This task is on GitHub.

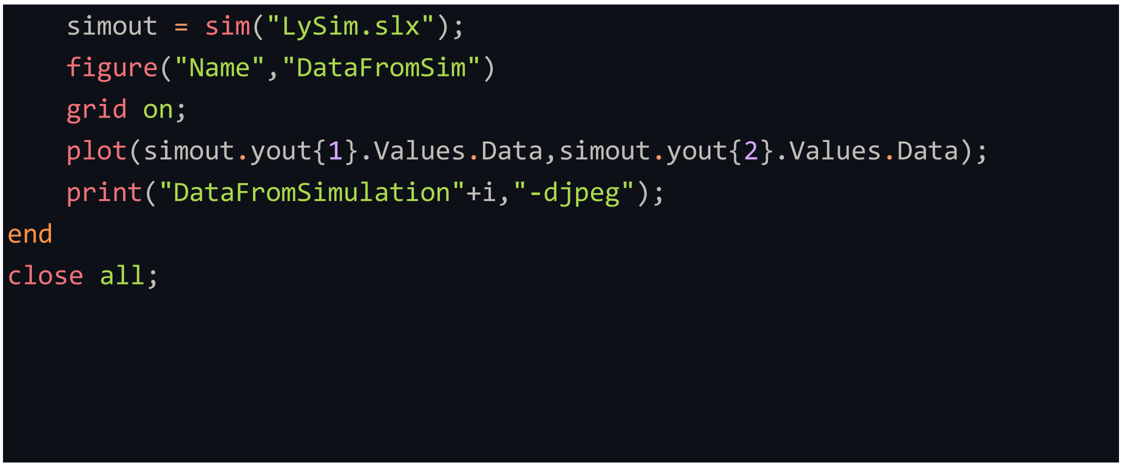
Disaster

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,""))
```

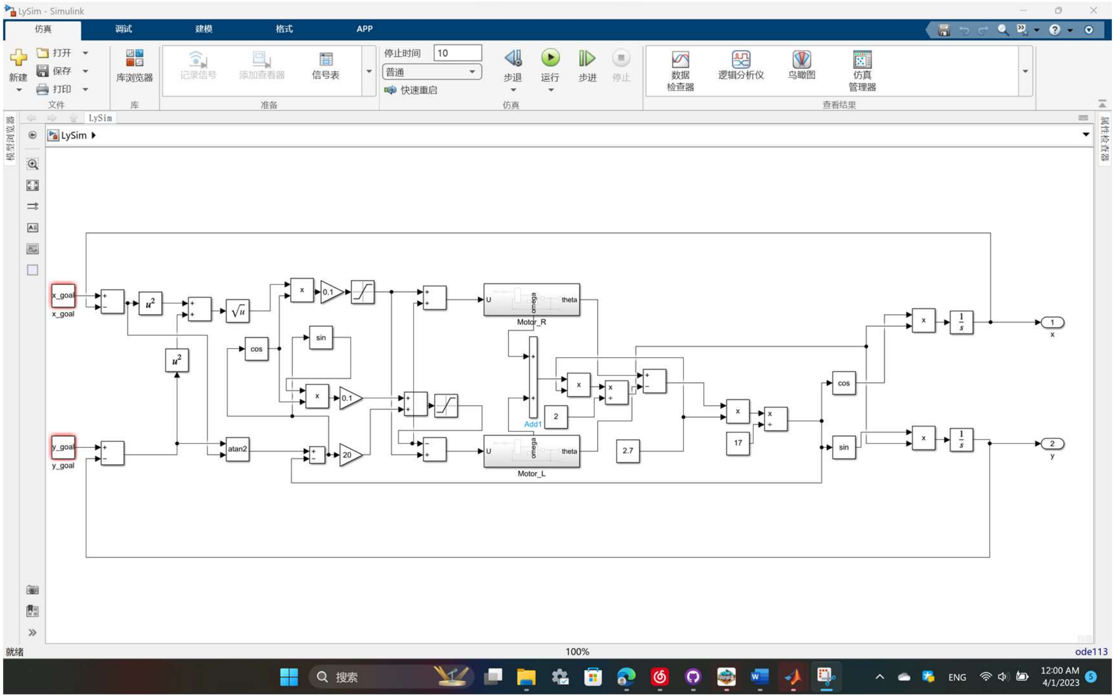
MATLAB Code

```
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);
```



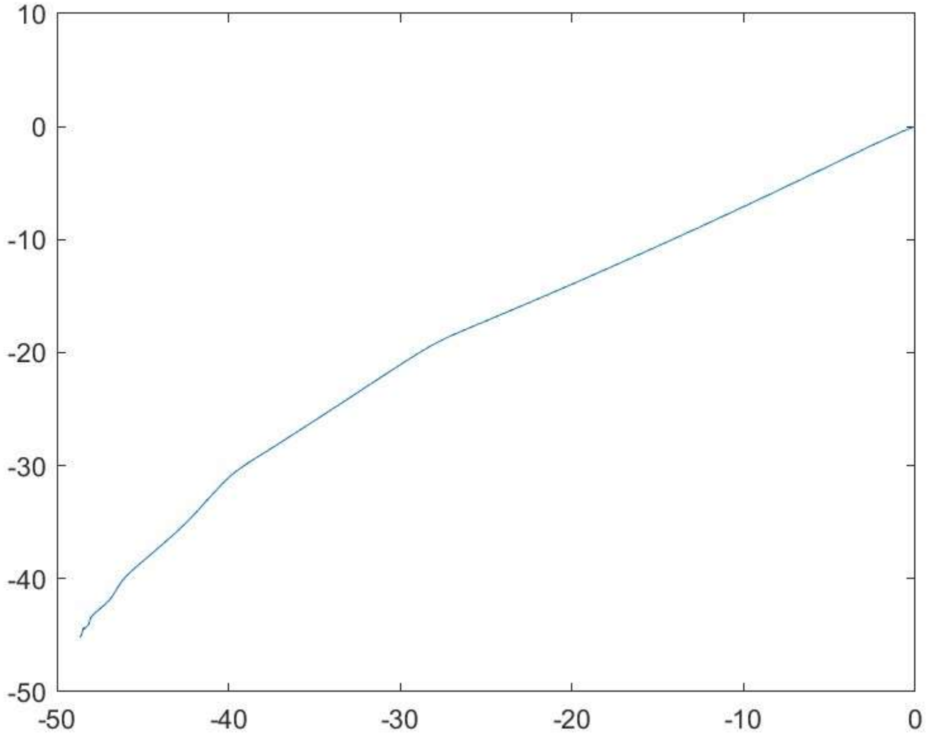
```
    simout = sim("LySim.slx");
    figure("Name","DataFromSim")
