# Furuta Pendulum: Rotating Cat Toy
# Control System Design


By

Jarrel Cook

Michelle Gomez

Miguel Rodriguez

Noah McCracken

Ryan Calbert


Fall 2021 – MECA482

Group 1


California State University, Chico
Department of Mechanical and Mechatronic Engineering and Advanced Manufacturing
Chico, Ca 95929


December 17, 2021

**Table of Contents**

## I. Introduction

In 2020 people worldwide were forced to remain inside, threatened by a global pandemic. With much more time on their hands, many turned to adapt pets to keep them company and busy. As a result of many new pet owners, pet toys and accessories are increasingly in demand. This page will review a prototype of an interactive cat toy. The toy's main component will be an inverted rotary pendulum, also known as a Furuta Pendulum. The pendulum will react to a displacement caused by the animal, which allows for minimal human interaction. Along with the pendulum, the toy will have shielding to protect the animal and a textured ball to stimulate the cat's natural prey drive.

A Furuta Pendulum consists of three main components, an electric DC motor and an arm for which the third component, the pendulum, is connected. As the system is initialized, the DC reacts to the pendulum's position and produces torque. The torque creates rotation of the arm in a horizontal plane. The pendulum is connected to the arm but can rotate freely in a vertical plane, with the arm as its focal point. The Furuta Pendulum is able to rotate and balance the arm in an upright position. The vertical arm is able to correct itself if a small displacement is applied to it, such as a cat hitting it. This allows for the furuta pendulum to be a great interactive cat toy.

The capabilities database can be seen in the figure below. It describes the requirements of the Furuta Pendulum, such as the feedback, rotation, and programming.

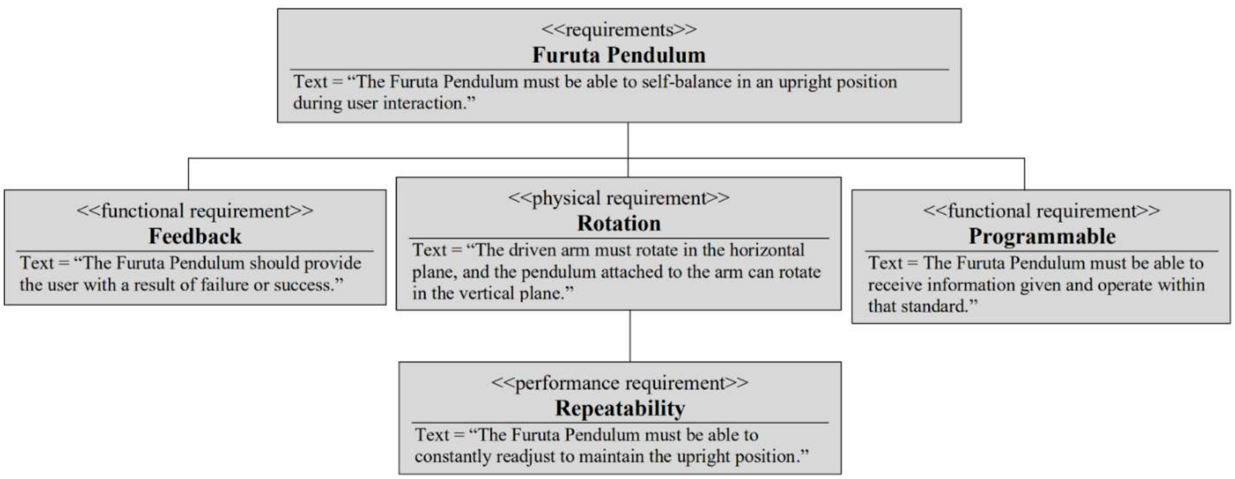


**Figure 1.** Capabilities Database diagram.

Logical/Functional Viewpoint Description

To design the system for the interactive cat toy, there must be a logical/functional viewpoint. The functional viewpoint addresses the analysis of abstract functional elements and their logical interactions. As seen in Figure 2, there is a motor. Once the motor becomes activated by certain displacement, it rotates a component. The component then allows the pendulum to rotate and there is a signal communicated to the sensor. Once the sensor is communicated the signal, it then sends out a measurement to the controller. The controller then exchanges inputs with the human machine interface and then sends a control signal to the driver.

