# Exercise 2C 🡪 equilibrium values

x1e = [0, 5/9]

x2e = [0, 5/8]

code:

%% Problem 2C

syms x1 x2

x1dot = 2\*x2\*(1-x1) - x1;

x2dot = 3\*x1\*(1-x2) - x2;

%solve equilibrium values

sol = solve(x1dot == 0, x2dot == 0);

x1e = (sol.x1)

x2e = (sol.x2)

# Exercise 2C 🡪 Linearization

A = [ - 2\*x2 - 1, 2 - 2\*x1]

[ 3 - 3\*x2, - 3\*x1 - 1]

A (about [5/9,5/9]): [ -9/4, 8/9]

[ 9/8, -8/3]

A (about [0,0]): [ -1, 2]

[ 3, -1]

Code:

%linearize

A = [diff(x1dot,x1), diff(x1dot,x2);...

diff(x2dot,x1), diff(x2dot,x2)]

%substitute

subs(A,{x1,x2},{double(x1e(2)),double(x2e(2))})

subs(A,{x1,x2},{double(x1e(1)),double(x2e(1))})

# Exercise 4 Part (a-c)

A (about [0,0]):

[0, 1]

[ -(g\*m - k\*w\*(w^2)^(1/2))/(L\*m), -(k\*(w^2)^(1/2))/m]

A (about [pi, 0]):

[0, 1]

[ (g\*m - k\*w\*(w^2)^(1/2))/(L\*m), -(k\*(w^2)^(1/2))/m]

Code:

%% Problem 4 - analytical solution

syms x1 x2 L k w m g

x1dot = x2;

x2dot = (-1/(m\*L))\*(k\*(L\*x2-w\*sin(x1))\*sqrt(L^2\*x2^2+w^2-2\*L\*w\*sin(x1)\*x2)+m\*g\*sin(x1));

sol = solve(x1dot == 0, x2dot == 0);

%solve equilibrium values

x1e = (sol.x1)

x2e = (sol.x2)

%linearize

A = [diff(x1dot,x1), diff(x1dot,x2);...

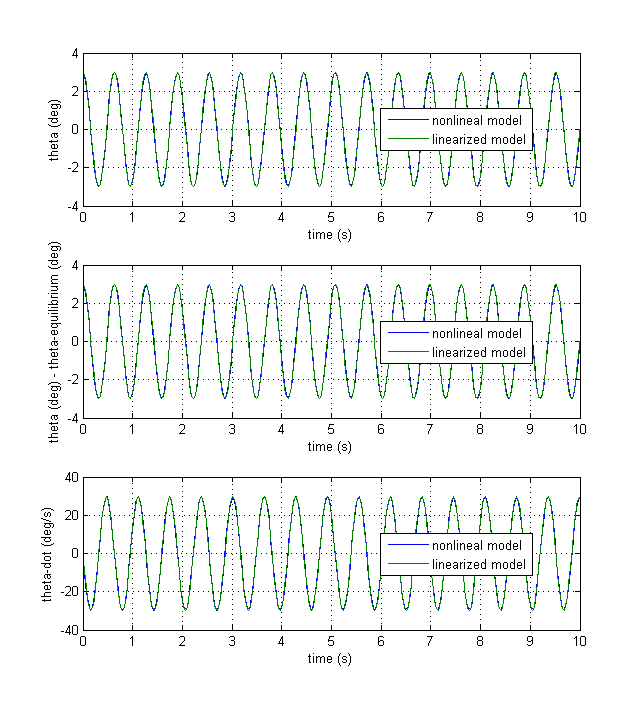
diff(x2dot,x1), diff(x2dot,x2)]

%substitute

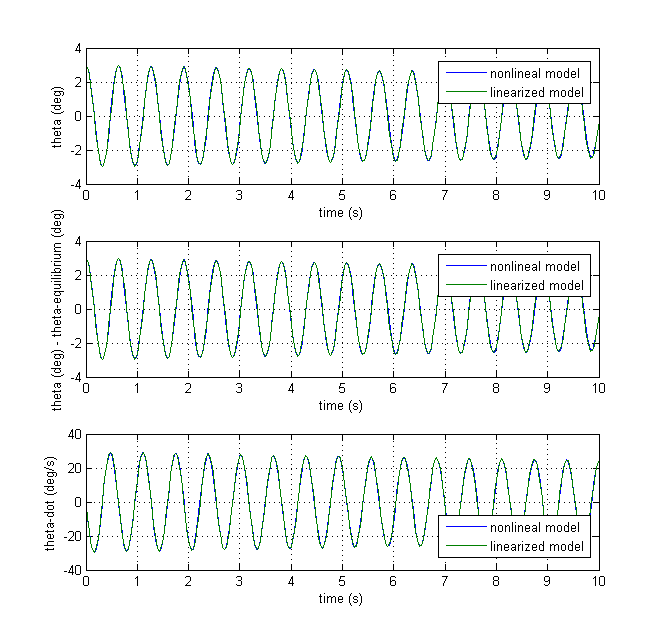
subs(A,{x1,x2},{0,0})

subs(A,{x1,x2},{pi,0})

