69	$\pm$ 5%
η_g	Geabox efficiency	eta_g	0.90	$\pm$ 10%
$J_{m,rotor}$	Rotor moment of inertia	Jm_rotor	3.90×10^{-7} kg $\cdot$ m^2	$\pm$ 10%
J_{tach}	Tachometer moment of inertia	Jtach	7.06×10^{-8} kg $\cdot$ m^2	$\pm$ 10%
J_{eq}	High-gear equivalent moment of inertia without external load	Jeq	9.76×10^{-5} kg $\cdot$ m^2	
	Low-gear equivalent moment of inertia without external load	Jeq	2.08×10^{-5} N $\cdot$ m / (rad/s)	
B_{eq}	High-gear Equivalent viscous damping coefficient	Beq	0.015 N $\cdot$ m / (rad/s)	
	Low-Gear Equivalent viscous damping coefficient	Beq	1.50×10^{-4} kg $\cdot$ m^2	
m_b	Mass of bar load	m_b	0.038 kg	
L_b	Length of bar load	L_b	0.1525 m	
m_d	Mass of disc load	m_d	0.04 kg	
r_d	Radius of disc load	r_d	0.05 m	
m_{max}	Maximum load mass		5 kg	
f_{max}	Maximum input voltage frequency		50 Hz	
I_{max}	Maximum input current		1 A	
ω_{max}	Maximum motor speed		628.3 rad/s	

2. Equation of Motion Equations of motion for the pendulum system are derived using the Lagrange method. Two equations of motion are required for the system, one that describes the motion of the arm and another that describes the motion of the pendulum with respect to motor voltage.

$$\frac{\delta^2 L}{\delta t \delta \dot{q}_i} - \frac{\delta L}{\delta q_i} = Q_i \qquad \text{Eq. 1}$$

In equation one, the variable qi is a generalized coordinate. In this system, its value and the value of its derivative are given by the following arrays.

$$q(t) = [\theta(t) \quad \alpha(t)] \qquad \text{Eq. 2}$$

$$\dot{q}(t) = \left[\frac{d\theta(t)}{dt} \quad \frac{d\alpha(t)}{dt}\right]$$

Substituting the values of the generalized coordinates into equation one gives the Euler-Lagrange equations for the system.

$$\frac{\delta^2 L}{\delta t \delta \dot{\theta}} - \frac{\delta L}{\delta \theta} = Q_1 \qquad \text{Eq. 3}$$

$$\frac{\delta^2 L}{\delta t \delta \dot{\alpha}} - \frac{\delta L}{\delta \alpha} = Q_2$$

The Lagrangian of the system is described by the following equation, where T is the total kinetic energy of the system and V is the total potential energy of the system.

$$Q_1 = \tau - B_r\dot{\theta} \qquad \text{Eq. 4}$$

$$Q_2 = -B_p\dot{\alpha}$$

The generalized forces acting on the rotary arm and the pendulum are torque and viscous damping (friction) and torque. They are given by the equations below.

$$\tau = \frac{\eta_g K_g \eta_m k_t \left(V_m - K_g k_m \dot{\theta}\right)}{R_m} \qquad \text{Eq. 5}$$

The torque of the motor is described by the following equation.

$$\left(m_p {L_r}^2 + \frac{1}{4} m_p {L_p}^2 - \frac{1}{4} m_p {L_p}^2 \cos^2(\alpha) + J_r\right)\ddot{\theta} - \left(\frac{1}{2} m_p L_p L_r \cos(\alpha)\right)\ddot{\alpha} \qquad \text{Eq. 6}$$

$$+ \left(\frac{1}{2} m_p {L_p}^2 \sin(\alpha)\cos(\alpha)\right)\dot{\theta}\dot{\alpha} + \left(\frac{1}{2} m_p L_p L_r \sin(\alpha)\right)\dot{\alpha}^2 = \tau - B_r\dot{\theta}$$

Given that, the non-linear equations of motion are given by the following equations.

$$-\frac{1}{2}m_pL_pL_r\cos(\alpha)\ddot{\theta} + \left(J_p + \frac{1}{4}m_pL_p{}^2\right)\ddot{\alpha} - \frac{1}{4}m_pL_p{}^2\cos(\alpha)\sin(\alpha)\dot{\theta}^2 - \frac{1}{2}m_pL_pgsin(\alpha) = -B_p\dot{\alpha} \quad \text{Eq. 7}$$