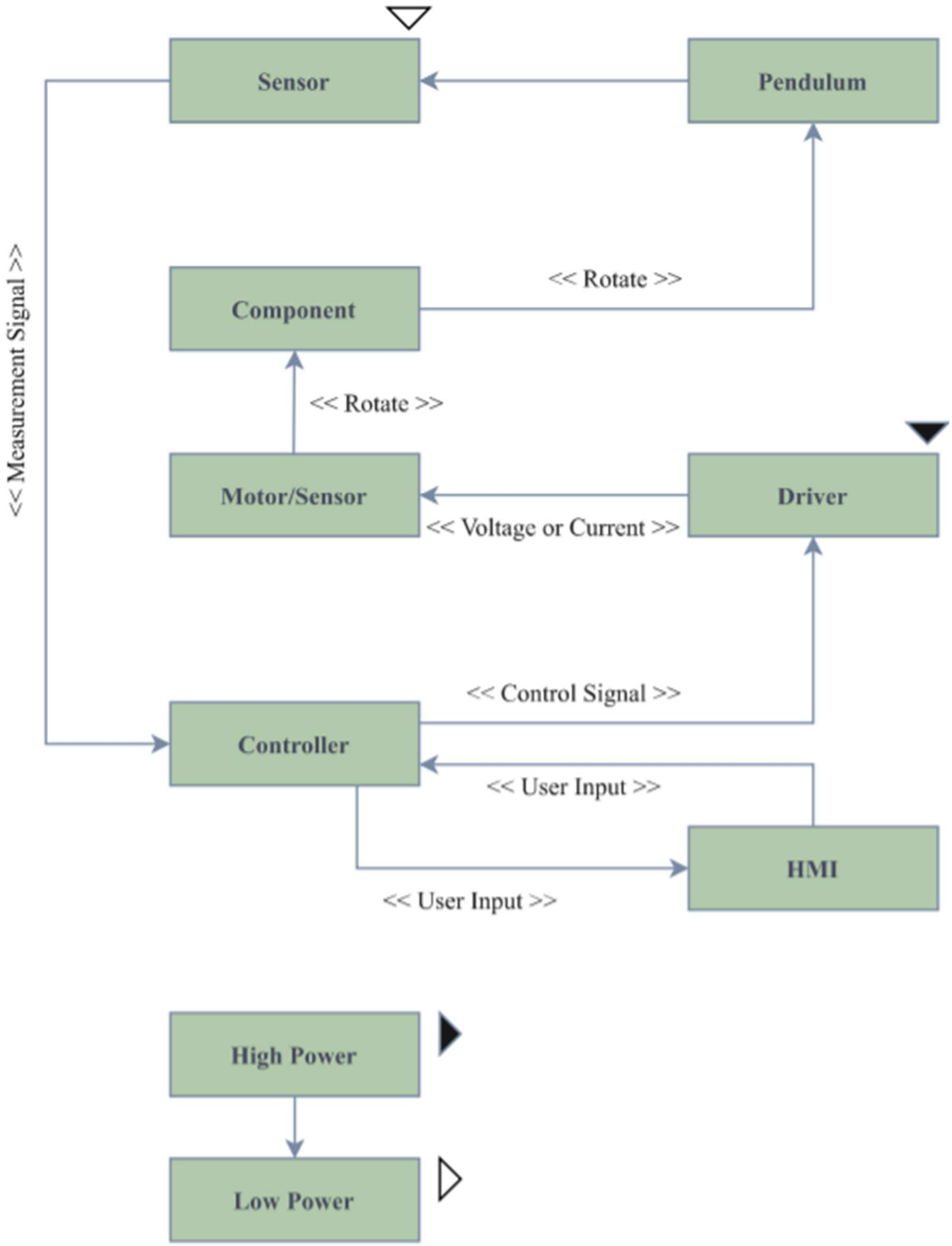


**Figure 2.** Logical/Functional viewpoint of furuta pendulum system.

Operational Viewpoint Description

The operational viewpoint of the interactive cat toy is displayed below in Figure 3. The operational viewpoint descries the tasks, activities, as well as the operational elements. It is the architectural build of the product.

