This task is on GitHub.

Disaster

Python Code

*#!/usr/bin/env pybricks-micropython*

from math import \*

from pybricks.hubs import EV3Brick

from pybricks.ev3devices import (Motor, TouchSensor, ColorSensor,

                                 InfraredSensor, UltrasonicSensor, GyroSensor)

from pybricks.parameters import Port, Stop, Direction, Button, Color

from pybricks.tools import wait, StopWatch, DataLog

from pybricks.robotics import DriveBase

from pybricks.media.ev3dev import SoundFile, ImageFile

ev3 = EV3Brick()

mr = Motor(port=Port.A)

ml = Motor(port=Port.D)

x = 0

y = 0

L = 0

R = 2.7

B = 17

k1= 1

k2= 20

MAX\_TIME = 15

time\_present = 0

Tg = (-50,-50)

def PID(past\_time,present\_time,error,ki=0,kp=1,kd=0,anti\_over=True,range\_u=7):

    i = ki\*error\*(present\_time-past\_time)

    p = kp\*error

    d = kd\*error/(present\_time-past\_time)

    rvalue = p+i+d

    if (anti\_over == False):

        return rvalue

    elif (abs(rvalue)<range\_u):

        return rvalue/range\_u\*100

    else :

        return rvalue/abs(rvalue)\*100

def Lf(langle,rangle,r):

    return (langle+rangle)\*r/2

def theta(langle,rangle,r,B):

    return (rangle-langle)\*r/B

def dist(x,y):

    return sqrt((x-Tg[0])\*(x-Tg[0])+(y-Tg[1])\*(y-Tg[1]))

def angle(x,y,theta):

    return atan2((Tg[1]-y),(Tg[0]-x))-theta

def Vfunc(misa=0.0,misd=0.0,ks=0.0,kr=0.0,mode="linear",range\_u=7):

    if(mode==("linear")):

        data.log("linear mode")

        rvalue = misa\*kr+misd\*ks

    else:

        data.log("not linear mode")

        rvalue = (k1\*cos(misa)\*sin(misa)+k2\*misa)\*kr+(misd\*cos(misa)\*k1)\*ks

    if(abs(rvalue)>range\_u):

        return 100\*abs(rvalue)/rvalue

    return rvalue/range\_u\*100

ev3.speaker.say("Running start!")

sw = StopWatch()

data = DataLog(append=True,name="BiggerK2NlData--")

data.log("x,y,L,t",k1,k2)

while True:

    time\_last = time\_present

    time\_present = sw.time()

    l\_angle = ml.angle()\*pi/180

    r\_angle = mr.angle()\*pi/180

    theta\_ = theta(langle=l\_angle,rangle=r\_angle,r=R,B=B)

    dL = Lf(langle=l\_angle,rangle=r\_angle,r=R)-L

    dx = dL\*cos(theta\_)

    x = x+dx

    dy = dL\*sin(theta\_)

    y = y+dy

    L = Lf(langle=l\_angle,rangle=r\_angle,r=R)

    data.log(x,y,L,time\_present,l\_angle,r\_angle,dx,dy,dL,theta\_)

    if ((time\_present>MAX\_TIME\*1000)|(dist(x,y)<5)):

        break

    ml.dc(Vfunc(angle(x,y,theta\_),dist(x,y),0.1,-20,""))

    mr.dc(Vfunc(angle(x,y,theta\_),dist(x,y),0.1,20,""))

MATLAB Code

CKNLDF =  ["BiggerK1NLData--","BiggerK2NLData--"];

LNDF =  ["Data0-","Data0+","Data-0","Data--","Data+0","Data+-","Data-+","Data++"];

NLDF =   ["NLData0-","NLData0+","NLData-0","NLData--","NLData+0","NLData+-","NLData-+","NLData++"];

pos = [0,-50;0,50;-50,0;-50,-50;50,0;50,-50;-50,50;50,50];

for i = (1:2)

    figure("Name","ChangedControl")

    grid on;

    data = readmatrix(CKNLDF(i));

    plot(data(:,1),data(:,2));

    print("DataChangedKLinearControl"+i,"-djpeg");

end

close all;

for i = (1:8)

    figure("Name","DataFromLinearControl")

    grid on;

    data = readmatrix(LNDF(i));

    plot(data(:,1),data(:,2));

    print("DataFromLinearControl"+i,"-djpeg");

end

close all;

for i = (1:8)

    figure("Name","DataFromNotLinearControl")

    grid on;

    data = readmatrix(NLDF(i));

    plot(data(:,1),data(:,2));

    print("DataFromNotLinearControl"+i,"-djpeg");

end

close all;

for i = (1:8)

    x\_goal = pos(i,1);

    y\_goal = pos(i,2);

    simout = sim("LySim.slx");

    figure("Name","DataFromSim")

    grid on;