The right-hand side of each equation is already linear, so they can be substituted with f(z), where z is an array of the system state variables as follows.

$$z = \left[\theta, \alpha, \dot{\theta}, \dot{\alpha}, \ddot{\theta}, \ddot{\alpha}\right]$$

$$f(z) = \left(m_pL_r{}^2 + \frac{1}{4}m_pL_p{}^2 - \frac{1}{4}m_pL_p{}^2\cos^2(\alpha) + J_r\right)\ddot{\theta} - \left(\frac{1}{2}m_pL_pL_r\cos(\alpha)\right)\ddot{\alpha} + \left(\frac{1}{2}m_pL_p{}^2\sin(\alpha)\cos(\alpha)\right)\dot{\theta}\dot{\alpha} + \left(\frac{1}{2}m_pL_pL_r\sin(\alpha)\right)\dot{\alpha}^2 \quad \text{Eq. 8}$$

All system state variables are set to 0 for linearization. The linearized functions are in the following form.

$$f(z) = -\frac{1}{2}m_pL_pL_r\cos(\alpha)\ddot{\theta} + \left(J_p + \frac{1}{4}m_pL_p{}^2\right)\ddot{\alpha} - \frac{1}{4}m_pL_p{}^2\cos(\alpha)\sin(\alpha)\dot{\theta}^2 - \frac{1}{2}m_pL_pgsin(\alpha) \quad \text{Eq. 9}$$

Solving for the terms of the linearized function for equation ten yields the linearized form of the first equation of motion.

$$f_{lin}(z) = f(z_0) + \left(\frac{\delta f(z)}{\delta\ddot{\theta}}\right)\Bigg|_{z=z_0}\ddot{\theta} + \left(\frac{\delta f(z)}{\delta\ddot{\alpha}}\right)\Bigg|_{z=z_0}\ddot{\alpha} + \left(\frac{\delta f(z)}{\delta\dot{\theta}}\right)\Bigg|_{z=z_0}\dot{\theta} + \left(\frac{\delta f(z)}{\delta\dot{\alpha}}\right)\Bigg|_{z=z_0}\dot{\alpha} + \left(\frac{\delta f(z)}{\delta\theta}\right)\Bigg|_{z=z_0}\theta + \left(\frac{\delta f(z)}{\delta\alpha}\right)\Bigg|_{z=z_0}\ddot{\alpha} + \left(\frac{\delta f(z)}{\delta\dot{\theta}}\right)\Bigg|_{z=z_0}\alpha \quad \text{Eq. 10}$$

Solving for the terms of the linearized function for equation ten yields the linearized form of the first equation of motion.

$$\left(\frac{\delta f(z)}{\delta \ddot{\theta}}\right)\bigg|_{z=z_0} = m_p {L_r}^2 + J_r \qquad \text{Eq. 11}$$

$$\left(\frac{\delta f(z)}{\delta \ddot{\alpha}}\right)\bigg|_{z=z_0} = -\frac{1}{2} m_p L_p L_r$$

$$\left(\frac{\delta f(z)}{\delta \dot{\theta}}\right)\bigg|_{z=z_0} = 0, \left(\frac{\delta f(z)}{\delta \theta}\right)\bigg|_{z=z_0} = 0, \left(\frac{\delta f(z)}{\delta \alpha}\right)\bigg|_{z=z_0} = 0, \left(\frac{\delta f(z)}{\delta \dot{\theta}}\right)\bigg|_{z=z_0} = 0, f(z) = 0$$

$$f_{lin}(z) = \left(m_p {L_r}^2 + J_r\right)\ddot{\theta} - \frac{1}{2} m_p L_p L_r \ddot{\alpha} = \tau - B_r \dot{\theta}$$

This process is repeated for the second equation of motion.