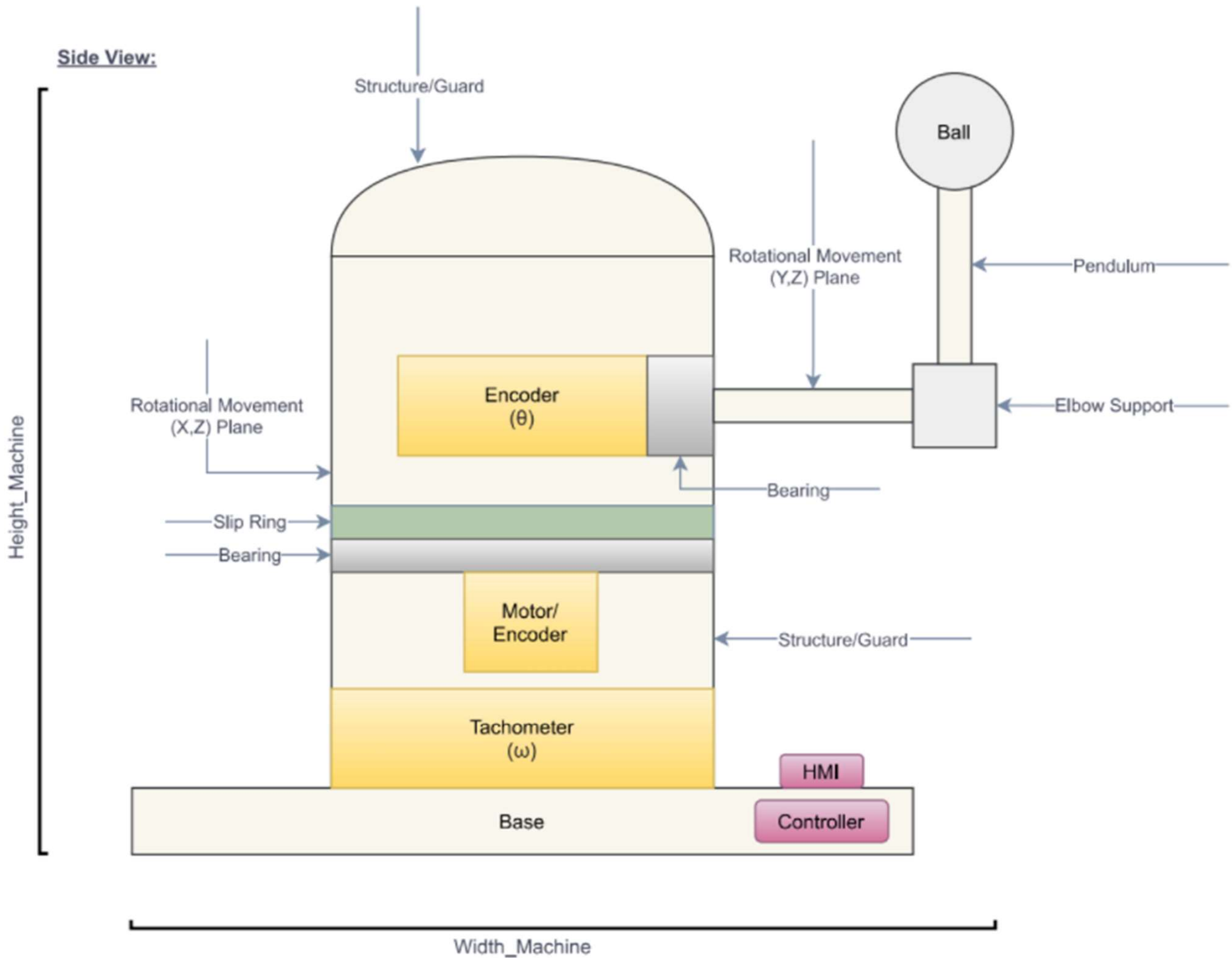


**Figure 3.** Operational viewpoint of furuta pendulum system.

## II. Modeling

A general representation of the Furuta Pendulum is shown below.

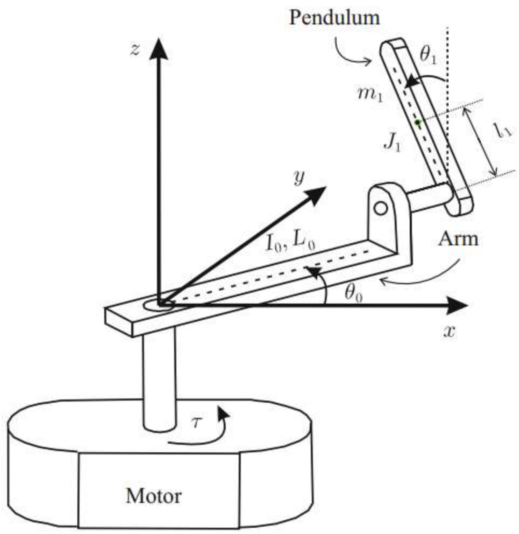


**Figure 4.** General Furuta Pendulum diagram model.

The main parameters seen in Figure 4 are individually listed below:

- $\theta_0$ : Angular position of the arm in radians
- $\theta_1$ : Angular position of the pendulum with respect to the pendulum in the downward position, in radians
- $\tau$ : Torque coming from the electric motor
- $I_0$ : Moment of inertia of the arm plus the motor inertia
- $L_o$ : Length of the arm
- $m_1$ : Pendulum mass
- $l_1$ : Location of the center mass of the Pendulum
- $J_1$ : Moment of inertia

From Hernández-Guzmán 2019, the Euler–Lagrange equations for the Furuta Pendulum are as follows.

$$\frac{d}{dt}\frac{\partial L}{\partial \dot{\theta}_0} - \frac{\partial L}{\partial \theta_0} = \tau,$$
$$\frac{d}{dt}\frac{\partial L}{\partial \dot{\theta}_1} - \frac{\partial L}{\partial \theta_1} = 0.$$

The Lagrangian is defined as:

$$L = K - P,$$
$$K = K_0 + K_1,$$

Where,

- $K_0$ = Arm's kinetic energy
- $K_1$ = Pendulum's kinetic energy
- P = Pendulum's potential energy

$$K_0 = \frac{1}{2} I_0 \dot{\theta}_0^2.$$

$$K_1 = \frac{1}{2} J_1 \dot{\theta}_1^2 + \frac{1}{2} m_1 v_1^T v_1$$

Velocity going through the pendulum center of mass, *v*, is represented by the matrix below, and X represents the coordinates of the center of mass.

$$v_1 = \begin{bmatrix} \frac{dX_x}{dt} \\ \frac{dX_y}{dt} \\ \frac{dX_z}{dt} \end{bmatrix}$$

$$X = \begin{bmatrix} X_x \\ X_y \\ X_z \end{bmatrix} = \begin{bmatrix} L_0 \cos(\theta_0) - l_1 \sin(\theta_1) \sin(\theta_0) \\ L_0 \sin(\theta_0) + l_1 \sin(\theta_1) \cos(\theta_0) \\ l_1 \cos(\theta_1) \end{bmatrix}$$

The next two diagrams represent the dynamic parameters of the system.