    grid on;
    plot(simout.yout{1}.Values.Data,simout.yout{2}.Values.Data);
    print("DataFromSimulation"+i,"-djpeg");
end
close all;
```

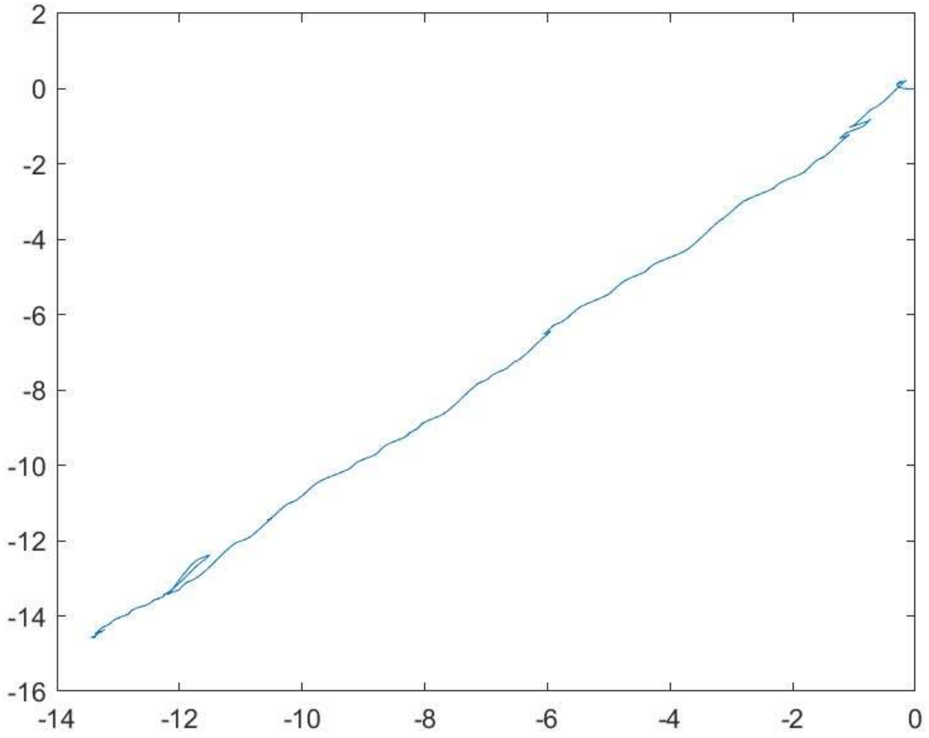
Simulink Model (by my own)



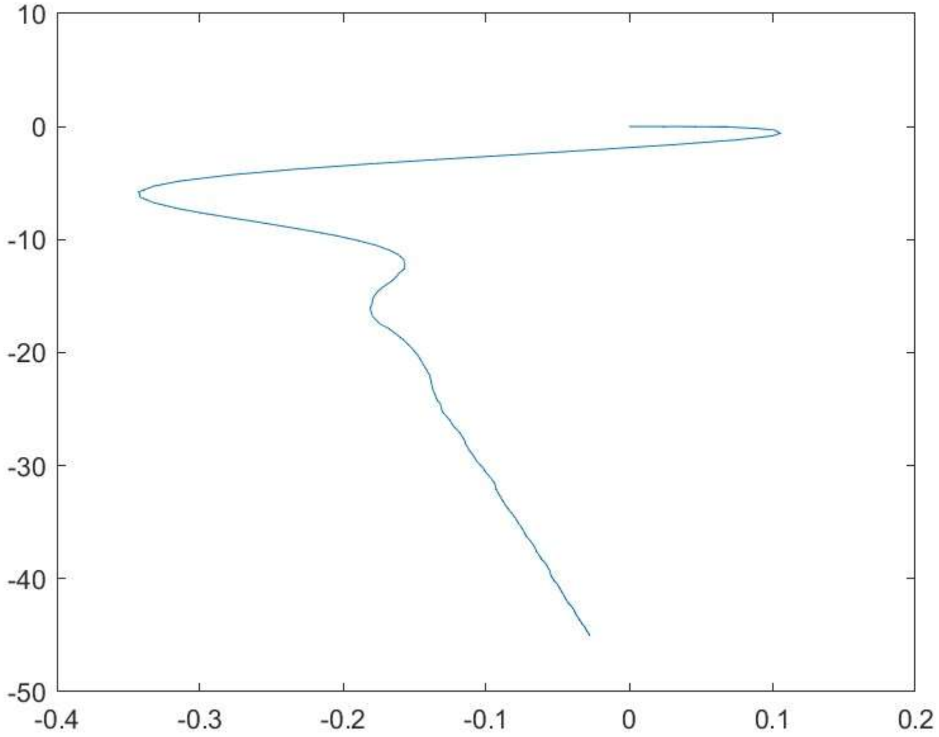
K changed non-linear controller

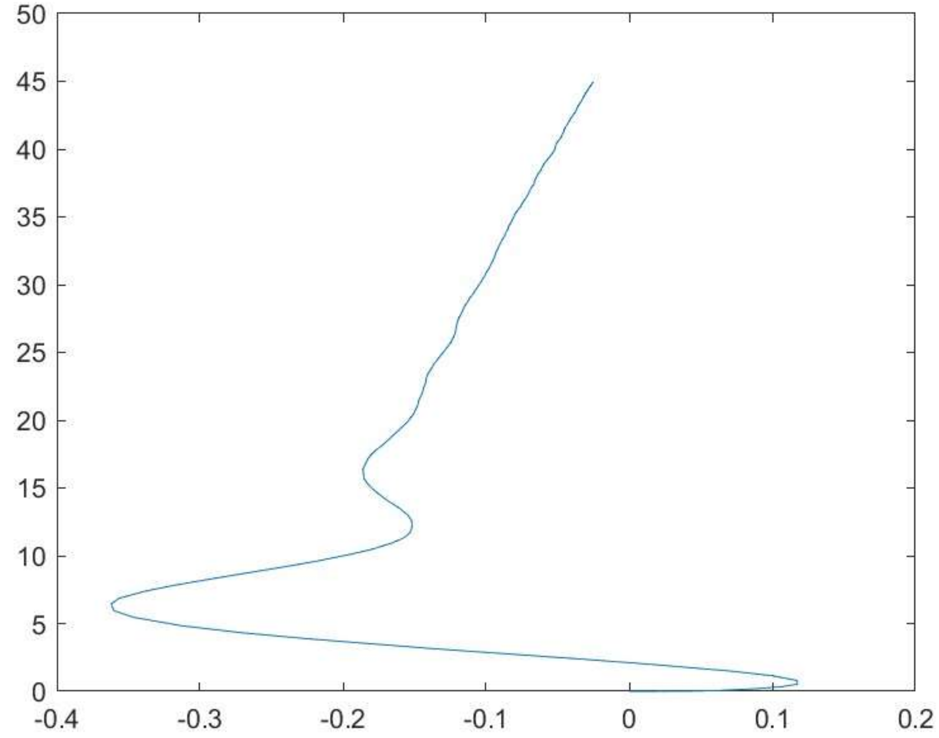
(First Bigger k1,Second Bigger k2)