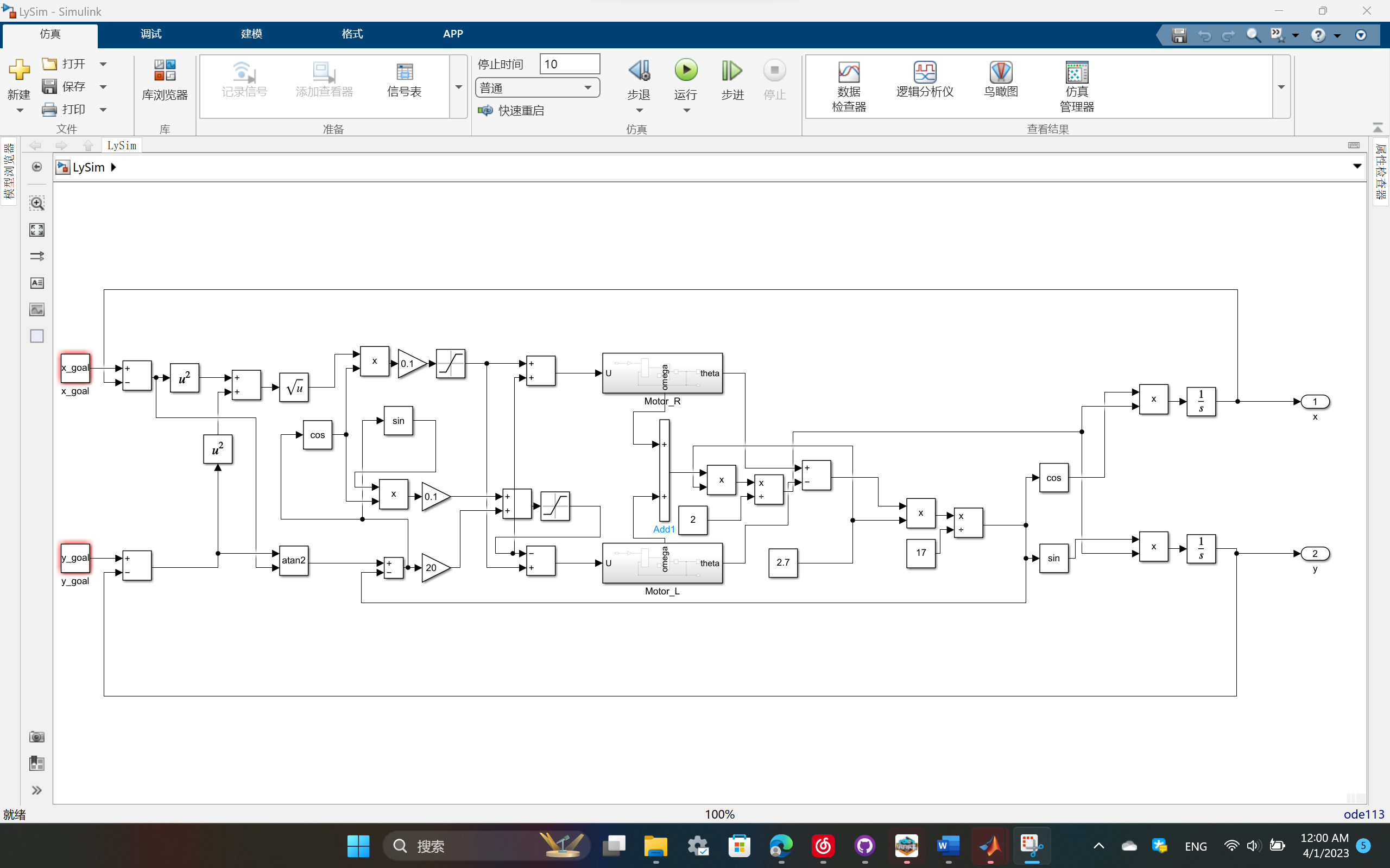
    plot(simout.yout{1}.Values.Data,simout.yout{2}.Values.Data);

    print("DataFromSimulation"+i,"-djpeg");

end

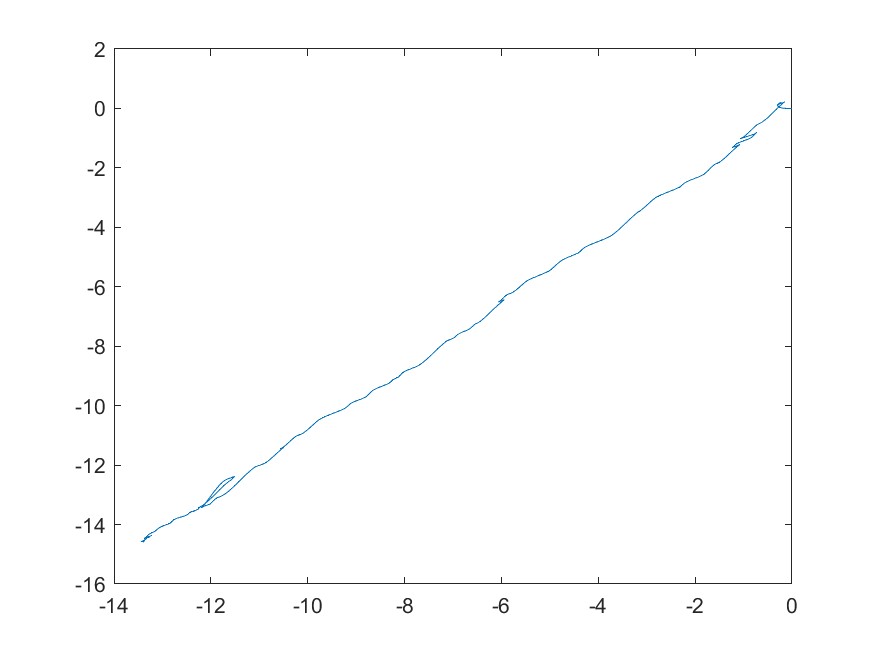
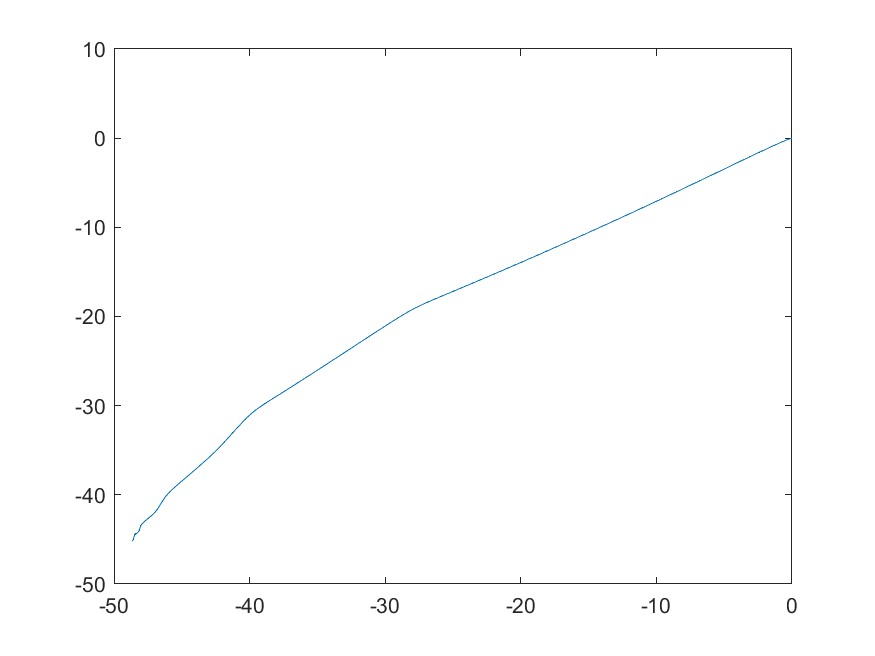
close all;

