EE615- Controls and Computation Lab

Experiment 1 :
Rotary Inverted Pendulum

Submitted by

Sayali Ravindra Duragkar
203070025
Abhishek Verma
213070024

Under the guidance of
Prof. Debraj Chakraborty

Department of Electrical Engineering
INDIAN INSTITUTE OF TECHNOLOGY BOMBAY

Contents

1 Objective

The objective of this experiment is to design two controllers for the given simulink model of inverted pendulum. Swing-up controller will be used to control the swing of the pendulum so as to properly swing it to up-right direction and balance controller(here we are using lqr controller) to stabilize the pendulum in inverted direction i.e. in the unstable position with 180° pendulum arm angle.

2 Model Description

The Inverted pendulum model provided moves in two rotary directions and have two degree of freedom. The rotating arm angle is denoted by the symbol α while the pendulum attached to the arm rotates about its pivot and its angle is called β.The input voltage of the motor is the control variable.

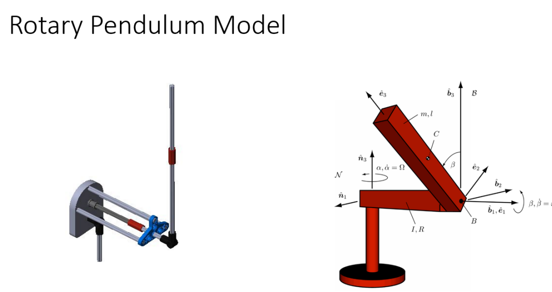


Figure 1: Given Pendulum Model

The Mathematical equations of the model derived are:-

$$\frac{d^2\alpha(t)}{dt^2} = \frac{(-Mp^2glp^2rcos(\alpha(t))(\beta(t))}{(Mpr^2sin(\alpha(t))^2-Jeq-Mpr^2)Jp-Mplp^2Jeq}$$

$$= \frac{-JpMp(r^2)cos\alpha(t)(\frac{d\alpha(t)}{dt})^2}{(Mpr^2sin\alpha(t)^2-Jeq-Mpr^2)Jp-Mplp^2Jeq}$$

$$= \frac{-JpTout+Mplp^2Tout}{(Mpr^2(sin\alpha(t))^2-Jeq-Mpr^2)Jp-Mplp^2Jeq}$$

$$\frac{d^2\beta(t)}{dt^2} = \frac{-lpMp(-Jeqg+Mpr^2sin\alpha(t)^2g-Mpr^2g)\beta(t)}{(Mpr^2sin\alpha(t)^2-Jeq-Mpr^2)Jp-Mplp^2Jeq}$$

$$= \frac{-lpMprsin\alpha(t)Jeq(\frac{d\alpha(t)}{dt})^2}{(Mpr^2sin\alpha(t)^2-Jeq-Mpr^2)Jp-Mplp^2Jeq}$$

$$\frac{lpMprToutcos\alpha(t)}{(Mpr^2sin\alpha(t)^2-Jeq-Mpr^2)Jp-Mplp^2Jeq}$$

(1)

After linearizing the model , we will get state space equation as -

$x = [x1, x2, x3, x4]^T$,

$x1 = \alpha, x2 = \beta, x3 = \frac{d\alpha}{dt}, x4 = \frac{d\beta}{dt}$

And the state space Model is :

$\frac{d(x(t))}{dt} = Ax(t) + BU(x)$

$y(t) = Cx(t) + DU(x)$

Where U(x)=Vm and after putting the values given to us the A, B , C, D matrices result as follows:-

Matrix A is Given By:-

$$A = \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & \frac{rMp^2lp^2g}{JpJeq+Mplp^2Jeq+JpMpr^2} & \frac{-KtKm(Jp+Mplp^2)}{(JpJeq+Mplp^2Jeq+JpMpr^2)Rm} & 0 \\ 0 & \frac{-Mplpg(Jeq+Mpr^2)}{JpJeq+Mplp^2Jeq+JpMpr^2} & \frac{MplpKtrKm}{(JpJeq+Mplp^2Jeq+JpMpr^2)Rm} & 0 \end{pmatrix}$$

Matrix B is given By:-

$$B = \begin{pmatrix} 0 \\ 0 \\ \frac{Kt(Jp+Mplp^2}{(JpJeq+Mplp^2Jeq+JpMpr^2)Rm} \\ \frac{-MplpKtKr}{(JpJeq+Mplp^2Jeq+JpMpr^2)Rm} \end{pmatrix}$$

Matrix C is Given By:-

$$C = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Matrix D is Given By:-

$$D = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

3 Controller Design

3.1 Swing-up Controller

In this section we have used Lagrange method to find the nonlinear equations of motion of the pendulum. Thus the kinematics, potential energy, and kinetic energy are first calculated and the equations of motion are found using Euler-Lagrange.