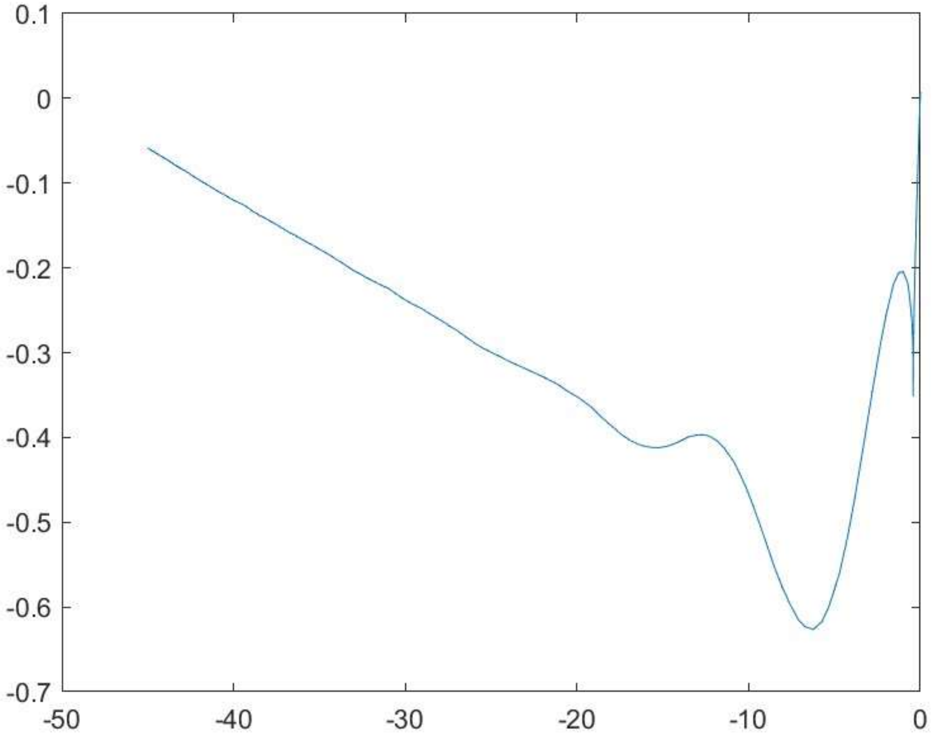


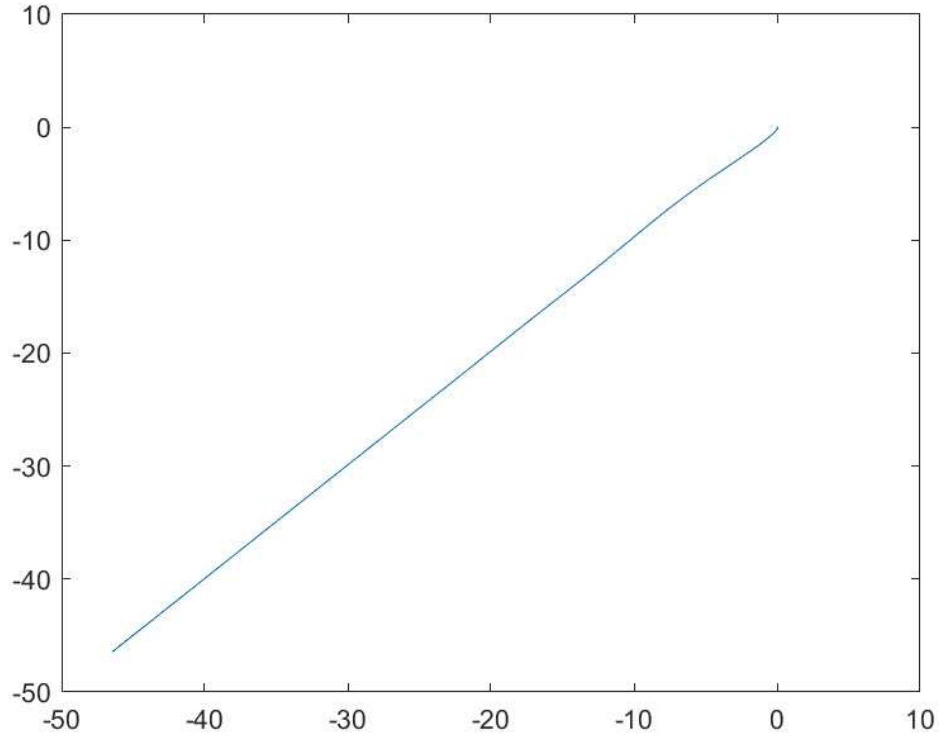


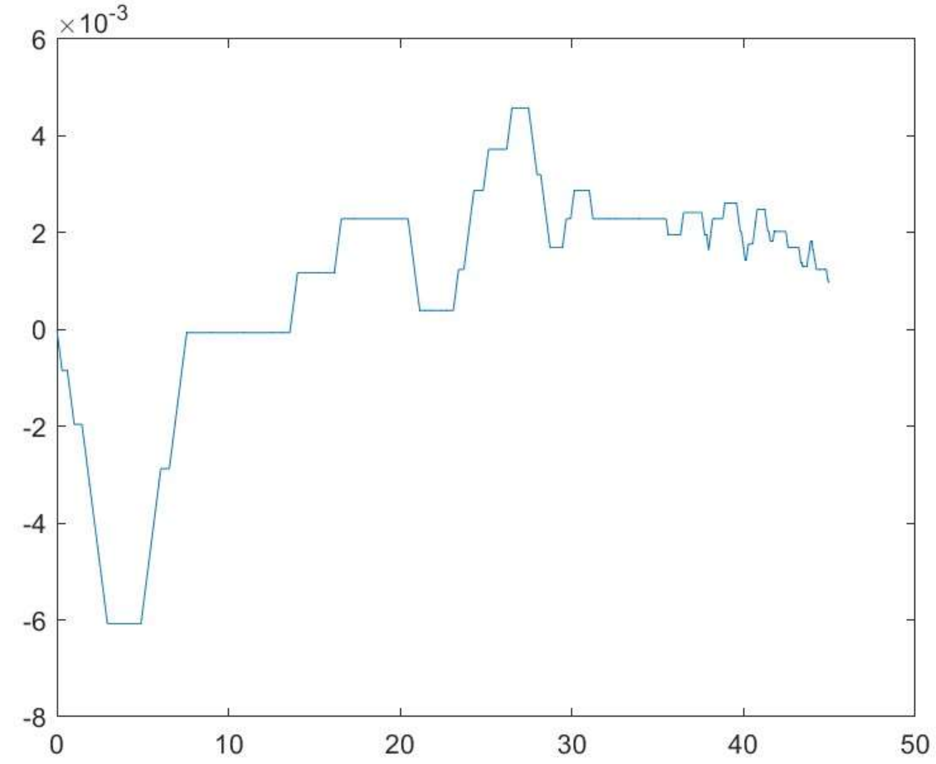
Linear controller

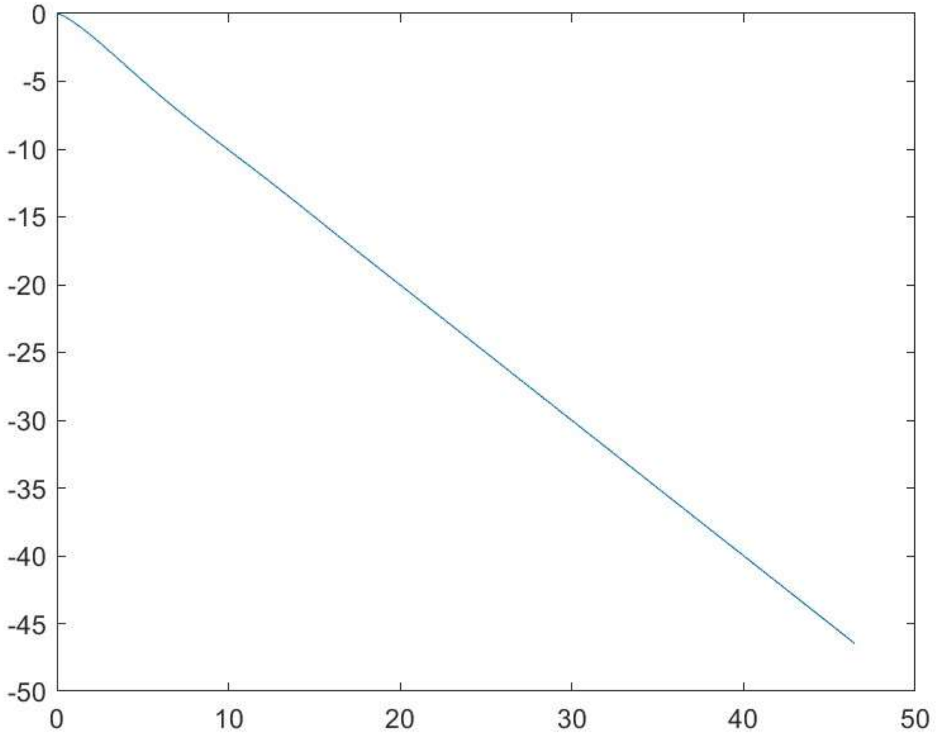

10
0
-10
-20
-30
-40
-50
-0.4
-0.3
-0.2
-0.1
0
0.1
0.2

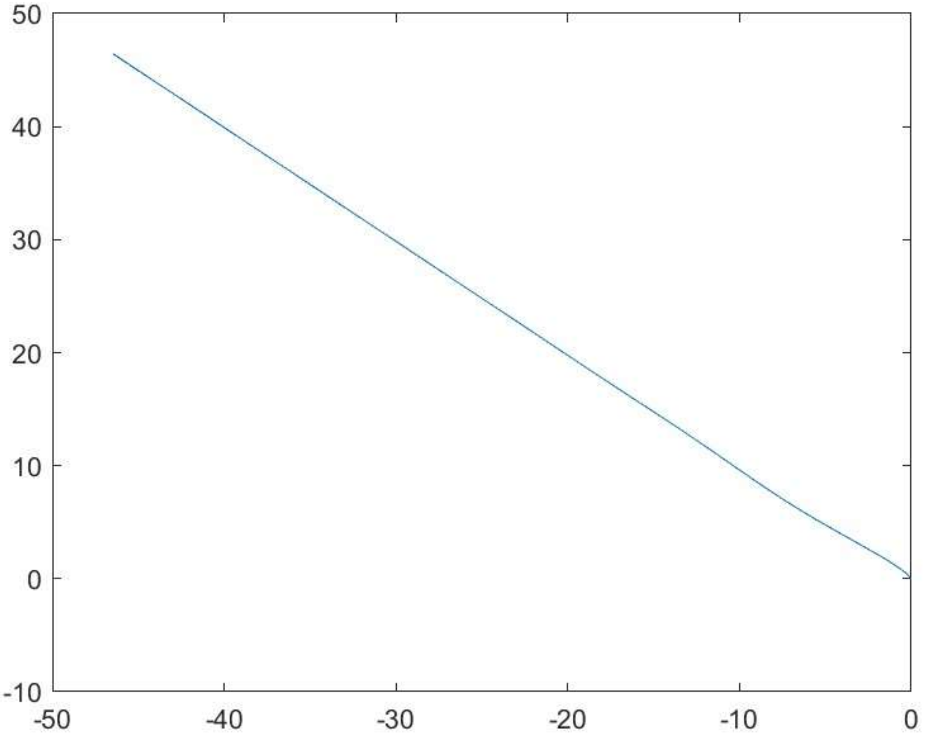

50
45
40
35
30
25
20
15
10
5
0
-0.4
-0.3
-0.2
-0.1
0