Simulink Model (by my own)

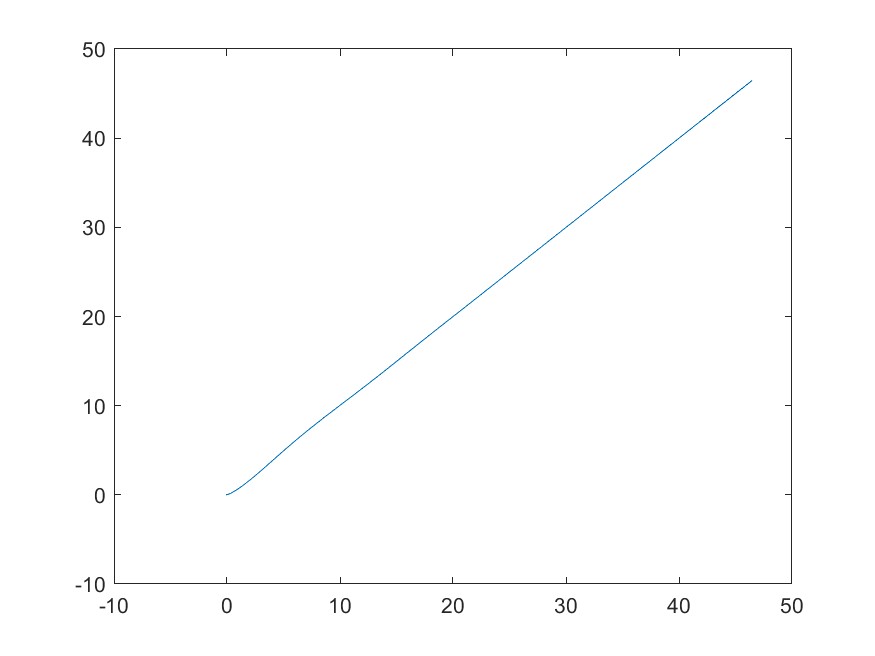
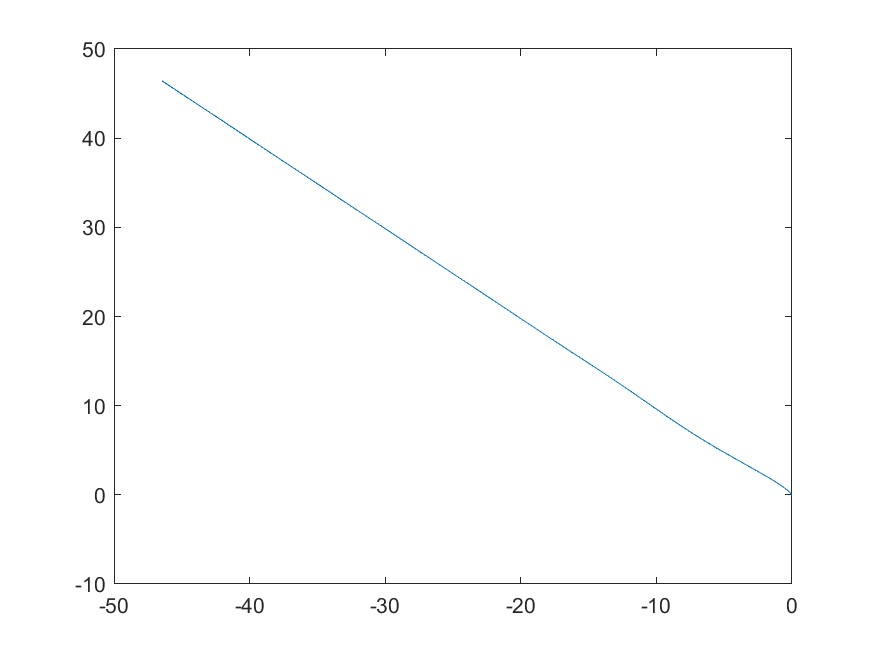