$$\left(\frac{\delta f(z)}{\delta \ddot{\theta}}\right)\bigg|_{z=z_0} = -\frac{1}{2} m_p L_p L_r, \qquad \text{Eq. 12}$$

$$\left(\frac{\delta f(z)}{\delta \ddot{\alpha}}\right)\bigg|_{z=z_0} = J_p + \frac{1}{4} m_p {L_p}^2,$$

$$\left(\frac{\delta f(z)}{\delta \alpha}\right)\bigg|_{z=z_0} = -\frac{1}{2} m_p L_p g$$

$$\left(\frac{\delta f(z)}{\delta \dot{\theta}}\right)\bigg|_{z=z_0} = 0, \left(\frac{\delta f(z)}{\delta \theta}\right)\bigg|_{z=z_0} = 0, \left(\frac{\delta f(z)}{\delta \dot{\theta}}\right)\bigg|_{z=z_0} = 0, f(z) = 0$$

$$f_{lin}(z) = -\frac{1}{2} m_p L_p L_r \ddot{\theta} + \left(J_p + \frac{1}{4} m_p {L_p}^2\right)\ddot{\alpha} - \frac{1}{2} m_p L_p g \alpha = -B_p \dot{\alpha}$$

3. State Space Representation

With the equations of motion derived for the system, the final step in acquiring the mathematical model is to represent the system in state space. Seen below are the general equations for state space representation. Note the equations have been slightly modified to fit the particular scenario of the inertia wheel pendulum.

$$\dot{\theta} = A\theta + Bu \qquad \text{Eq. 13}$$

$$y = C\theta \qquad \text{Eq. 14}$$

To represent the system in state space the phase variables must first be defined. Eqs.(15, 16) below show the system-particular state variable definitions.

$$\dot{\theta} = \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \dot{\theta}_3 \end{bmatrix} \qquad \text{Eq. 15}$$

$$\theta = \begin{bmatrix} \theta_1 \\ \theta_2 \\ \theta_3 \end{bmatrix} \qquad \text{Eq. 16}$$

In order to acquire the equations of motion in phase-variable form, Eqs. (11, 12) are utilized to solve for θ_1 double-dot and θ_2 double-dot. From here, the relationships below are applied.

$$\dot{\theta}_1 = \theta_2 \qquad \text{Eq. 17}$$

$$\dot{\theta}_2 = \ddot{\theta}_1 \qquad \text{Eq. 18}$$

$$\dot{\theta}_3 = \ddot{\theta}_2 \qquad \text{Eq. 19}$$

Applying these relationships yields the vector-matrix form for the state space equation. The results can be seen below.

$$A = \frac{1}{J_T}\begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & \frac{1}{4}m_p{}^2L_p{}^2L_r g & -\left(J_p + \frac{1}{4}m_pL_r{}^2\right)B_r & -\frac{1}{2}m_pL_pL_rB_p \\ 0 & \frac{1}{2}m_pL_pg\left(J_r + m_pL_r{}^2\right) & \frac{1}{2}m_pL_pL_rB_r & -\left(J_r + m_pL_r{}^2\right)B_p \end{bmatrix} \qquad \text{Eq. 20}$$

$$B = \frac{1}{J_T}\begin{bmatrix} 0 \\ 0 \\ J_p + \frac{1}{4}m_pL_p{}^2 \\ \frac{1}{2}m_pL_pL_r \end{bmatrix} \qquad \text{Eq. 21}$$

The final step in aquiring the complete state equation is to define the output equation. Given the context of the inertia wheel pendulum, the output equation is represented below.

$$C = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \qquad \text{Eq. 22}$$

Sensor Calibration

No sensor calibration was necessary for implementation.

Controller Design & Simulation

In order to implement the equation of motion into CopelliaSIM and acheive the desired function, an appropriate controller architechture must be devised. To balance the system in the inverted upright position, the motor must act according to feedback from the encoder that tracks the degree of rotation of the pendulum arm.

The control architecture was found from Gupta's implementation [1] in combination with Quanser manthematics. Architecture that was implemented is shown in the figure below.