0.1
0.2

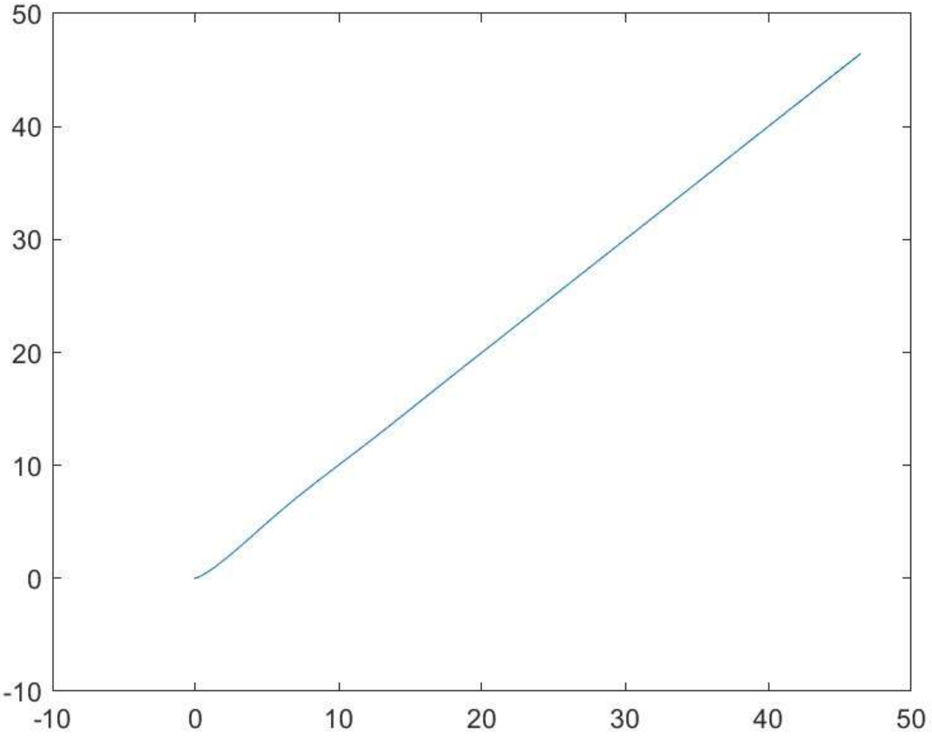

0.1
0
-0.1
-0.2
-0.3
-0.4
-0.5
-0.6
-0.7
-50
-40
-30
-20
-10
0


10
0
-10
-20
-30
-40
-50
-50
-40
-30
-20
-10
0
10

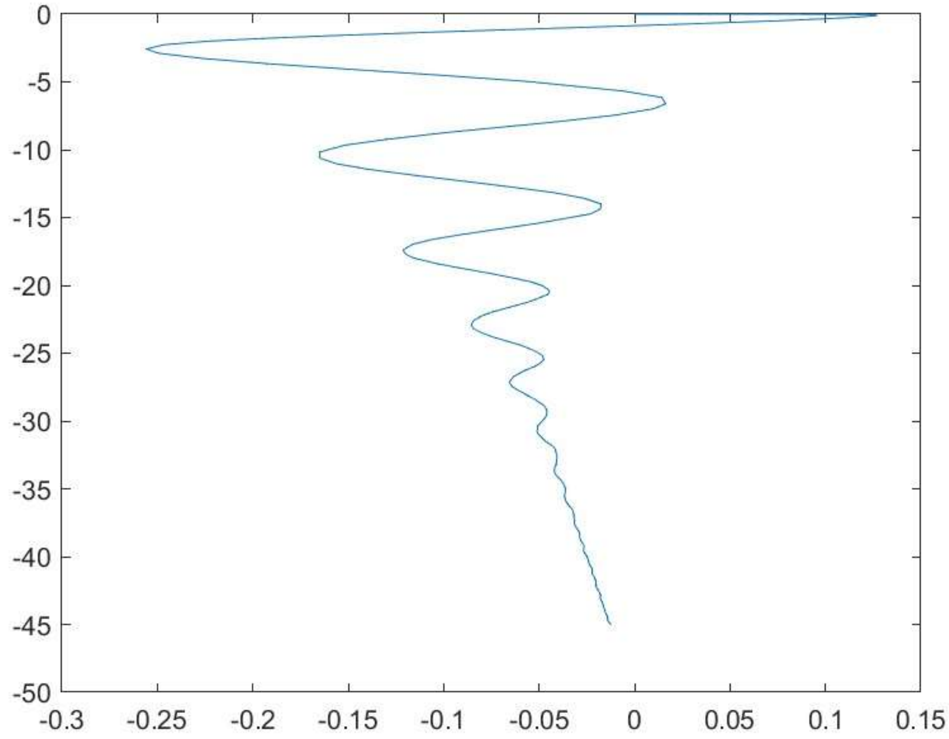

×10^-3
6
4
2
0
-2
-4
-6
-8
0
10
20
30
40
50

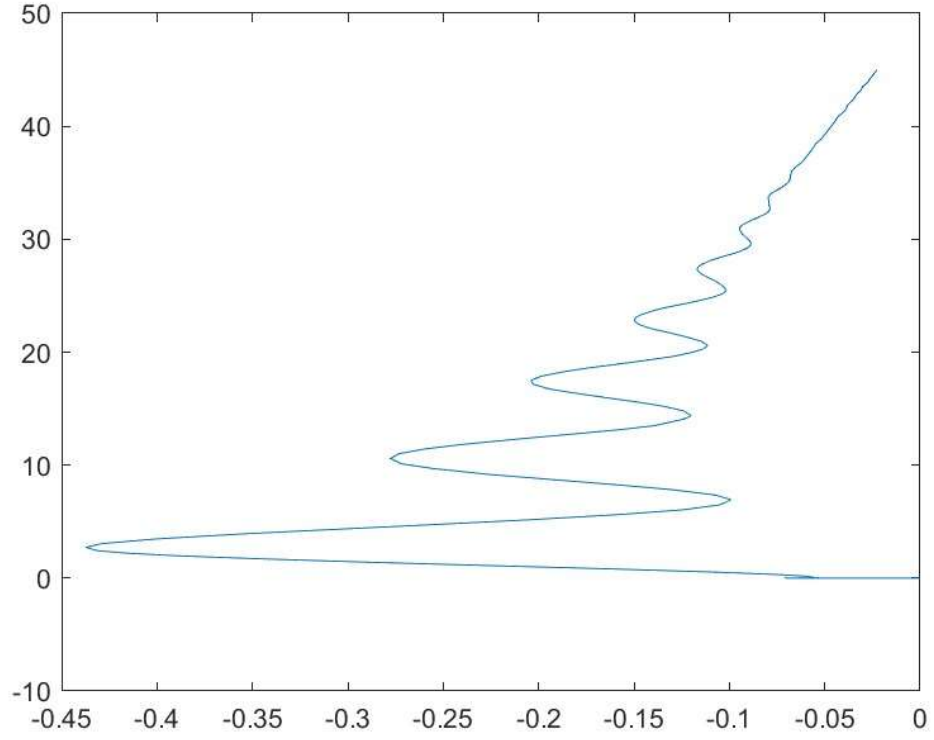

0
-5
-10
-15
-20
-25
-30
-35
-40
-45
-50
0
10
20
30
40
50