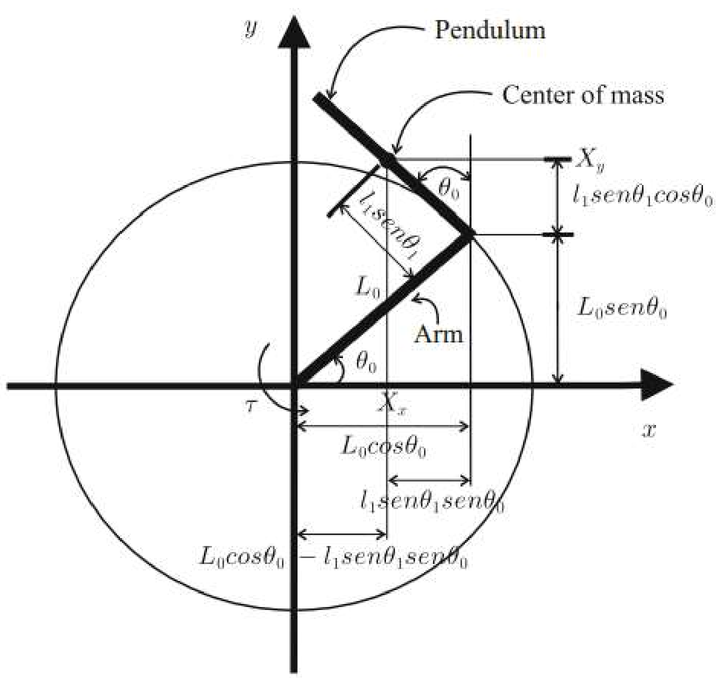


**Figure 5.** Superior view of geometric relationships in the Furuta Pendulum.

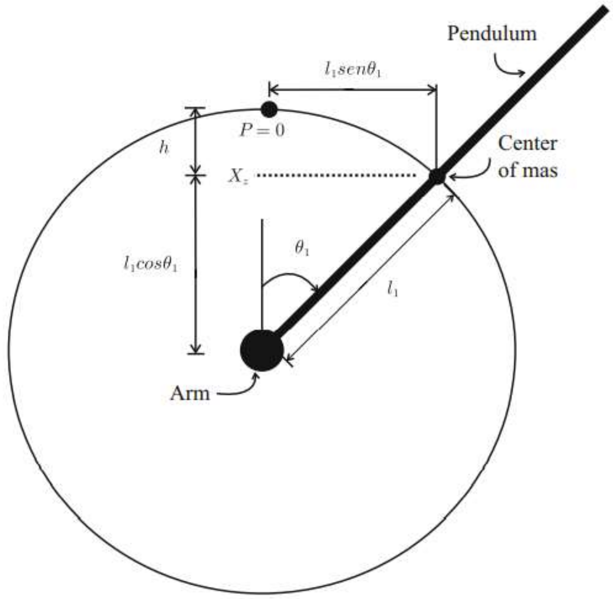


**Figure 6.** View of the plane orthogonal to arm.

$$K_1 = \frac{1}{2}J_1\dot{\theta}_1^2 + \frac{1}{2}m_1\left[(L_0\dot{\theta}_0)^2 + (l_1\dot{\theta}_0)^2\sin^2(\theta_1) + (l_1\dot{\theta}_1)^2 + 2\dot{\theta}_0\dot{\theta}_1 L_0 l_1\cos(\theta_1)\right]$$

$$P = m_1 g l_1(\cos(\theta_1) - 1)$$

The following equations represent the Furuta pendulum mathematical model in a nonlinear form.

$$\begin{aligned}
&M(q)\ddot{q} + C(q,\dot{q})\dot{q} + g(q) = F,\\
M(q) &= \begin{bmatrix} I_0 + m_1(L_0^2 + l_1^2\sin^2(\theta_1)) & m_1 l_1 L_0\cos(\theta_1) \\ m_1 l_1 L_0\cos(\theta_1) & J_1 + m_1 l_1^2 \end{bmatrix},\\
C(q,\dot{q}) &= \begin{bmatrix} \frac{1}{2}m_1 l_1^2\dot{\theta}_1\sin(2\theta_1) & -m_1 l_1 L_0\dot{\theta}_1\sin(\theta_1) + \frac{1}{2}m_1 l_1^2\dot{\theta}_0\sin(2\theta_1) \\ -\frac{1}{2}m_1 l_1^2\dot{\theta}_0\sin(2\theta_1) & 0 \end{bmatrix},\\
g(q) &= \begin{bmatrix} 0 \\ -m_1 l_1 g\sin(\theta_1) \end{bmatrix}, \quad F = \begin{bmatrix} \tau \\ 0 \end{bmatrix}, \quad q = \begin{bmatrix} \theta_0 \\ \theta_1 \end{bmatrix}.
\end{aligned}$$

To control the swing up of the pendulum is performed when the pendulum is in its bottom stable position and inverted into a unstable position. The following equations are used to perform this process.

$$\tau = \frac{-\frac{k_\omega}{\det(M(q))}F(q,\dot{q}) - k_\theta\theta_0 - k_\delta\dot{\theta}_0}{k_E E(q,\dot{q}) + \frac{k_\omega}{\det(M(q))}(J_1 + m_1 l_1^2)},$$

$$E(q,\dot{q}) = \frac{1}{2}\dot{q}^T M(q)\dot{q} + m_1 g l_1(\cos\theta_1 - 1),$$

$$F(q,\dot{q}) = -(J_1 + m_1 l_1^2)m_1 l_1^2\dot{\theta}_1\dot{\theta}_0\sin(2\theta_1) - \frac{1}{2}m_1^2 l_1^3 L_0\dot{\theta}_0^2\cos\theta_1\sin(2\theta_1)$$
$$-m_1^2 l_1^2 L_0 g\cos\theta_1\sin\theta_1 + (J_1 + m_1 l_1^2)m_1 l_1 L_0\dot{\theta}_1^2\sin\theta_1,$$

Where $E$ is the pendulum's total energy and the $k$-values are positive constants that satisfy the following:

$$\frac{k_\omega}{k_E} > 2m_1 g l_1(I_0 + m_1 l_1^2 + m_1 L_0^2)$$

If the total energy of the system is equal to zero then,

$$\dot{\theta}_1 = \pm\sqrt{\frac{2m_1 g l_1}{J_1 + m_1 l_1^2}(1 - \cos\theta_1)}.$$

$$V(q,\dot{q}) = \frac{k_E}{2}E^2(q,\dot{q}) + \frac{k_\omega}{2}\dot{\theta}_0^2 + \frac{k_\theta}{2}\theta_0^2$$

<u>Linear System Model</u>

The state vector that can be chosen for the system is as follows.

$$x = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} \theta_0 \\ \dot{\theta}_0 \\ \theta_1 \\ \dot{\theta}_1 \end{bmatrix}$$