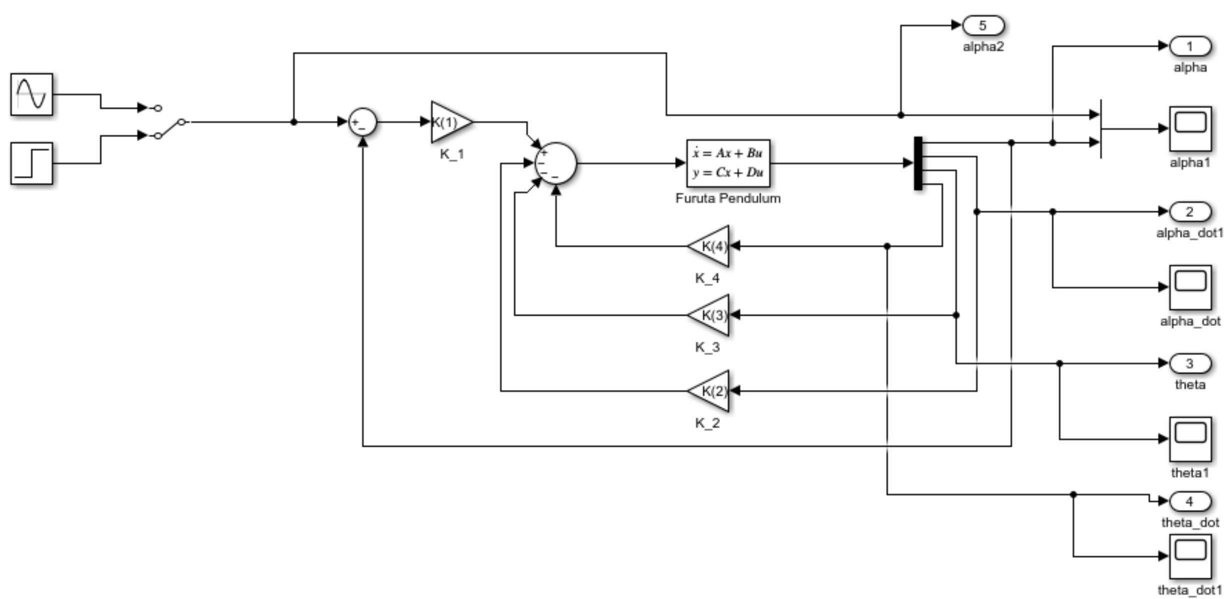

Figure 4: Feedback controller architecture.

The step response associated with the controller in Figure 4 can be seen below.

feedbackResp

Figure 5: Step response of feedback controller.

The pendulum was also modeled in CopelliaSim in the manner shown below. Unfortunately the simulation was not sucessfully implemented.

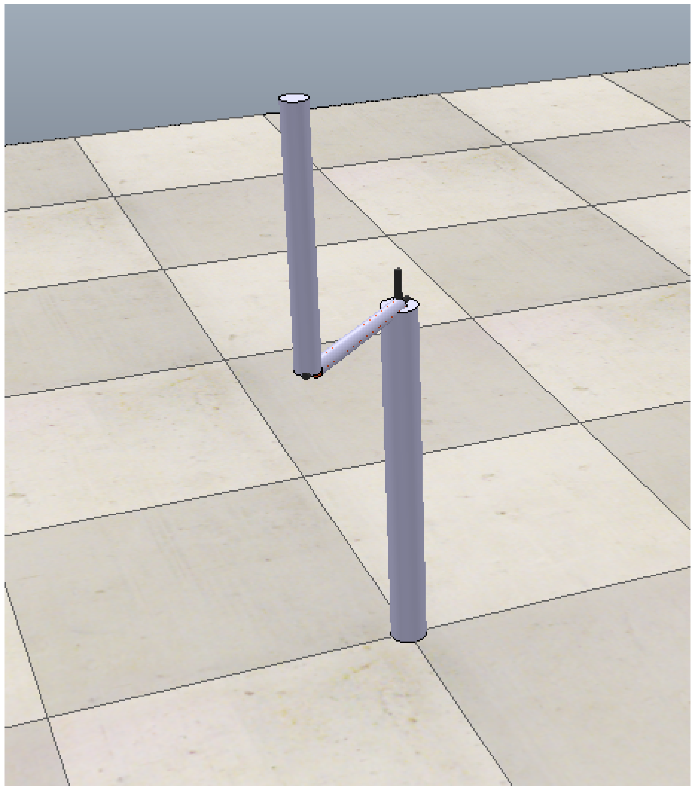

Figure 8: CopelliaSim model forthe Furuta Pendulum.