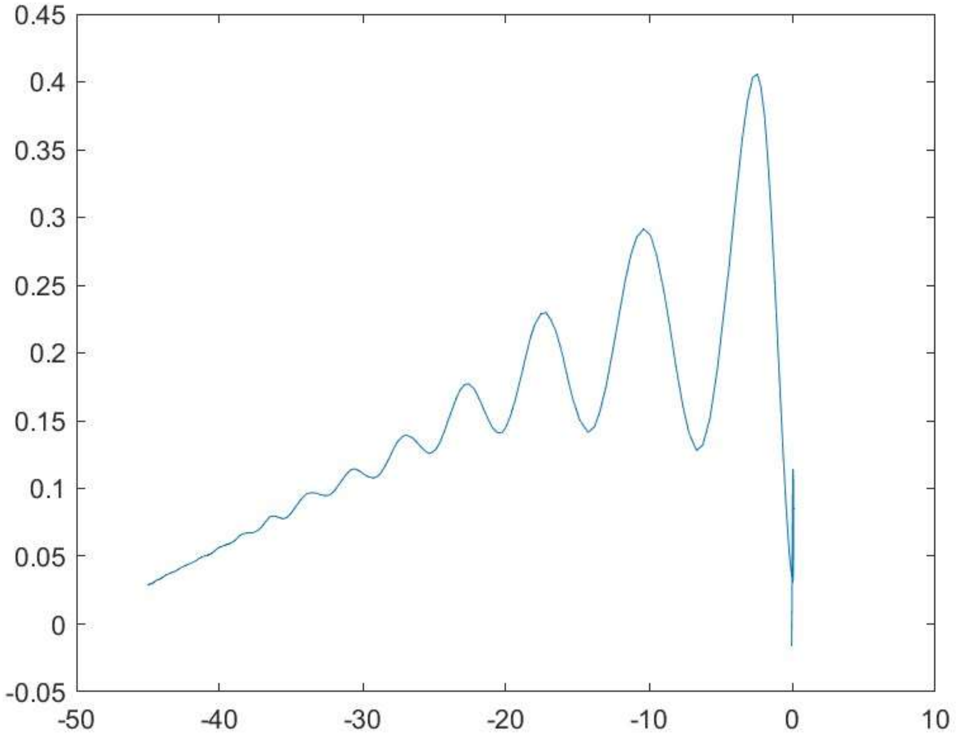

50
40
30
20
10
0
-10
-50
-40
-30
-20
-10
0

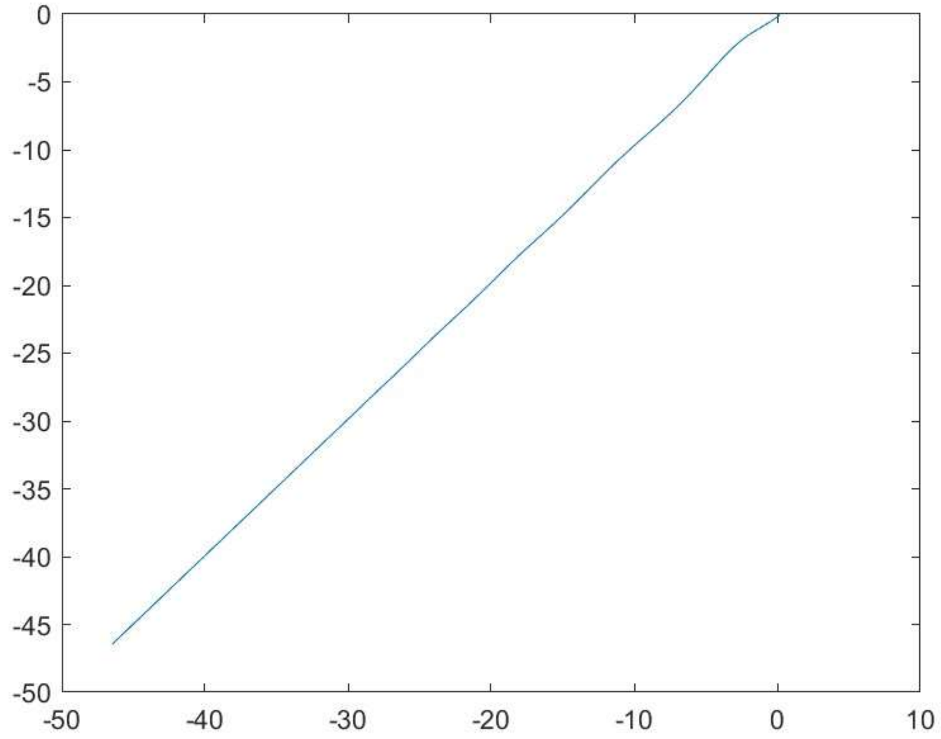

50
40
30
20
10
0
-10
-10
0
10
20
30
40
50

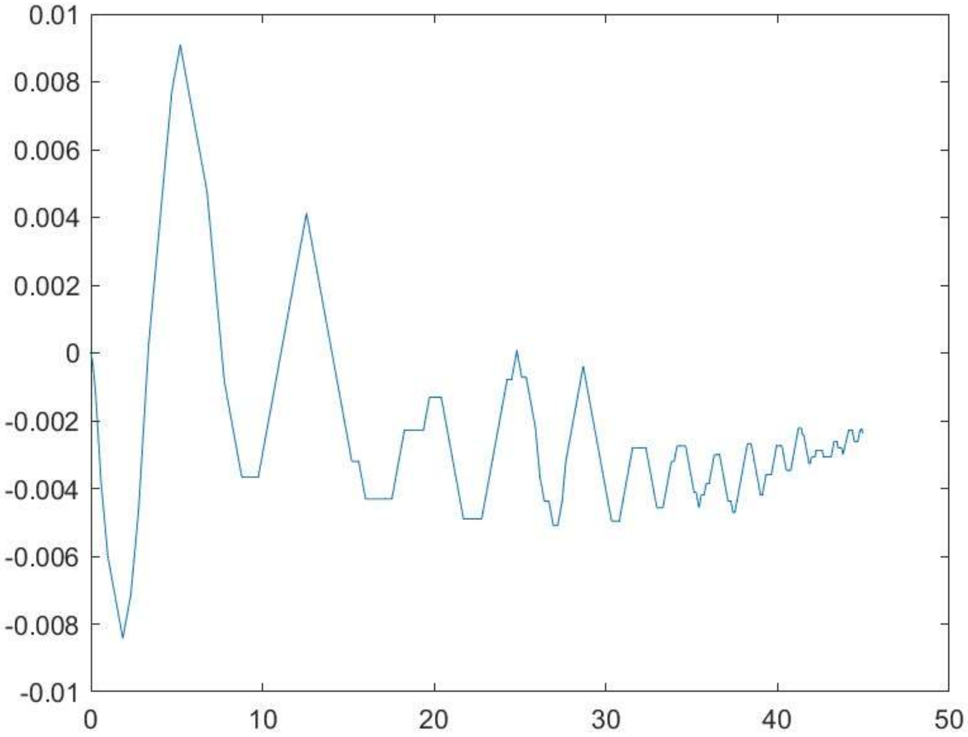
Non-Linear Controller



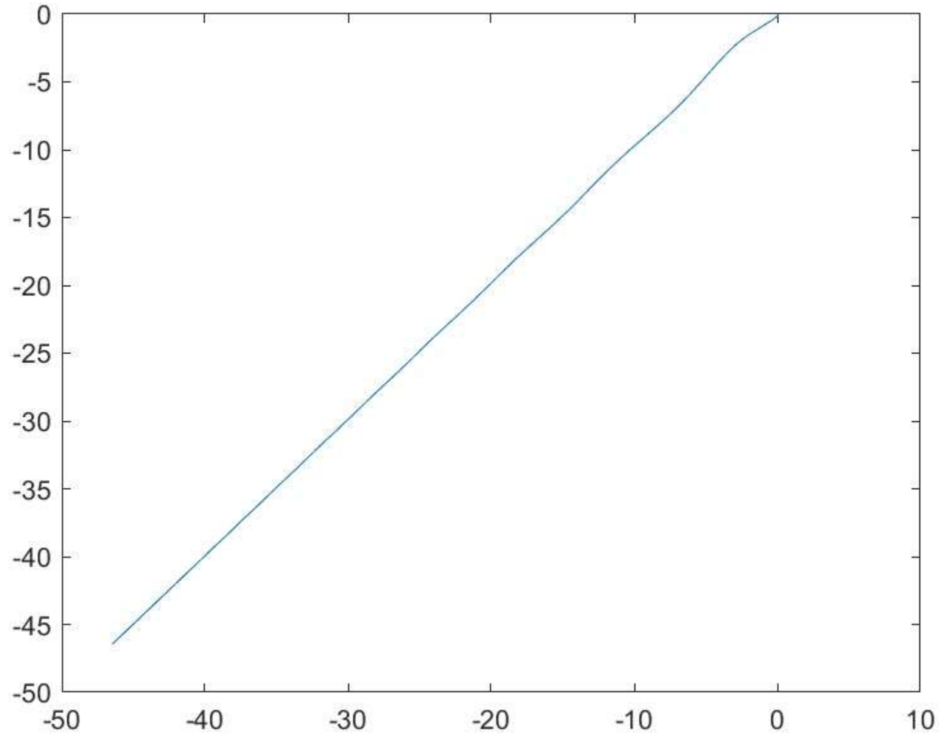



0.45
0.4
0.35
0.3
0.25
0.2
0.15
0.1
0.05
0
-0.05
-50
-40
-30
-20
-10
0
10