Using the previous equations and solving,

$$\ddot{q} = \begin{bmatrix} w_1(x_1, x_2, x_3, x_4, \tau) \\ w_2(x_1, x_2, x_3, x_4, \tau) \end{bmatrix},$$
$$= M^{-1}(x_1, x_3)[-C(x_1, x_2, x_3, x_4)[x_2, x_4]^T - g(x_1, x_3) + F]$$

If *M(q)* is nonsingular and the corresponding equation is as follows.

$$\dot{x} = f(x, u),$$
$$f(x, u) = \begin{bmatrix} f_1(x, u) \\ f_2(x, u) \\ f_3(x, u) \\ f_4(x, u) \end{bmatrix} = \begin{bmatrix} x_2 \\ w_1(x_1, x_2, x_3, x_4, u) \\ x_4 \\ w_2(x_1, x_2, x_3, x_4, u) \end{bmatrix}, \quad u = \tau.$$

The approximate linear model is now obtained that is valid around an operation point. The operation points are pairs *(x*,u*)*.

$$f(x^*, u^*) = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

$$x_2^* = \dot{\theta}_0^* = 0, \quad x_4^* = \dot{\theta}_1^* = 0,$$
$$\begin{bmatrix} w_1(x_1^*, 0, x_3^*, 0, u^*) \\ w_2(x_1^*, 0, x_3^*, 0, u^*) \end{bmatrix} = M^{-1}(x_1^*, x_3^*)$$
$$\times \left\{ -C(x_1^*, 0, x_3^*, 0) \begin{bmatrix} 0 \\ 0 \end{bmatrix} - g(x_1^*, x_3^*) + \begin{bmatrix} u^* \\ 0 \end{bmatrix} \right\}$$
$$= \begin{bmatrix} 0 \\ 0 \end{bmatrix}.$$

$$\begin{bmatrix} u^* \\ 0 \end{bmatrix} = g(x_1^*, x_3^*) = \begin{bmatrix} 0 \\ -m_1 l_1 g \sin(\theta_1^*) \end{bmatrix},$$

$$u^* = 0, \quad x_3^* = \theta_1^* = \pm n\pi,$$

$$x^* = \begin{bmatrix} x_1^* \\ x_2^* \\ x_3^* \\ x_4^* \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \quad u^* = 0$$

$$\dot{z} = Az + Bv,$$

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & \dfrac{-gm_1^2 l_1^2 L_0}{I_0(J_1 + m_1 l_1^2) + J_1 m_1 L_0^2} & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & \dfrac{(I_0 + m_1 L_0^2) m_1 l_1 g}{I_0(J_1 + m_1 l_1^2) + J_1 m_1 L_0^2} & 0 \end{bmatrix}, \; B = \begin{bmatrix} 0 \\ \dfrac{J_1 + m_1 l_1^2}{I_0(J_1 + m_1 l_1^2) + J_1 m_1 L_0^2} \\ 0 \\ \dfrac{-m_1 l_1 L_0}{I_0(J_1 + m_1 l_1^2) + J_1 m_1 L_0^2} \end{bmatrix}$$