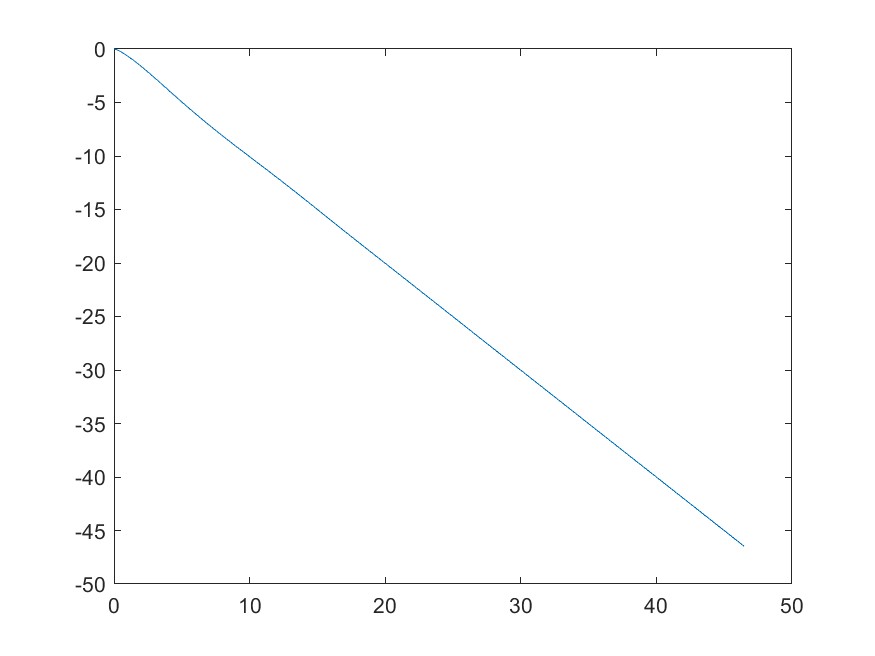
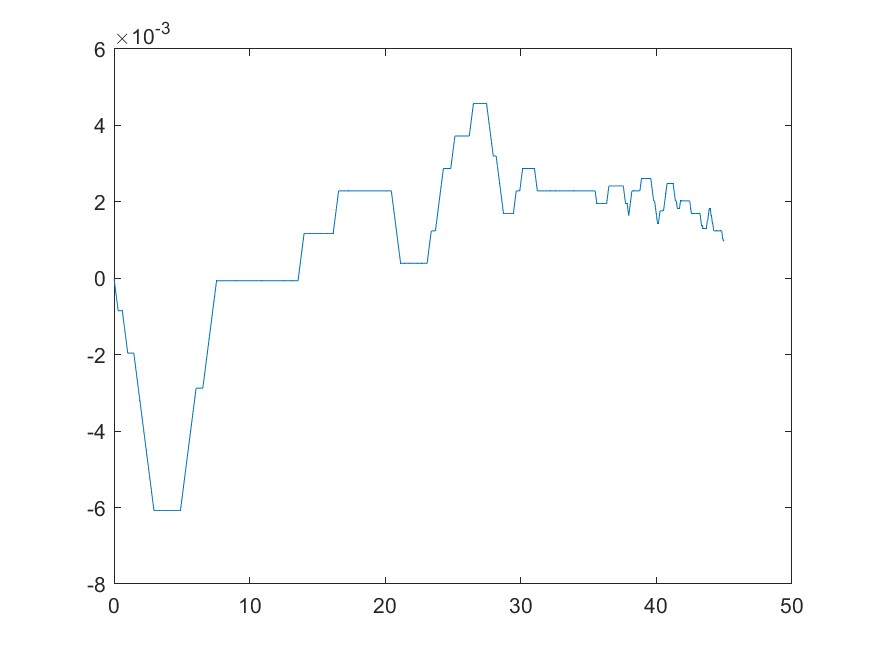
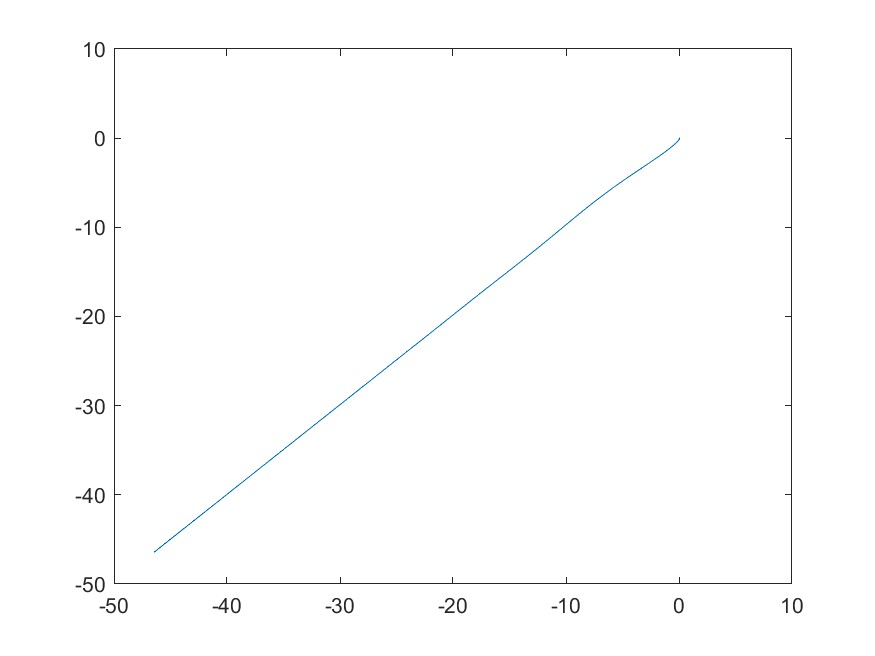