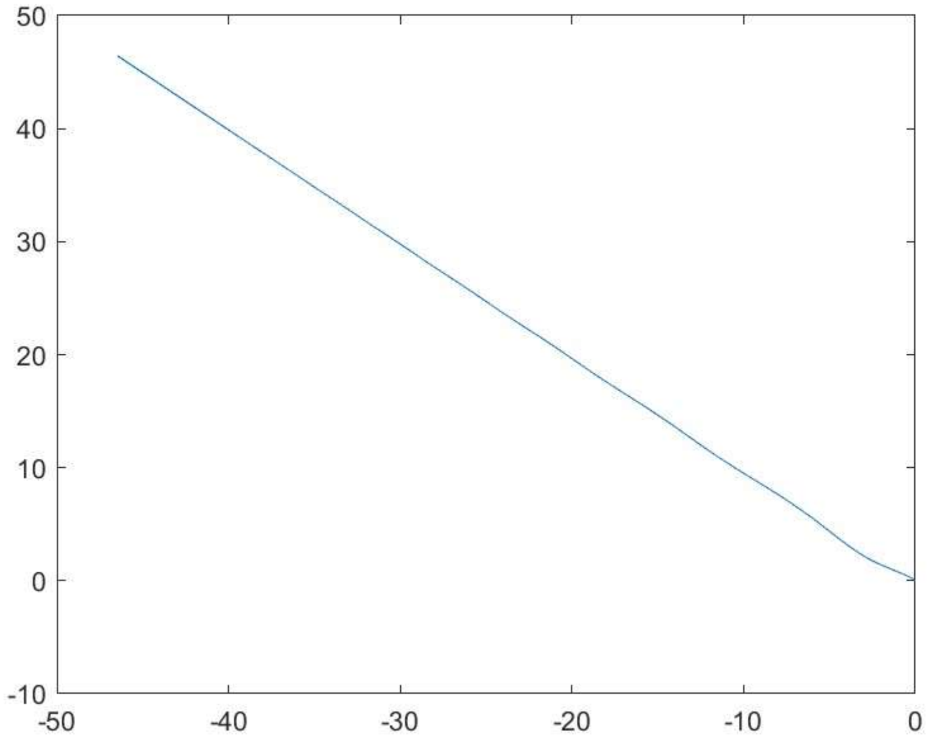

0
-5
-10
-15
-20
-25
-30
-35
-40
-45
-50
-50
-40
-30
-20
-10
0
10

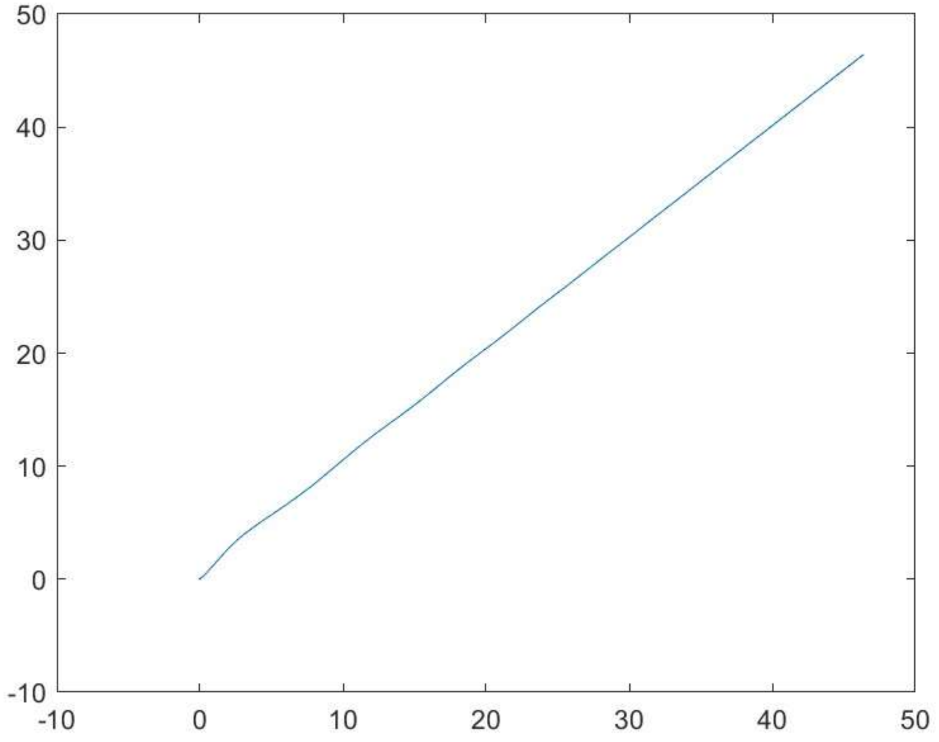


0.01
0.008
0.006
0.004
0.002
0
-0.002
-0.004
-0.006
-0.008
-0.01
0
10
20
30
40
50

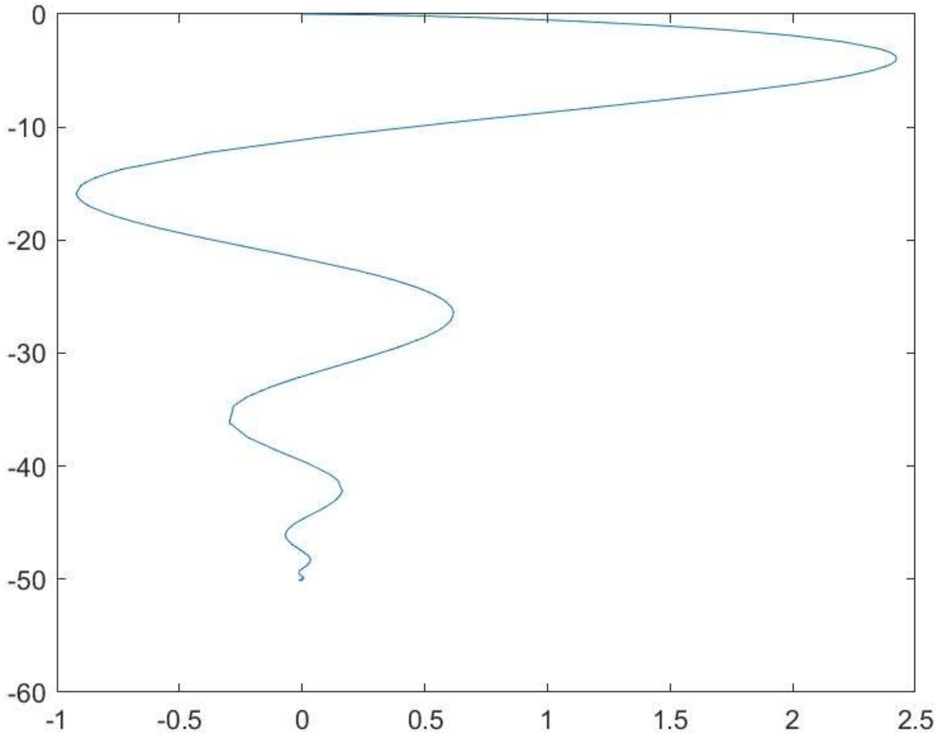


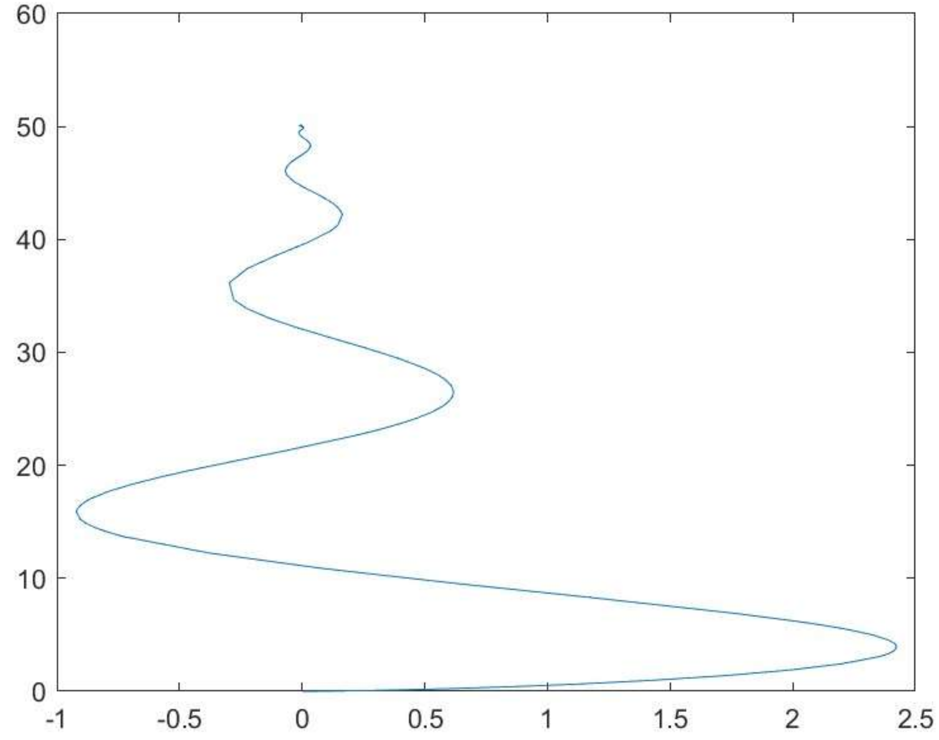
0
-5
-10
-15
-20
-25
-30
-35
-40
-45
-50
-50