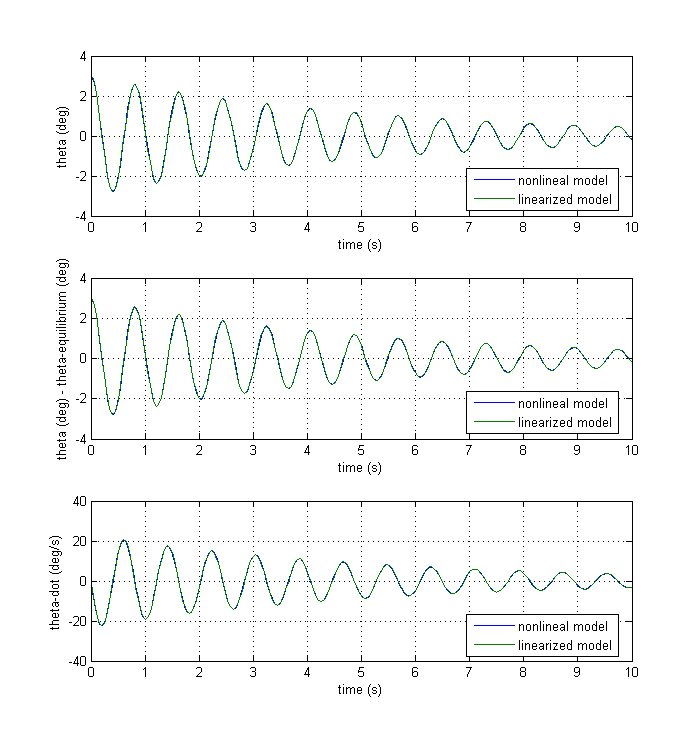
# Exercise 4 Part (d)



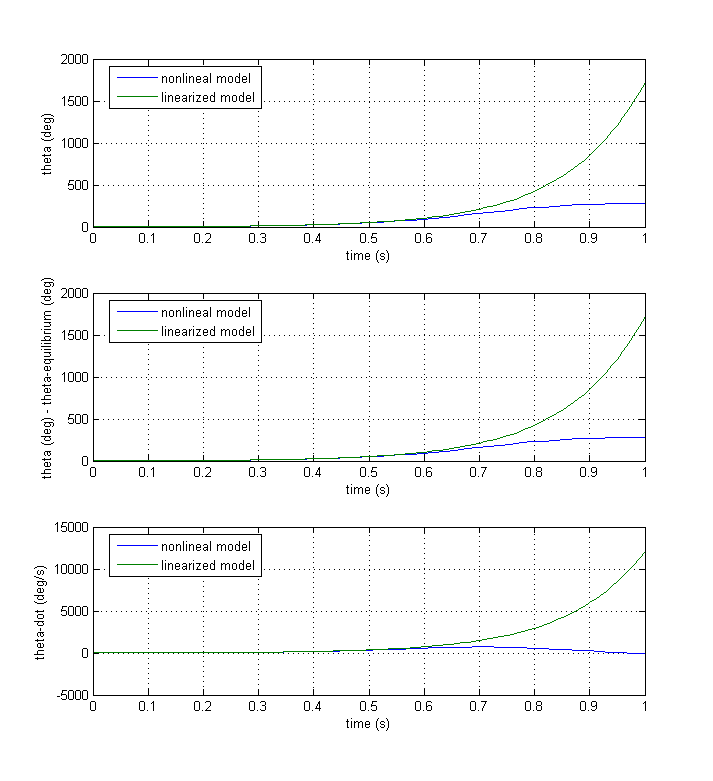
We observe that the solution is perfectly oscillatory without any damping. The linear solution agrees closely with the solution.