The Pendulum Lagrange system is given by:-

$L(\beta(t), \frac{d\beta(t)}{dt}) = T - U$

Where T and U are the Kinetic and Potential Energy of the System.

The controller designed attempts to minimize an expression that is a function of the system's total energy. In order to rotate the pendulum into its upwards vertical position, the total energy of the pendulum and its dynamics must be redefined in terms of the angle:

$\beta(up) = \beta - \pi$

Total Energy Expressed in terms of β is :-

E$(\beta(up)) = L(\beta(up) + \pi)$

which is calculated to be:-

E = k$(\cos(\beta(t)) - 1)$

Where k is mgl

Now,

The final Control equation that is derived and implemented is :-

$U = sat[k(cos\beta - 1)sgn[(\dot{\beta})cos\beta]$

where sat() is the saturation function and sgn() is the signum function makes for a control with the largest variance and overall tends to perform very well.

3.2 LQR Controller

Parameter Values:-
$g = 9.81;$
$Jeq = 1.23 * 10^{(} - 4);$
$Jp = 1.1 * 10^{(} - 4);$
$Rm = 3.30;$
$Kt = 0.02797;$
$Km = 0.02797;$
$r = 0.159;$
$Mp = 3.3 * 10^{(} - 3);$
$lp = 0.153;$

Symbol	*Description*
M_p	Mass of the pendulum assembly (weight and link combined).
l_p	Length of pendulum center of mass from pivot.
L_p	Total length of pendulum.
r	Length of arm pivot to pendulum pivot.
J_m	Motor shaft moment of inertia.
M_{arm}	Mass of arm.
g	Gravitational acceleration constant.
J_{eq}	Equivalent moment of inertia about motor shaft pivot axis.
J_p	Pendulum moment of inertia about its pivot axis.
B_{eq}	Arm viscous damping.
B_p	Pendulum viscous damping.
R_m	Motor armature resistance.
K_t	Motor torque constant.
K_m	Motor back-electromotive force constant.

The balance controller is used to balance the pendulum arm in the angular threshold of $30 * \pi/180 = 0.523$. The amount of voltage supplied depends on the angular position and speed of both the arm and the pendulum.
Given the state space eqaution is :-

$\frac{d(x(t)}{dt} = Ax(t) + Bu(t)$
The cost Function that is to be minimized is:-

J = $\int_0^\infty x(t)^T Qx(t) + u(t)TRu(t)\,dt$

From the given parameter values matrix A and B are calculated as:-

$$A = \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 12.3454 & -1.3782 & 0 \\ 0 & -31.7445 & 0.5909 & 0 \end{pmatrix}$$

and

$$B = \begin{pmatrix} 0 \\ 0 \\ 162.6074 \\ -21.1255 \end{pmatrix}$$

where A is 4*4 and B is 4*1 matrix

Now Q is 4*4 matrix and R is 1*1 matrix and the value calculated are:-

$$Q = \begin{pmatrix} 0.0015 & 0 & 0 & 0 \\ 0 & 680 & 0 & 0 \\ 0 & 0 & 0.002 & 0 \\ 0 & 0 & 0 & 0.12 \end{pmatrix}$$

R = 1;

Now , Putting these all values of A , B ,C ,D , Q and R in the LQR command in the Matlab , We get K values such that , U=-K(x).
The K is the matrix consisting of K1 , K2 , K3 , K4.