-40
-30
-20
-10
0
10

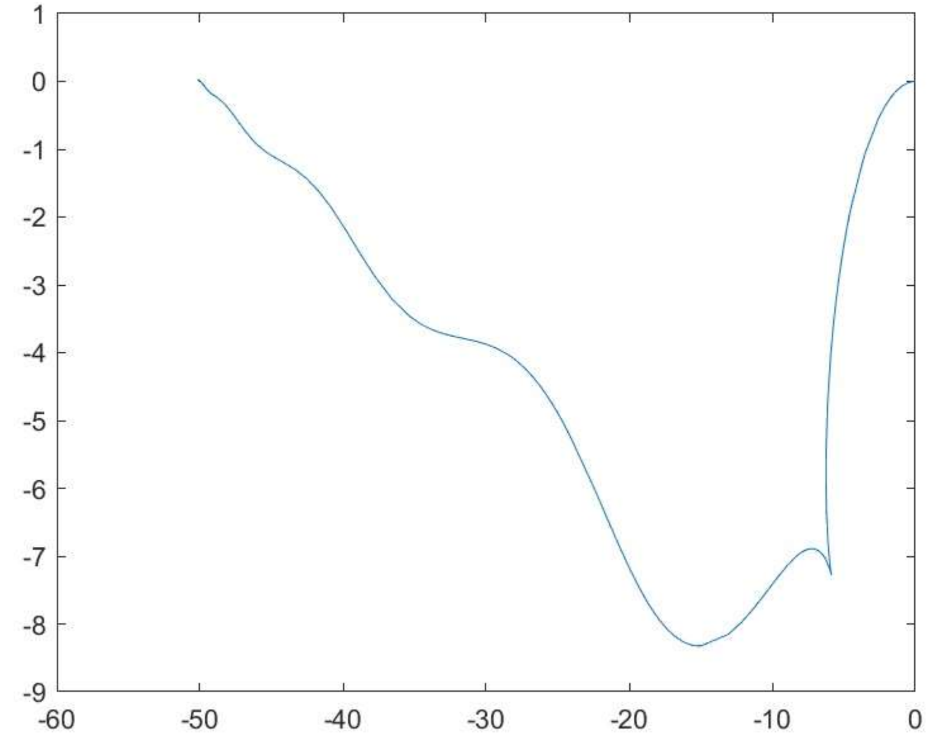

50
40
30
20
10
0
-10
-50
-40
-30
-20
-10
0

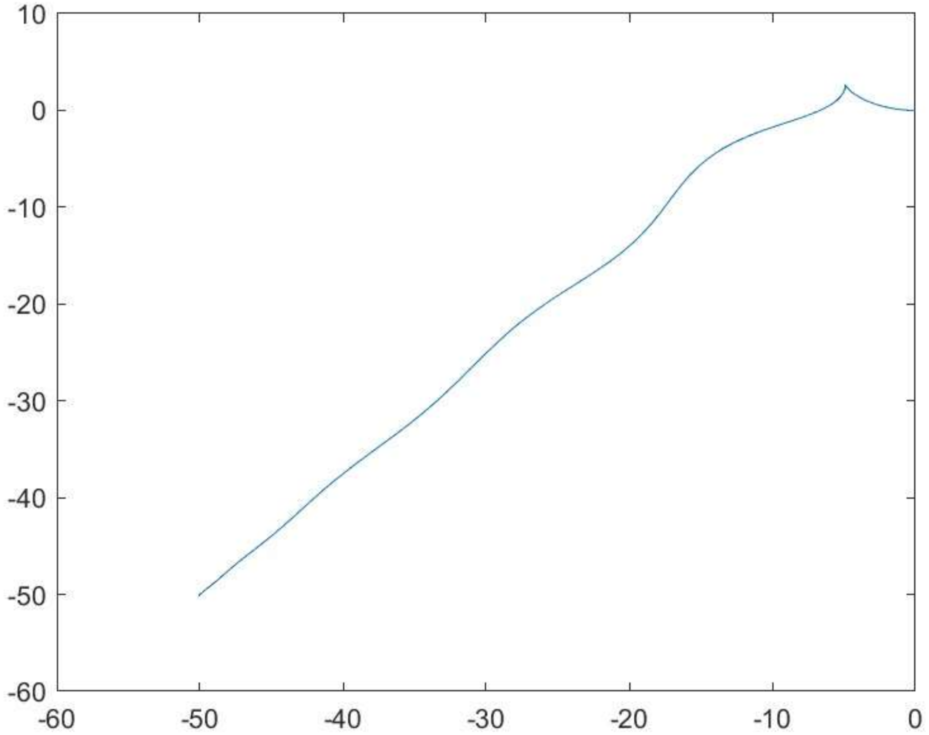

50
40
30
20
10
0
-10
-10
0
10
20
30
40
50

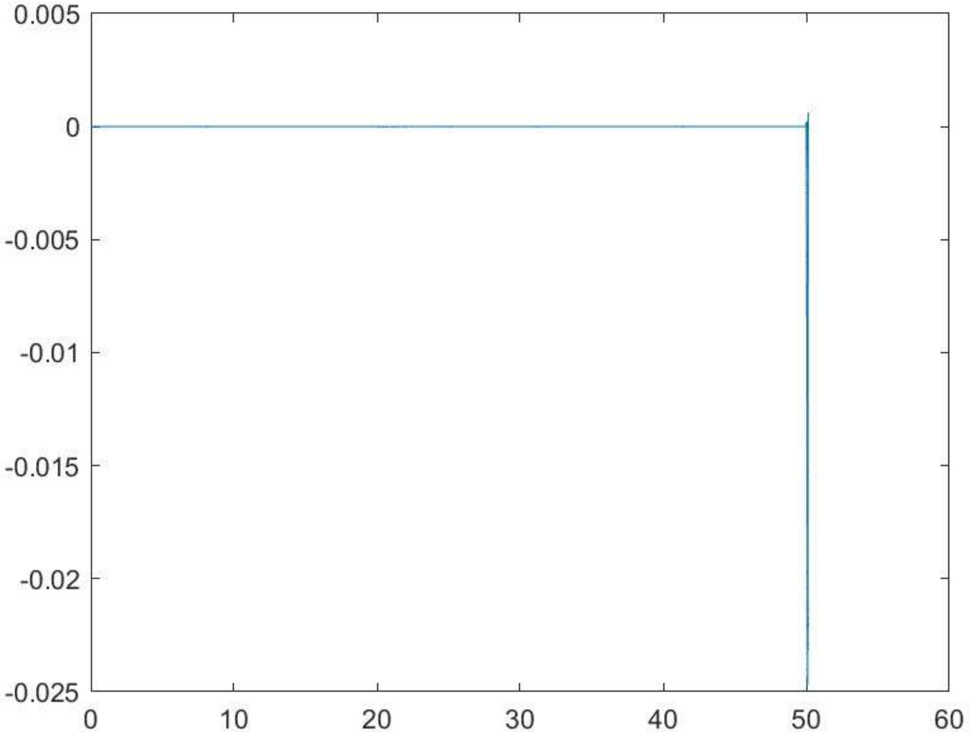
Non-Linear Controller From Simulink

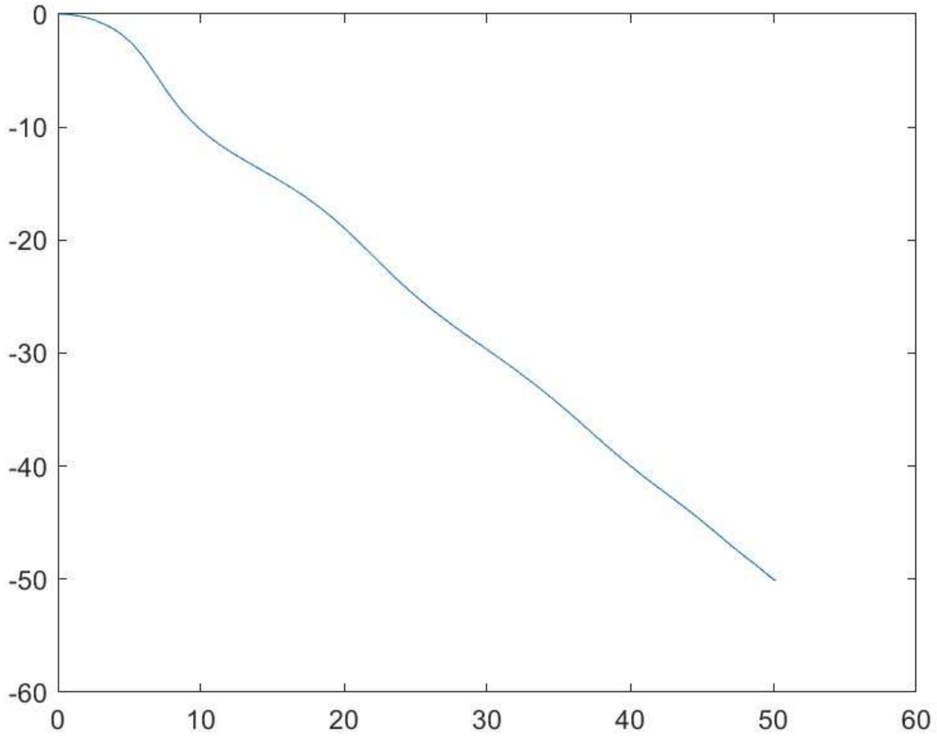


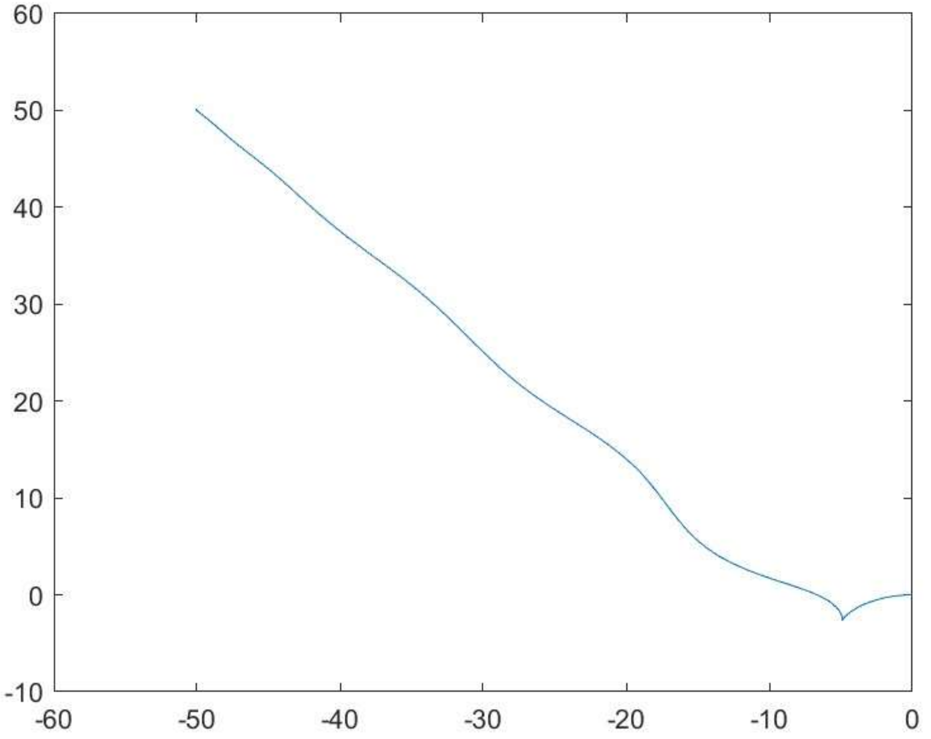