## III. Controller Design and Simulation

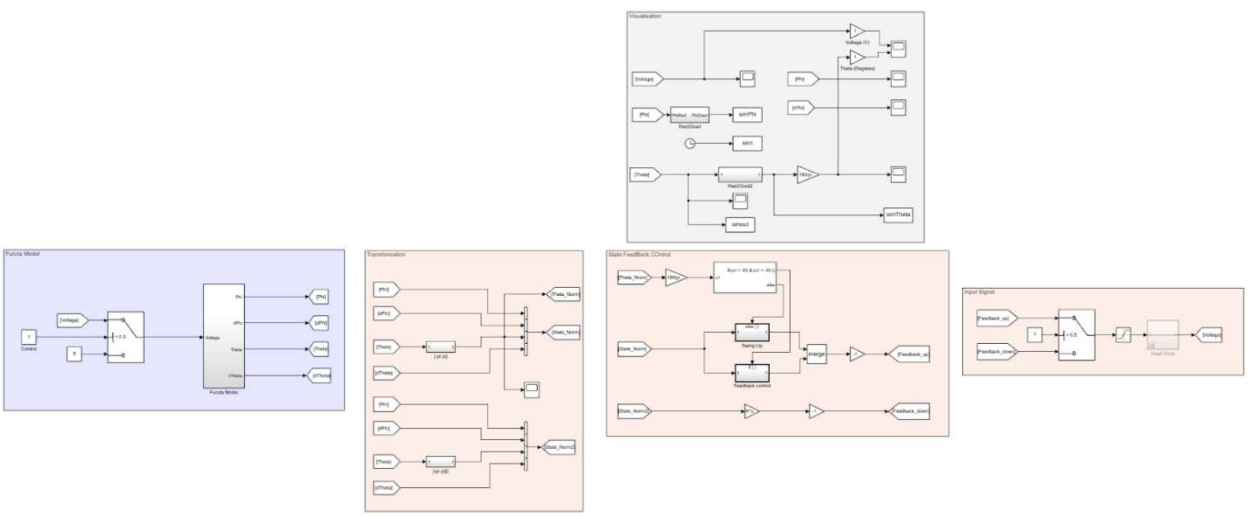

**Figure 7.** Simulink Control System of the Furuta Pendulum

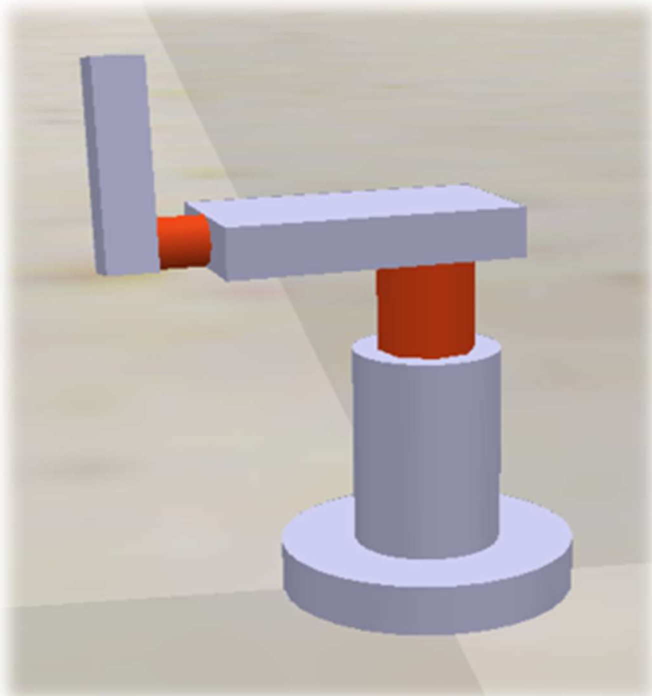

**Figure 8.** CoppeliaSim Furuta Pendulum system.

## IV. Appendix A: MATLAB Code

Matlab Code

The following code is the main code for the system.

```
g = 9.81;
% 1 - Arm
% 2 - Pendulum

m1 = 0.380;
m2 = 0.054;
L1 = 0.066;
L2 = 0.146;
M = 0.044;

J = 3.5256e-4;
kb_p = 4.7940e-04 ;

kb_m = 6.75e-4 ;
ke = 0.5;
Re = 14.5;  %Medido

alpha = J + (M+m1/3+m2)*L1^2;
beta = (M + m2/3)*L2^2;
gamma = (M + m2/2)*L2*L1;
sigma = (M + m2/2)*g*L2;

%% Simulation parameters

initial_state = pi;
Ts = 0.001;
dtDisc = 0.01;
Reference = [0 0 0 0];
Zn = 3;   %Dead Zone

StepX = 10;
distrub = 12;
disturb = distrub*pi/180;

%% Linearization

% A matrix
A = zeros(4,4);
A(1,2) = 1;
A(2,3) = -(sigma*gamma)/(alpha*beta-gamma^2);
A(3,4) = 1;
A(4,3) = (alpha*sigma)/(alpha*beta-gamma^2);

% B matrix
B = zeros(4,2);
B(2,1) = beta/(alpha*beta-gamma^2);
B(2,2) = -gamma/(alpha*beta-gamma^2);
B(4,1) = -gamma/(alpha*beta-gamma^2);
B(4,2) = alpha/(alpha*beta-gamma^2);

% C matrix
C = [0 0 1 0;
     0 0 0 1];

%% Pseudo Linear System
```

```
% Pseudo A matrix
Ap = zeros(4,4);

Ap(1,2) = 1;

Ap(2,1) = 0;      Ap(2,2) = -B(2,1)*(ke^2/Re + kb_m);
Ap(2,3) = A(2,3); Ap(2,4) = -B(2,2)*kb_p;

Ap(3,4) = 1;

Ap(4,1) = 0;      Ap(4,2) = -B(4,1)*(ke^2/Re + kb_m);
Ap(4,3) = A(4,3); Ap(4,4) = -B(4,2)*kb_p;

% Pseudo B matrix

Bp = zeros(4,1);

Bp(2) = B(2,1)*ke/Re;
Bp(4) = B(4,1)*ke/Re;

% Controlability and Observability
Control = rank(ctrb(Ap,Bp));
Observ = rank(obsv(Ap,C));

% State Feedback Control

%K = place(Ap,Bp,[-5 -4 -2+2j -2-2j]);
%K = place(Ap,Bp,[-8+8j -8-8j  -4+4j -4-4j]);

Q = [0.1 0 0 0;
     0 0.01 0 0;
     0 0 100 0;
     0 0 0 10];
R = 10;
[K, ~, E] = lqr(Ap,Bp,Q,R);

%K -0,4473 -1,3203 -33,3556 -3,0467
%K -0.2000   -1.1342  -30.9420   -3.2791  - Funciona pendulo 16/01

%% Down position (pi)

% Pseudo A matrix around pi
Ap2 = zeros(4,4);

Ap2(1,2) = 1;

Ap2(2,1) = 0;      Ap2(2,2) = -B(2,1)*(ke^2/Re + kb_m);
Ap2(2,3) = A(2,3); Ap2(2,4) = B(2,2)*kb_p;

Ap2(3,4) = 1;

Ap2(4,1) = 0;      Ap2(4,2) = B(4,1)*(ke^2/Re + kb_m);
Ap2(4,3) = -A(4,3); Ap2(4,4) = -B(4,2)*kb_p;

% Pseudo B matrix around pi

Bp2 = zeros(4,1);

Bp2(2) = B(2,1)*ke/Re;
Bp2(4) = -B(4,1)*ke/Re;
```

```
K2 = place(Ap2,Bp2,[-5 -4 -2+2j -2-2j]);
R2 = 1;
Q2=[1 0 0 0;
    0 10 0 0;
    0 0 1000 0;
    0 0 0 10];
[K2, ~, E] = lqr(Ap2,Bp2,Q2,R2);
```

