MECA 482

Riley Kildare

Edgar Leon

Eric Hansen

Eduardo Turcios

Design Report

December 24, 2019

Department of Mechanical and Mechatronic Engineering and Sustainable Manufacturing
California State University, Chico
Chico, CA 95929-07

Contents

Introduction

A Furuta Inverted pendulum consists of a single motor mounted vertically which rotates an intermediary horizontal shaft. A pendulum which is free to rotate about the intermediary shaft is to be held vertically by the motion of the motor as shown in figure 1. Since the pendulum can rotate about two separate axes, this system has two degrees of freedom.

An encoder is to be used to obtain a measurement of the rotation between the intermediary bar and the pendulum (θ_2 in figure 1).

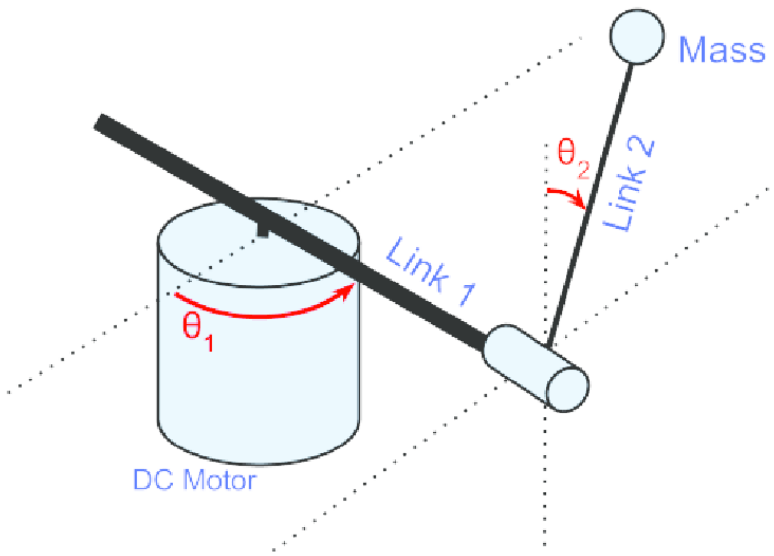


Figure 1: Example Furuta Pendulum

Since this is a nonlinear system, a nonlinear control system and, thus, mathematical model for the system is to be created. This control system is to be modeled within V-Rep to simulate the model as a whole.

Deliverables

- Web page detailing the project, as well as a Github repository.
- Mathematical model of the system written in Matlab (for use with Simulink).
- Control system presented with Simulink and an STM32 based microcontroller.
    - Would be nice if it were rewritten in C for use with a broader number of microcontrollers.
- Simulation of the system in V-Rep

Modeling

A mathematical Furuta Pendulum model is discussed below. This derivation is credited to Nguyen Duc Quyen and Ngo Van Thuyen of the University of Technical Education in Ho Chi Minh City.

Supposing that the gravity of pendulum is at point B. Point B carries out a rotational motion compared with point A and the rotational velocity of point B has the following components

$$\begin{cases} \dot{x}_{AB} = -L\dot{\alpha}\cos\alpha, \\ \dot{y}_{AB} = -L\dot{\alpha}\sin\alpha. \end{cases} \tag{1}$$

Besides, the pendulum still rotates around point O with the velocity $r\dot{\theta}$. Therefore, the velocity of point B compared with fixed point O can be described by the equation as follows

$$\begin{cases} \dot{x}_{B} = r\dot{\theta} - L\dot{\alpha}\cos\alpha, \\ \dot{y}_{B} = -L\dot{\alpha}\sin\alpha. \end{cases} \tag{2}$$

Differentiate two sides of (2), we obtain

$$\begin{cases} \ddot{x}_{B} = r\ddot{\theta} + L\dot{\alpha}^2\sin\alpha - L\ddot{\alpha}\cos\alpha, \\ \ddot{y}_{B} = -L\dot{\alpha}^2\cos\alpha - L\ddot{\alpha}\sin\alpha. \end{cases} \tag{3}$$

This is the result of applying the Newton's Second Law in direction x and in direction y. Figure 2 describes forces acting on the arm and the pendulum. From that,

$$m\ddot{x}_B = \sum F_x \Rightarrow mr\ddot{\theta} + mL\dot{\alpha}^2\sin\alpha - mL\ddot{\alpha}\cos\alpha = A_x, \tag{4}$$

$$m\ddot{y}_B = \sum F_y \Rightarrow -mL\dot{\alpha}^2\cos\alpha - mL\ddot{\alpha}\sin\alpha + mg = A_y. \tag{5}$$