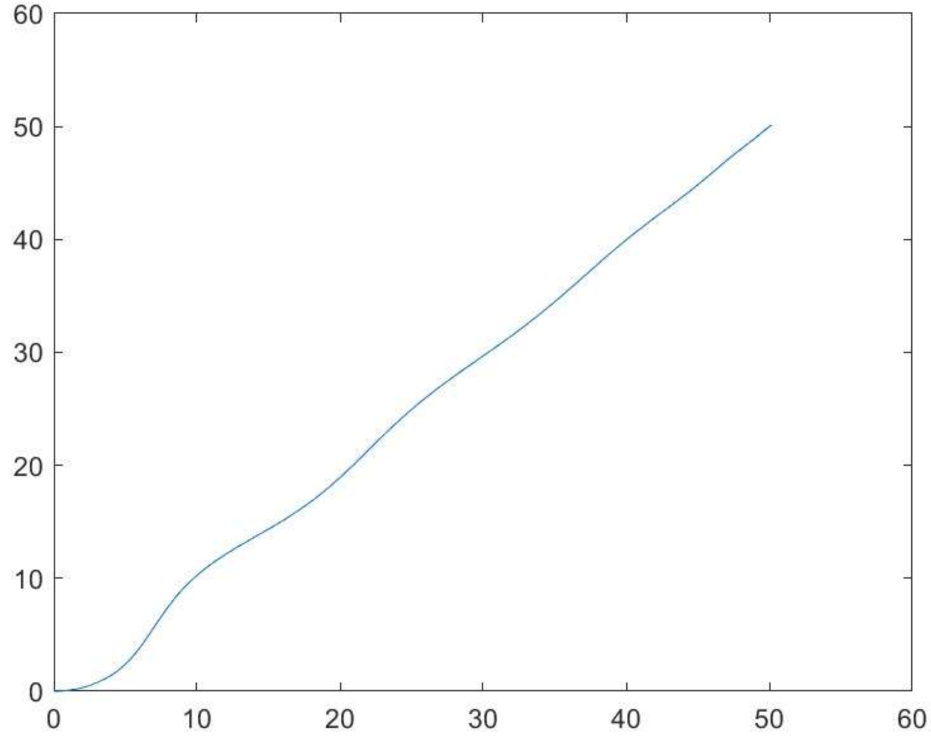

1
0
-1
-2
-3
-4
-5
-6
-7
-8
-9
-60
-50
-40
-30
-20
-10
0


10
0
-10
-20
-30
-40
-50
-60
-60
-50
-40
-30
-20
-10
0


0.005
0
-0.005
-0.01
-0.015
-0.02
-0.025
0
10
20
30
40
50
60


0
-10
-20
-30
-40
-50
-60
0
10
20
30
40
50
60


60
50
40
30
20
10
0
-10
-60
-50
-40
-30
-20
-10
0


60
50
40
30
20
10
0
0
10
20
30
40
50
60