The K matrix obtained was:

K= [0.03 25 0.02 1.15]
where -
$k1 = 0.03;$
$k2 = 25;$
$k3 = 0.01;$
$k4 = 1.15$

The sum of these k values are given to the Balance Controller.
The figure of matlab code:-

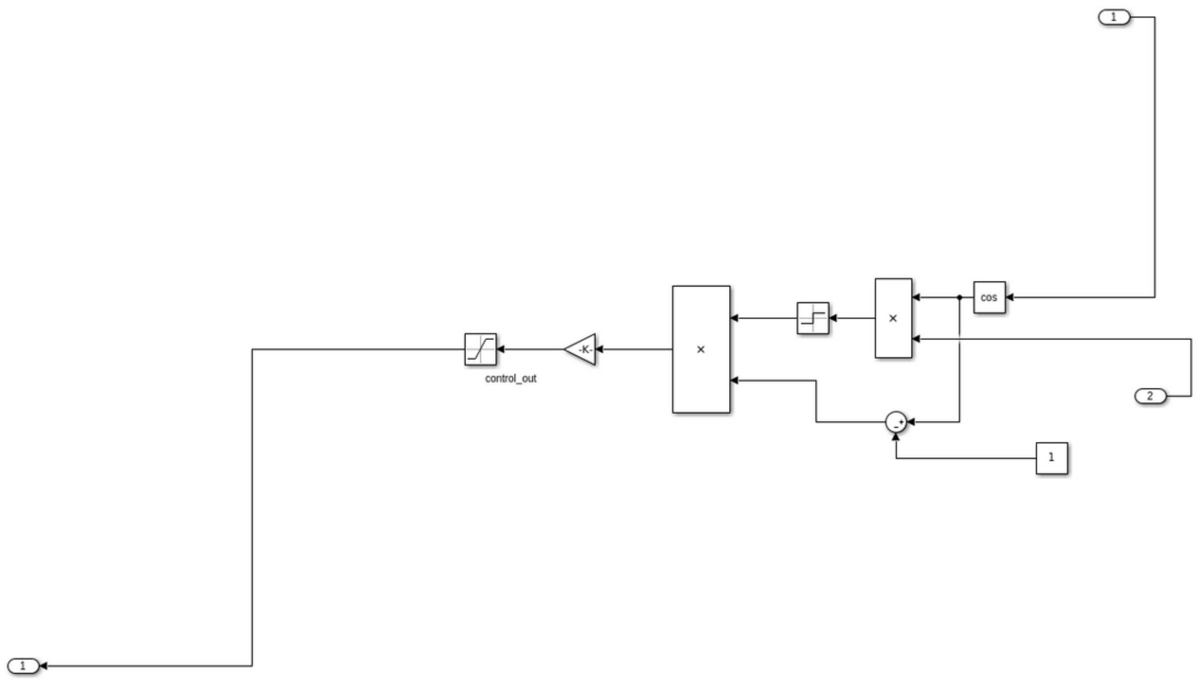
Editor - /home/cclab/Documents/Experiment 1 RotaryPendulum/RotaryPendulum/models/rele