Applying the Euler equation to the rotational motion of pendulum around point B, we obtain

$$J_B\ddot{\alpha} = \sum M_B \Rightarrow -\frac{1}{12}m\left(2L\right)^2\ddot{\alpha} = A_xL\cos\alpha + A_yL\sin\alpha \Rightarrow \frac{1}{3}mL^2\ddot{\alpha} = A_xL\cos\alpha + A_yL\sin\alpha. \tag{6}$$

The equation for the rotational motion of arm around point O has the form

$$J_O\ddot{\theta} = \sum M_O \Rightarrow J_{eq}\ddot{\theta} = T_L - B_{eq}\dot{\theta} - A_xr. \tag{7}$$

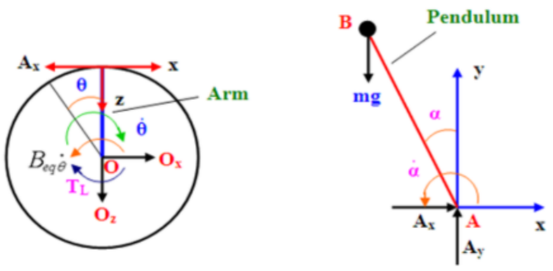


Figure 2. Forces acting on arm and pendulum

Substituting (4) and (5) into (6), we have

$$\frac{1}{3}mL^2\ddot{\alpha} = \left(mr\ddot{\theta} + mL\dot{\alpha}\sin\alpha - mL\ddot{\alpha}\cos\alpha\right)L\cos\alpha + (mg - mL\dot{\alpha}\cos\alpha - mL\ddot{\alpha}\sin\alpha)\,L\sin\alpha$$

$$\Rightarrow -mLr\ddot{\theta}\cos\alpha + \frac{4}{3}mL^2\ddot{\alpha} - mgL\sin\alpha = 0. \tag{8}$$

Substituting (4) into (7), we obtain

$$J_{eq}\ddot{\theta} = T_L - B_{eq}\dot{\theta} - \left(mr\ddot{\theta} + mL\dot{\alpha}^2\sin\alpha - mL\ddot{\alpha}\cos\alpha\right)r$$

$$\Rightarrow \left(J_{eq} + mr^2\right)\ddot{\theta} - mLr\ddot{\alpha}\cos\alpha + mLr\dot{\alpha}^2\sin\alpha = T_L - B_{eq}\dot{\theta}, \tag{9}$$

where

$$T_L = T_m - J_m\ddot{\theta} = \mu_m K_M I_m - J_m\ddot{\theta} = \mu_m K_M \frac{V_m - K_E\dot{\theta}}{R_m} - J_m\ddot{\theta}. \tag{10}$$

Substituting (9) into (10), we have

$$\left(J_{eq} + mr^2 + J_m\right)\ddot{\theta} - mLr\ddot{\alpha}\cos\alpha + mLr\dot{\alpha}^2\sin\alpha + \left(B_{eq} + \mu_m K_M\frac{K_E}{R_m}\right)\dot{\theta} = \mu_m K_M\frac{V_m}{R_m}. \tag{11}$$

The system of equations describing the nonlinear kinetic characteristics of system has the form

$$\ddot{\theta} = \frac{1}{a}\left(b\ddot{\alpha}\cos\alpha - b\dot{\alpha}^2\sin\alpha - e\dot{\theta} + fV_m\right), \ddot{\alpha} = \frac{1}{c}\left(d\sin\alpha + b\ddot{\theta}\cos\alpha\right), \tag{12}$$

where the parameters a, b, c, d, e and f have the form

$$\begin{aligned} &a = J_{eq} + mr^2 + J_m, b = mLr, c = 4mL^2/3, d = mdl, \\ &e = B_{eq} + \mu_m K_M K_E/R_m, f = \mu_m K_M/R_m \end{aligned} \tag{13}$$