The following code is for the Furuta Pendulum’s visualization and it includes comments to easier maneuver and describe through the code.

```
view(135,20)                        %Starting view
AL = 5;                             %Define graph axis limits
axis([-AL AL -AL AL -AL AL]);
grid on

L1=3;  %Rotary arm length
L2=2;  %Pendulum length

Xh=[0 ; L1]';
Yh=[0 ; 0]';
Zh=[0 ; 0]';

Xv=[Xh(2) ; L1]';
Yv=[Yh(2) ; 0]';
Zv=[Zh(2) ; -L2]';

hold on
Harm  = fill3(Xh,Yh,Zh,'b');
Varm  = fill3(Xv,Yv,Zv,'g');

s=8;
M=scatter3(Xv(2),Yv(2),Zv(2),s,'filled','MarkerFaceColor','b','MarkerEdgeColor','k');

theta=0;
phi=0;
c = [0 0 0];

%save video
v = VideoWriter('simulation');
open(v);

TXT=title('Time: ');
for t=1:10:size(simTheta,1)
    TXT2=sprintf('Time:%.2f',simt(t));
    set(TXT,'String',TXT2);

    phi=simPhi(t);
    theta =-simTheta(t);

    Xh(2)= L1*cos(phi);
    Yh(2)=L1*sin(phi);

    Xva = 0;
    Yva = L2*sin(theta);
    Zva = -L2*cos(theta);

    Xvb = Xva*cos(phi)-Yva*sin(phi)+L1*cos(phi);
```

```
    Yvb = Xva*sin(phi)+Yva*cos(phi) + L1*sin(phi);
    Zvb = Zva;

    Xv=[Xh(2);Xvb]';
    Yv=[Yh(2);Yvb]';
    Zv=[ 0   ;Zvb]';

    set(Harm,'XData',Xh);
    set(Harm,'YData',Yh);
    set(Harm,'ZData',Zh);

    set(Varm,'XData',Xv);
    set(Varm,'YData',Yv);
    set(Varm,'ZData',Zv);

    rem(t,30)

    %trace the end of the pendulum
    %scatter3(Xv(2),Yv(2),Zv(2),s,'filled','MarkerFaceColor',c,'MarkerEdgeColor','k');

    set(M,'XData',Xv(2));
    set(M,'YData',Yv(2));
    set(M,'ZData',Zv(2));

    %save video
    frame = getframe(gcf);
    writeVideo(v, frame);
    if(simt(t) > 5) %break after given time
        break;
    end
end
close(v);
```

## V. Appendix B: References

Nise, N. S. (2011). *Control Systems Engineering*. John Wiley & Sons.

Hernández-Guzmán Victor Manuel, & Silva-Ortigoza Ramón. (2019). *Automatic control with experiments*. Springer.