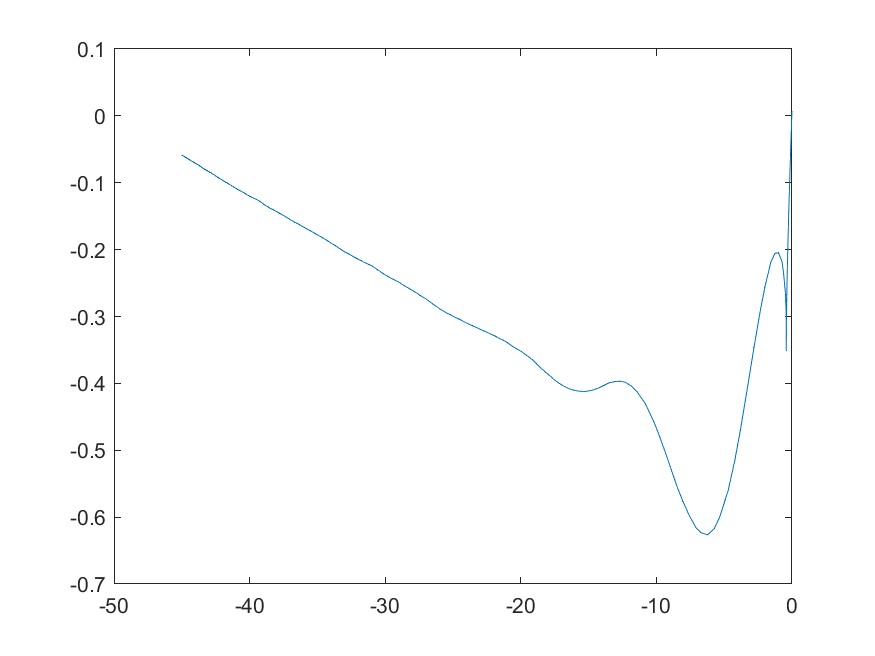
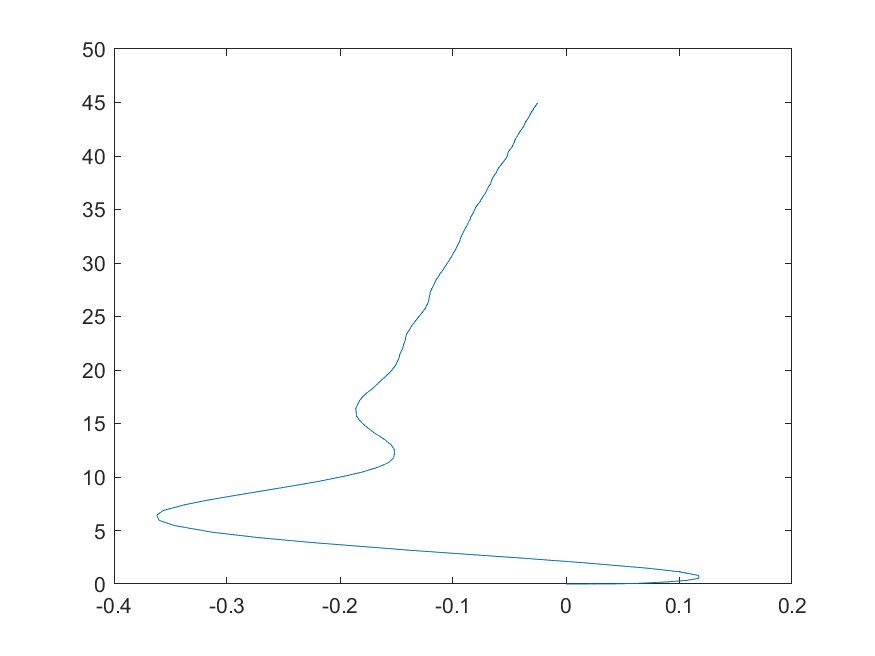
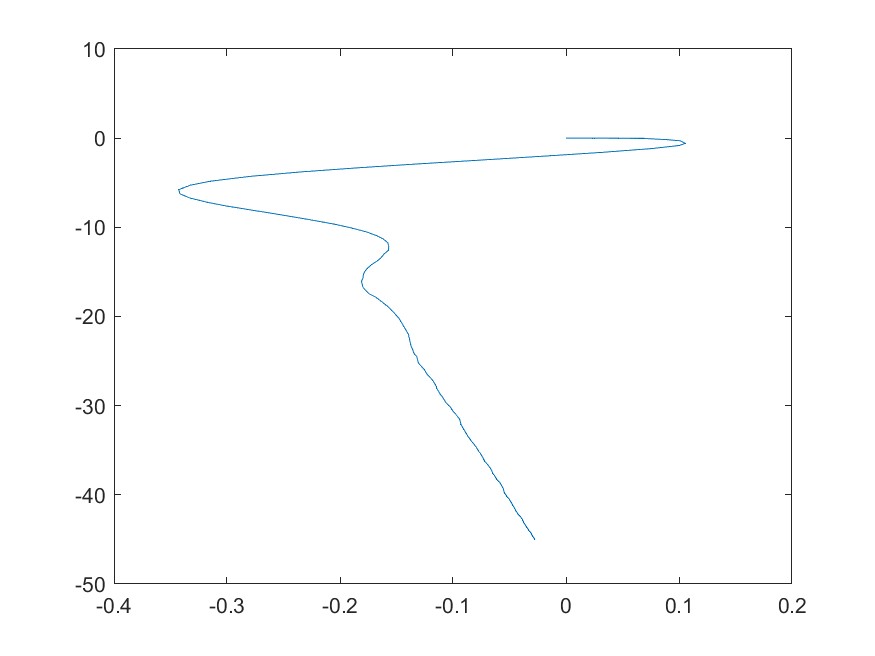