Appendix A: Simulation Code

▼ MATLAB Full State Feedback

```
%% List of parameters

L_r = 0.36; %length of radial arm
L_p = 0.58; %length of the pendulum
m_p = 0.31; %mass of pendulum (kg)
m_r = 0.53; %mass of the radial arm.
B_p = 0.025; %damping of the pendulum.
B_r = 0.01; %damping of the radial arm.
fre = 0.1; %the frequency of the sin wave input

J_r=((m_r+m_p)*L_r^2)/3;
J_p=(m_p*L_p^2)/3;
J_T=J_p*m_p*L_r^2+J_r*J_p+0.25*J_r*m_p*L_p^2;
g=9.81; % gravity

%% Matrices
```

```
A=[0 0 1 0;
   0 0 0 1;
   0 0.25*m_p^2*L_p^2*L_r*g -(J_p+0.25*m_p*L_p^2)*B_r -0.5*m_p*L_p*L_r*B_p;
   0 0.5*m_p*L_p*g*(J_r+m_p*L_r^2) 0.5*m_p*L_p*L_r*B_r -(J_r+m_p*L_r^2)*B_p];

B=1/J_T*[0;0;J_p+0.25*m_p*L_p^2;0.5*m_p*L_p*L_r];

C=[1 0 0 0;
    0 1 0 0];
D=[0;0;0;0];

P=[-17.1 8.34 -2.87 0];

K=acker(A,B,P);

%tout=0:0.1:30;
subplot(4,1,1)

plot(tout,yout(:,1)*180/pi);grid on;
xlabel('Time');ylabel('\alpha');

subplot(4,1,2)

plot(tout,yout(:,2)*180/pi);grid on;
xlabel('Time');ylabel('\dot{\alpha}');

subplot(4,1,3)

plot(tout,yout(:,3)*180/pi);grid on;
xlabel('Time');ylabel('\theta');

subplot(4,1,4)

plot(tout,yout(:,4)*180/pi);grid on;
xlabel('Time');ylabel('\dot{\theta}');
```

▼ MATLAB API

```
wheel=remApi('remoteApi'); % using the prototype file (remoteApiProto.m)
wheel.simxFinish(-1); % just in case, close all opened connections
clientID=wheel.simxStart('127.0.0.1',19999,true,true,5000,5);
w = 30;
 if (clientID>-1)
        disp('Connected to remote API server');
            %output
```

```
[returnCode,motor_encoder]=wheel.simxGetObjectHandle(clientID,'motor_encoder',wheel.

            %input

[returnCode,encoder1]=wheel.simxGetObjectHandle(clientID,'encoder',wheel.simx_opmode



[returnCode,encoder]=wheel.simxGetJointPosition(clientID,encoder1,wheel.simx_opmode_

%
[returnCode,encoder]=wheel.simxGetIntegerParameter(clientID,encoder,wheel.simx_opmod

        %Execute This
        %Moves forward

        while (1)

[returnCode,encodernum]=wheel.simxGetJointPosition(clientID,encoder1,wheel.simx_opmc

  %
[returnCode,encoder]=wheel.simxGetIntegerParameter(clientID,encoder,wheel.simx_opmod

 encoderout = (encodernum * (180 / 3.14))

%          data = '{}\n'.format(encoderout)

[returnCode]=wheel.simxSetJointTargetVelocity(clientID, motor_encoder, w
,wheel.simx_opmode_blocking);
             pause(0.5)
        end
        wheel.simxFinish(-1);
 end
 wheel.delete(); % call the destructor!
```

Other simulation files can be found here.

Appendix B: Project Documents/Video

- Presentation Slides
- Presentation Video
- GitHub .pdf version

References

[1] Vikash Gupta (2019). Full State Feedback of Furuta Pendulum (https://www.mathworks.com/matlabcentral/fileexchange/25585-full-state-feedback-of-furuta-pendulum), MATLAB Central File Exchange. Retrieved May 19, 2022.

[2] Jacob Apkarian, Michel Lévis and Hakan Gurocak, Inverted Pendulum Experiment, Student Workbook. Ontario, Canada: Quanser, 2011.

[3] Jacob Apkarian, Michel Lévis and Hakan Gurocak, SRV02 Rotary Servo Base Unit User Manual. Ontario, Canada: Quanser, 2011.

Releases

No releases published
Create a new release

Packages

No packages published
Publish your first package

Contributors 3

kapenso1

t-ches-csuc

QuincyatChico

Languages

- MATLAB 100.0%