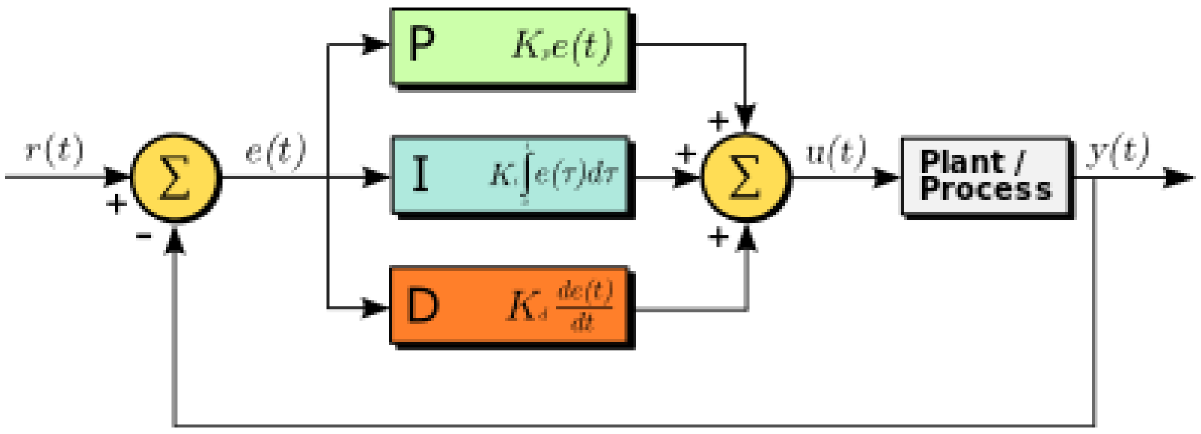
For small $\alpha, \sin\alpha \approx \alpha, \cos\alpha \approx 1$, the linearization of (12) leads to the system of equations as follows

$$\ddot{\theta} = \frac{1}{a}\left(b\ddot{\alpha} - e\dot{\theta} + fV_m\right), \ddot{\alpha} = \frac{1}{c}\left(d\alpha + b\ddot{\theta}\right). \tag{14}$$

The mathematical model obtained in the system of equations (12) is used to build the model of inverted pendulum on the Simulink of Matlab in next sections.

Controller Design and Simulation

The controller used for this project is a PI Controller. In the case of this system, the goal position is when the pendulum is completely vertical in relation to the base. This angle is 0 degrees and is the setpoint of the controller. By measuring the angle of the pendulum, the controller attempts to reach this setpoint using proportional and integral algorithms. A derivative algorithm was not implemented as the system was able to reach stability without it. Figure 2 shows an example block diagram of a PID controller.



This PI controller was programmed with Python in the Spyder IDE. The physical model was then simulated using VREP. The two were interfaced to simulate the Furuta pendulum operating in a physical system.

Without a start up sequence, the system has trouble reaching the set point from initial angles far from it. By initializing the pendulum from an angle close to 0 degrees, the controller is able to hold that angle and maintain stability.

Appendix: A – Python Code

```
import vrep
import sys
import time

velcoef = 0

#-----Try to connect---------------
vrep.simxFinish(-1)
clientID = vrep.simxStart('127.0.0.1',19999,True,True,10000,5)
if clientID != -1:
    print("Connected to remote API server")
else:
    print("Not connected to remote API server")
    sys.exit ("could not connect")

#-----Start the Paused Simulation
err_code = vrep.simxStartSimulation(clientID,vrep.simx_opmode_oneshot)


#-----Initialize Joint Handles---------
err_code,j1 = vrep.simxGetObjectHandle(clientID,"J1",vrep.simx_opmode_blocking)
err_code,j2 = vrep.simxGetObjectHandle(clientID,"J2",vrep.simx_opmode_blocking)

#-----Function to get the position of joint 2 (the free joint)---------
def jupdate():
    err_code,posj2 = vrep.simxGetJointPosition(clientID, j2,vrep.simx_opmode_streaming)
    return posj2


#-----PI----

kp = 5#5
ki = 90#90
erri = 0
inv = 0.01
target = 0


while True:
    po = jupdate() #Get Pendulum Position

    #-----Establish a target - Had trouble with reversing the pendulums momentum without this
    if (po<0):
        target = 0.1
    elif(po>0):
```

```
    target = -0.1
else:
    target = 0
#------

err = target-po #Calculate Error
P = err * kp #Calculated value of P
I = erri * ki #Calculated value of I
erri = erri + inv*err # Keep a running sum of error for the integral function
velcoef = (P + I) #Summation of P and I to get the new velocity of joint 1
#Set the velocity of joint 1
err_code = vrep.simxSetJointTargetVelocity(clientID,j1,velcoef,vrep.simx_opmode_streaming)
time.sleep(inv) #Wait a short abount of time
```

References

QUYEN, NGUYEN. D. (2012). ROTARY INVERTED PENDULUM AND CONTROL OF ROTARY INVERTED PENDULUM BY ARTIFICIAL NEURAL NETWORK. Proc. Natl. Conf. Theor. Phys, 243–249.