K changed non-linear controller

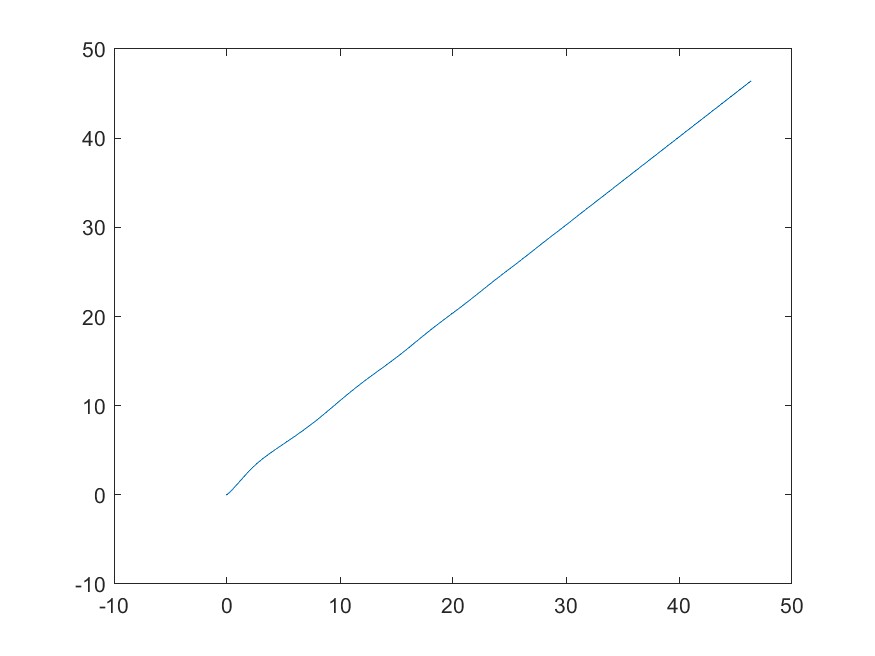
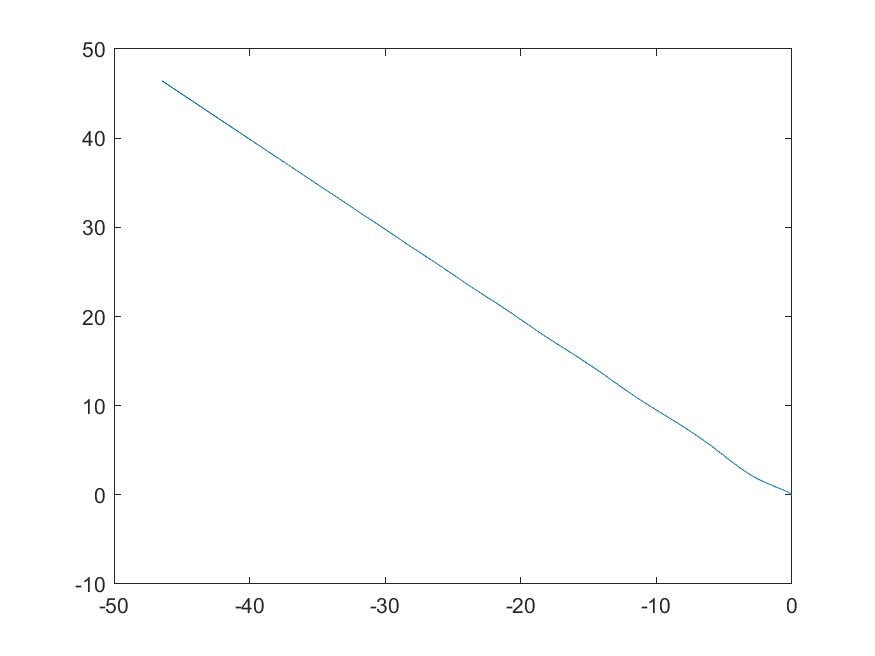
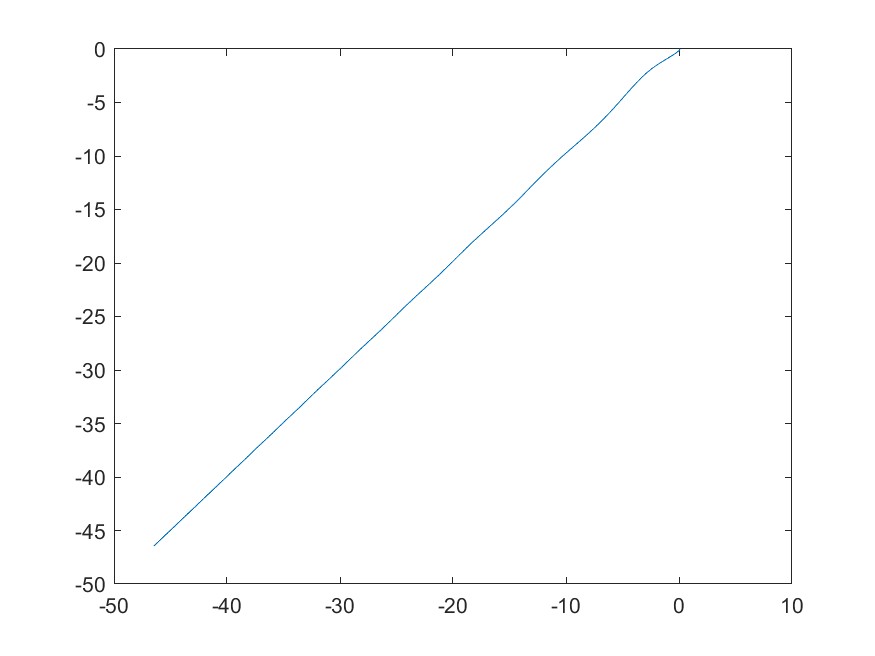
