Chart

Description automatically generated

**Python Code**

#Program to create Figure 3.6 from Giordano

import pandas as pd

from matplotlib import pyplot as plt

from math import cos,sin,exp,sqrt,pi,radians

import sys

#lists of data from calculations

omega0list=[]

omegahalflist=[]

omega1\_2list=[]

t0list=[]

thalflist=[]

t1\_2list=[]

#variables

length = 9.8

t = 0

thalf = 0

t1\_2 = 0

dt = 0.04

dthalf = 0.04

dt1\_2 = 0.04

theta = 0.2

thetahalf = 0.2

theta1\_2 = 0.2

g = 9.8

omega = 0

omegahalf = 0

omega1\_2 = 0

omegaD = 2/3

q = 0.5

FD1 = 0

FD2 = 0.5

FD3 = 1.2

#Euler-Cromer method for Chaotic Pendulum

for i in range(1500):

omega += ((-(g/length)\*sin(theta)) - q\*omega + FD1\*sin(omegaD\*t))\*dt

theta += omega\*dt

if theta >= pi:

theta -= 2\*pi

if theta <= -pi:

theta += 2\*pi

t += dt

omega0list.append(omega)

t0list.append(t)

for i in range(1500):

omegahalf += ((-(g/length)\*sin(thetahalf)) - q\*omegahalf + FD2\*sin(omegaD\*thalf))\*dthalf

thetahalf += omegahalf\*dthalf

if thetahalf >= pi:

thetahalf -= 2\*pi

if thetahalf <= -pi:

thetahalf += 2\*pi

thalf += dthalf

omegahalflist.append(omegahalf)

thalflist.append(thalf)

for i in range(1500):

omega1\_2 += ((-(g/length)\*sin(theta1\_2)) - q\*omega1\_2 + FD3\*sin(omegaD\*t1\_2))\*dt1\_2

theta1\_2 += omega1\_2\*dt1\_2

if theta1\_2 >= pi:

theta1\_2 -= 2\*pi

if theta1\_2 <= -pi:

theta1\_2 += 2\*pi

t1\_2 += dt1\_2

omega1\_2list.append(omega1\_2)

t1\_2list.append(t1\_2)

K = -1

L = -4

omegahalf2list=[x + K for x in omegahalflist]

omega1\_2\_2list=[x + L for x in omega1\_2list]

my\_yticks = ['', '', '-1', '0', '1', '', '0', '0', '1']

yticks = [-7,-6,-5,-4,-3,-2,-1,0,1]

#Creates a plot with both sets of data

%matplotlib

fig = plt.gcf()

fig.set\_size\_inches(5.65, 5)

plt.plot(t0list, omega0list, 'k-')

plt.plot(thalflist, omegahalf2list, 'k-')

plt.plot(t1\_2list, omega1\_2\_2list, 'k-')

plt.xlim(0,60)

plt.ylim(-7,1)

plt.yticks(yticks, my\_yticks)

plt.xticks(ticks = (0,20,40,60))

plt.title("\u03C9 versus time")

plt.xlabel("time (s)")

plt.ylabel("\u03C9 (radians/s)")

plt.text(45, 0.1, "$F\_{D} = 0$", fontsize=14, horizontalalignment='left', verticalalignment='bottom')

plt.text(45, -2.1, "$F\_{D} = 0.5$", fontsize=14, horizontalalignment='left', verticalalignment='bottom')

plt.text(39, -5.7, "$F\_{D} = 1.2$", fontsize=14, horizontalalignment='left', verticalalignment='bottom')

plt.tick\_params(direction = 'in', right = True, top = True)

plt.show()