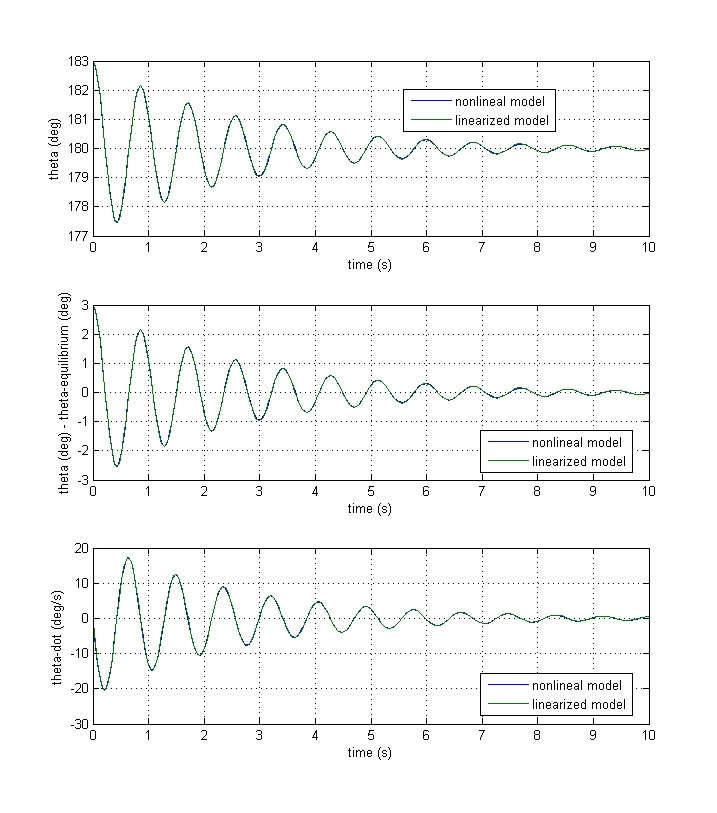
We see that the angle slowly damps due to the low wind. Linear model still very accurate



We see that wind more heavily dampens the angle over time. Trim angle is still 0 degrees. Linear model is still very accurate.



We see that the linear solution diverges from the nonlinear solution. Due to high wind, the nonlinear solution shows a trim angle of 180 degrees. Even though the pendulum starts at 3 degrees, the wind pushes the pendulum to an equilibrium value of 180 degrees. However, we specified for the use of the linear model that assumed a trim angle of 0 degrees. We should be using the linear model that assumes a trim angle of 180 degrees.



We see that the linear model now agrees with the nonlinear model once we specify the correct equilibrium angle. The wind is strong enough in this scenario to push the pendulum to an equilibrium of 180 degrees

Code:

%% Problem 4 - SIMULINK solution

clear all

L = 0.1; %m

m = 0.01; %kg

g = 9.81; %m/s

CD = 0.2;

S = 0.01; %m^2

rho = 0.3809; %kg/m^3

k = rho\*S\*CD/2;

w = 20; %m/s

n = 1;

theta\_trim = pi\*n;

delta\_theta0 = 3\*0.0174532925; %radians

theta0 = theta\_trim + delta\_theta0;

theta\_dot0 = 0;

t\_stop = 10;%s

%use simulink model to determine trim

[Xe]=trim('hw4\_4model')

%run nonlinear simulink model

sim('hw4\_4model');

t\_nonlin = nonlinear\_simout.Time; %s

theta\_nonlin = nonlinear\_simout.Data(:,1);

theta\_dot\_nonlin = nonlinear\_simout.Data(:,2);

%linearize simulink model

%[A, B, C, D] = linmod('hw4\_4model');

%sys = ss(A,B,C,D);

A = [0, 1;...

(-1)^n\*(1/(m\*L))\*(k\*w^2-m\*g), -k\*w/m]

sys = ss(A,[0,0]',[1,0],0);

[~,t\_lin,X] = initial(sys,[delta\_theta0, theta\_dot0],t\_stop);

delta\_theta\_lin = X(:,1);

theta\_dot\_lin = X(:,2);

theta\_lin = delta\_theta\_lin + theta\_trim;

%plot

figure(1)

subplot(3,1,1)

plot(t\_nonlin,theta\_nonlin\*57.2957795,t\_lin,theta\_lin\*57.2957795)

xlabel('time (s)')

ylabel('theta (deg)')

grid on

legend('nonlineal model','linearized model',0')

subplot(3,1,2)

plot(t\_nonlin,(theta\_nonlin-theta\_trim)\*57.2957795,t\_lin,(theta\_lin-theta\_trim)\*57.2957795)

xlabel('time (s)')

ylabel('theta (deg) - theta-equilibrium (deg)')

grid on

legend('nonlineal model','linearized model',0')

subplot(3,1,3)

plot(t\_nonlin,theta\_dot\_nonlin\*57.2957795,t\_lin,theta\_dot\_lin\*57.2957795)

xlabel('time (s)')

ylabel('theta-dot (deg/s)')

grid on

legend('nonlineal model','linearized model',0')

%% analytic solution

syms x1 x2

x1dot = x2;

x2dot = (-1/(m\*L))\*(k\*(L\*x2-w\*sin(x1))\*sqrt(L^2\*x2^2+w^2-2\*L\*w\*sin(x1)\*x2)+m\*g\*sin(x1));

sol = solve(x1dot == 0, x2dot == 0);

%solve equilibrium values

x1e = (sol.x1);

x2e = (sol.x2);

%linearize

A = [diff(x1dot,x1), diff(x1dot,x2);...

diff(x2dot,x1), diff(x2dot,x2)];

double(subs(A,{x1,x2},{pi,0}))