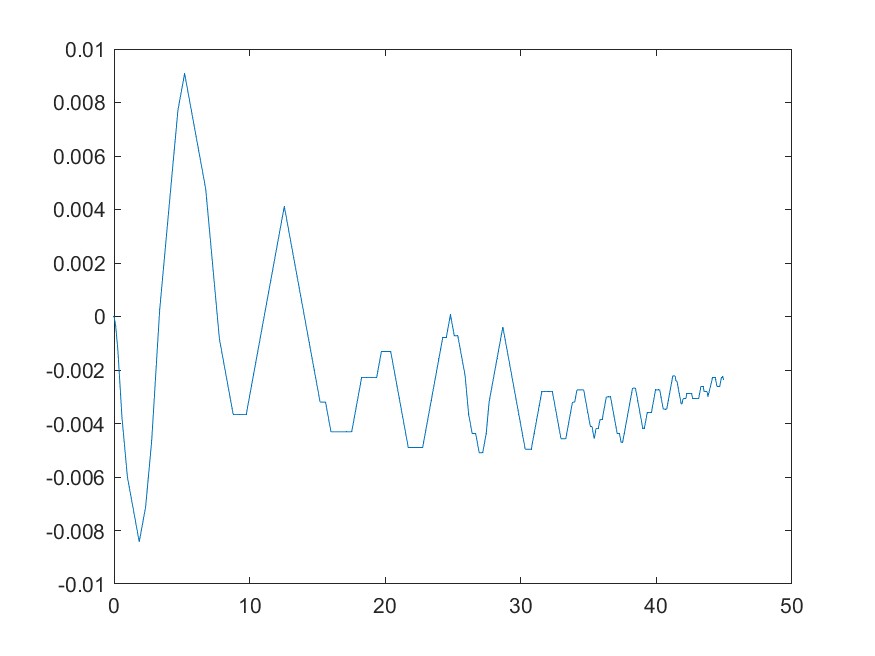
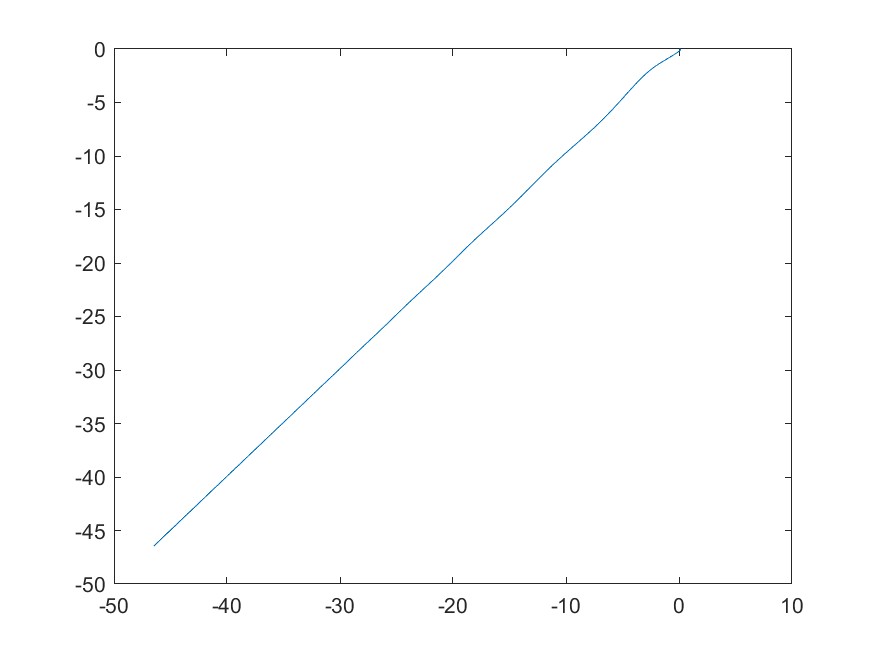
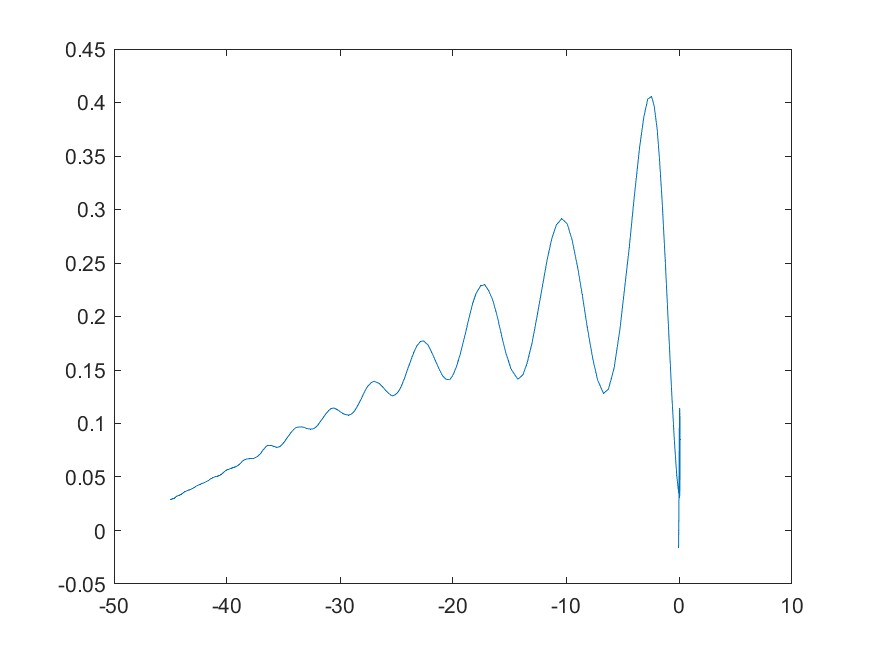
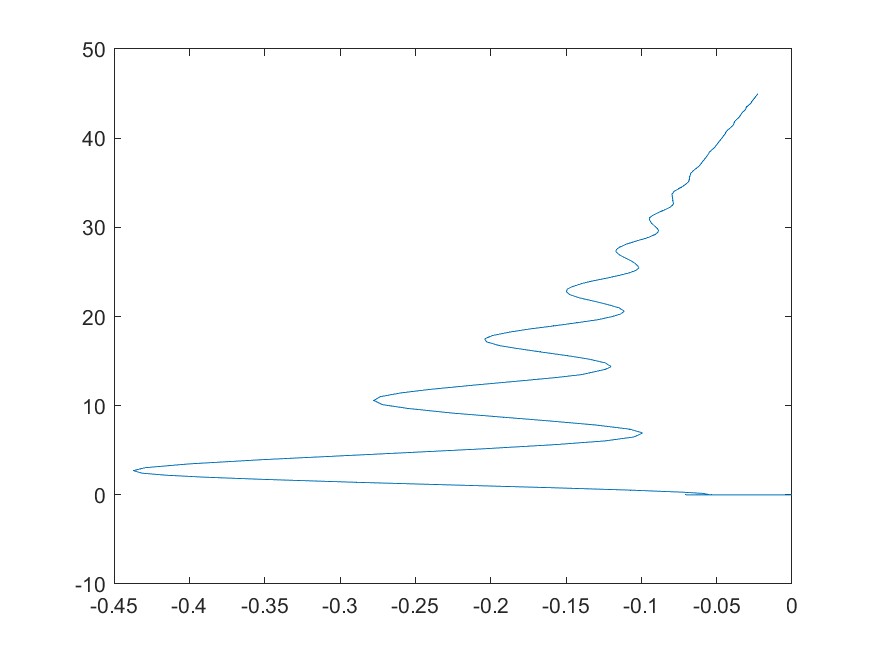