sayali.m*

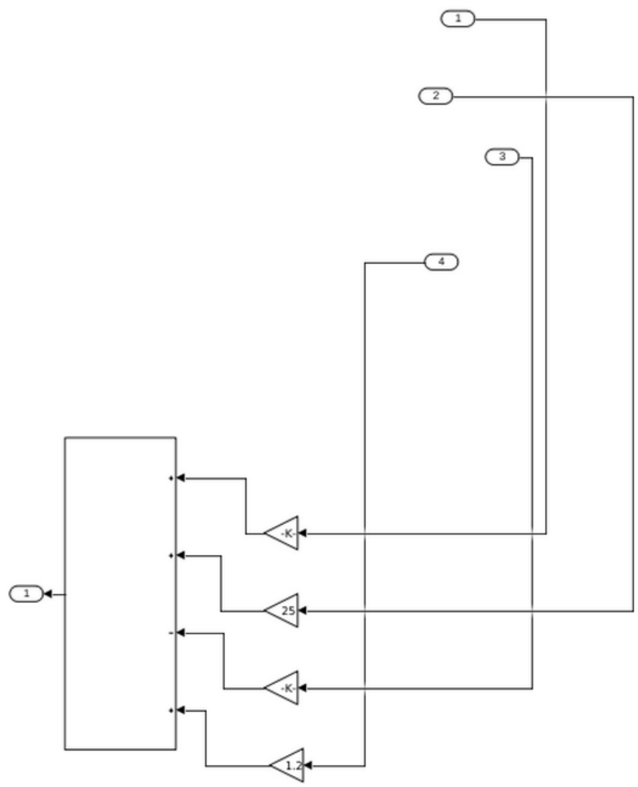
```
clc;
clear all;close all;
g = 9.81;
Jeq = 1.23*10^(-4);
Jp = 1.1*10^(-4);
Rm = 3.30;
Kt = 0.02797;
Km = 0.02797;
r = 0.159;
Mp = 3.3*10^(-3);
lp = 0.153;

A32 = (r*(Mp^2)*(lp^2)*g)/(Jp*Jeq+(Mp*(lp^2)*Jeq)+(Jp*Mp*(r^2)))
a33 = ((Kt*Km*(Jp+(Mp*(lp^2)))))/((Jp*Jeq+(Mp*(lp^2)*Jeq)+(Jp*Mp*(r^2)))*Rm)
A33 = -a33
a42 = (Mp*lp*g*(Jeq+(Mp*(r^2))))/(Jp*Jeq+(Mp*(lp^2)*Jeq)+(Jp*Mp*(r^2)))
A42 = -a42
A43 = (Mp*lp*Kt*r*Km)/((Jp*Jeq+(Mp*(lp^2)*Jeq)+(Jp*Mp*(r^2)))*Rm)

A = [0 0 1 0; 0 0 0 1; 0 A32 A33 0; 0 A42 A43 0]

B31 = (Kt*(Jp+(Mp*(lp^2))))/(Jp*Jeq+(Mp*(lp^2)*Jeq)+(Jp*Mp*(r^2)))
b41 = (Mp*lp*Kt*r)/((Jp*Jeq+(Mp*(lp^2)*Jeq)+(Jp*Mp*(r^2)))*Rm)
B41 = -b41
B = [0;0;B31;B41]
sy = ctrb(A,B)
ansf = rank(sy)
Q = [0.0015 0 0 0;0 680 0 0;0 0 0.0002 0;0 0 0 0.12];
R = 1
k = lqr(A,B,Q,R)
k1 = k(1,1)
k2 = k(1,2)
k3 = k(1,3)
k4 = k(1,4)
```