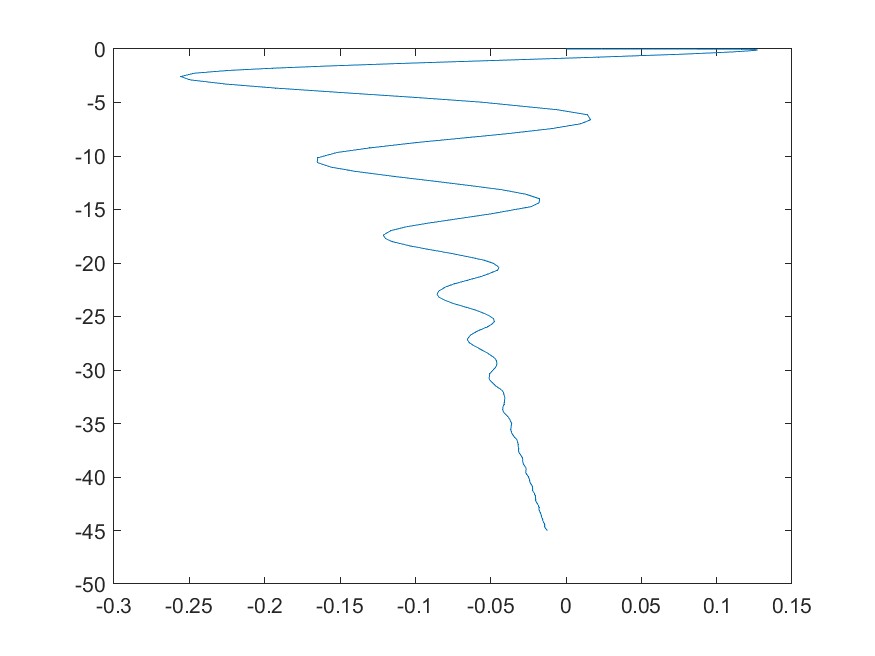
(First Bigger k1,Second Bigger k2)



Linear controller



Non-Linear Controller



Non-Linear Controller From Simulink