3.3 Figures of controllers

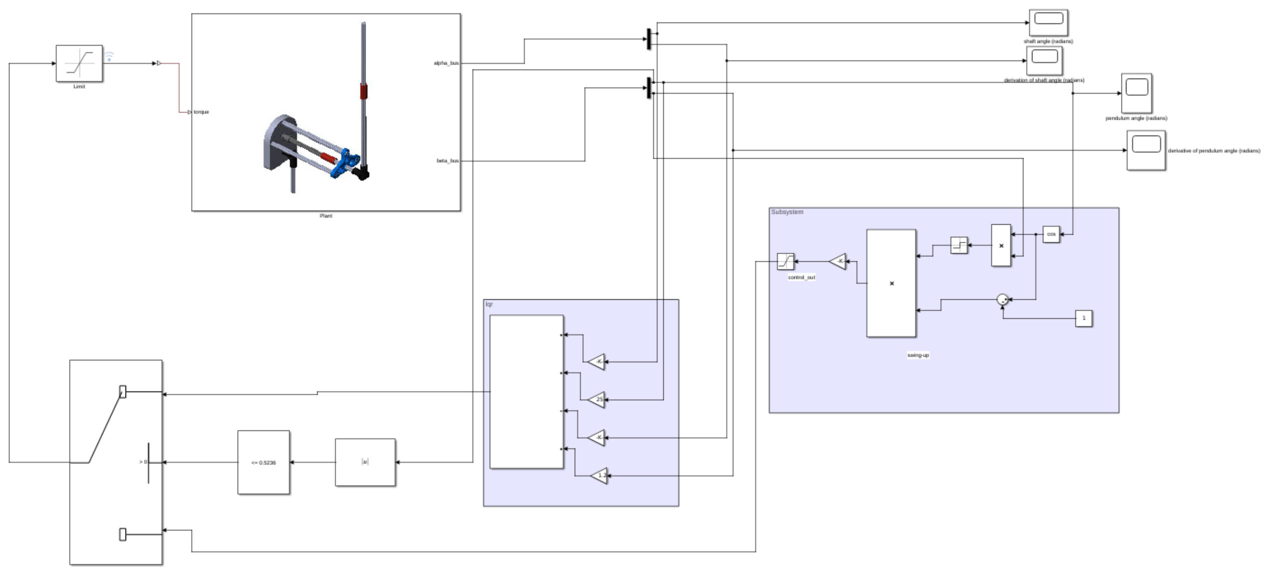
1. Swing-up Controller



2. LQR Controller

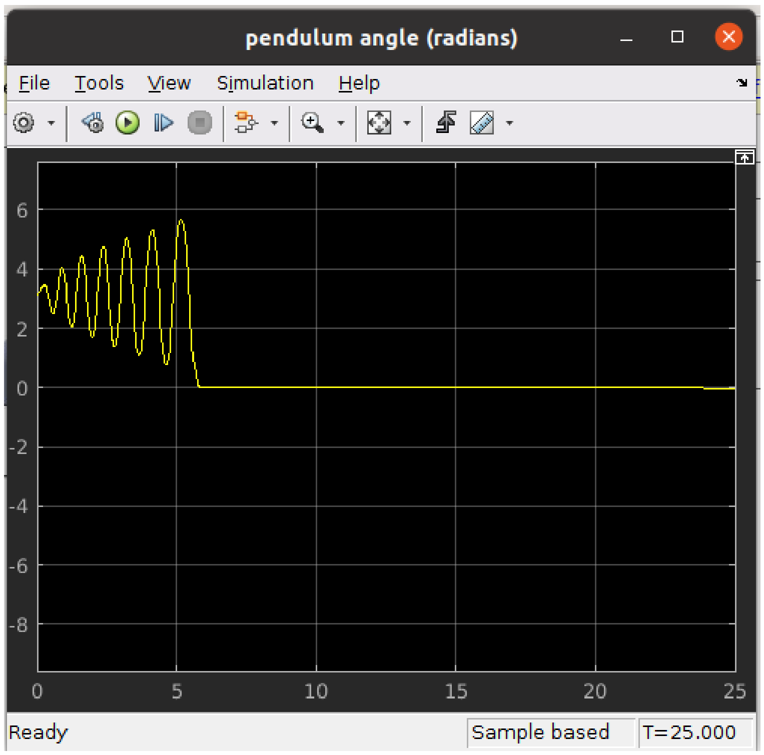


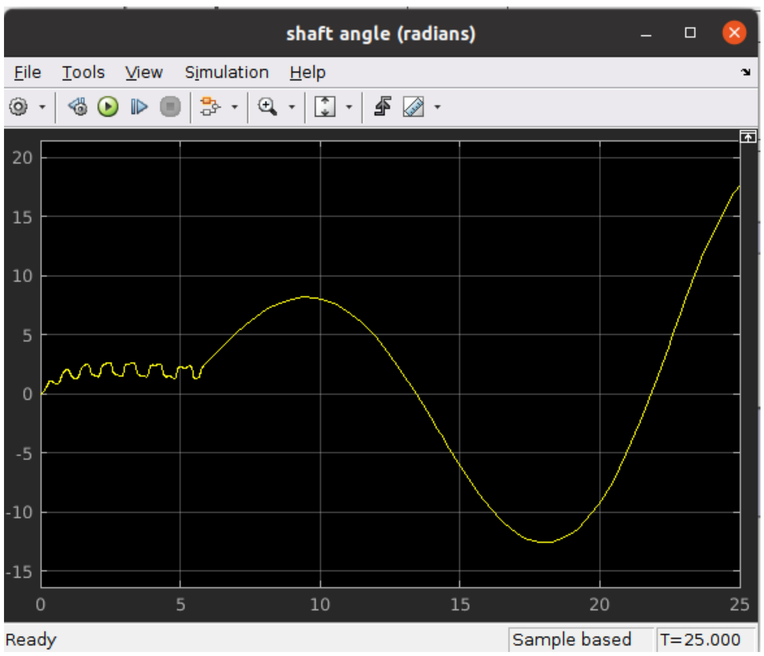
4 Overall Circuit Diagram



5 Result

With the help of above swing up and balance controller, we are able to stable the pendulum at inverted position.




shaft angle (radians)
File Tools View Simulation Help
20
15
10
5
0
-5
-10
-15
0
5
10
15
20
25
Ready
Sample based
T=25.000