# Table of Contents

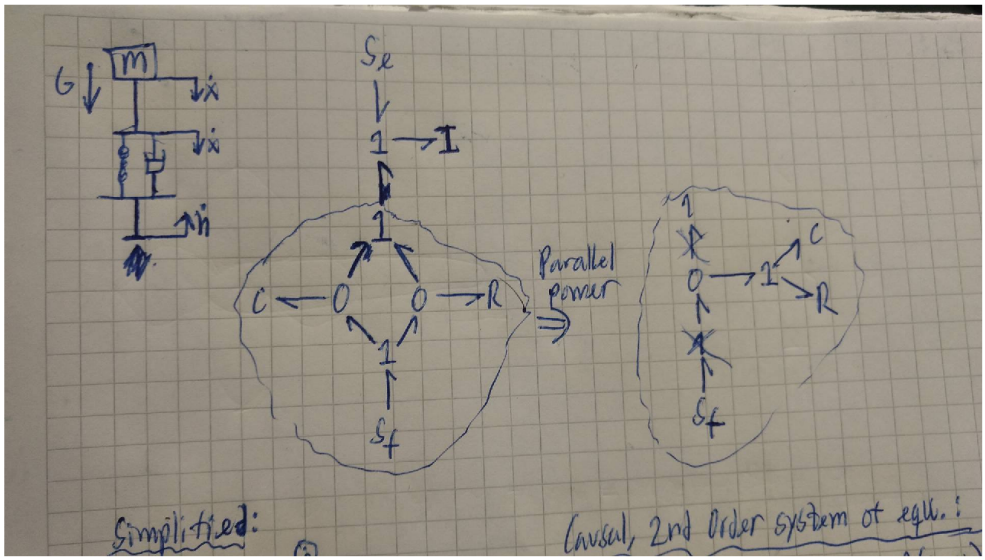


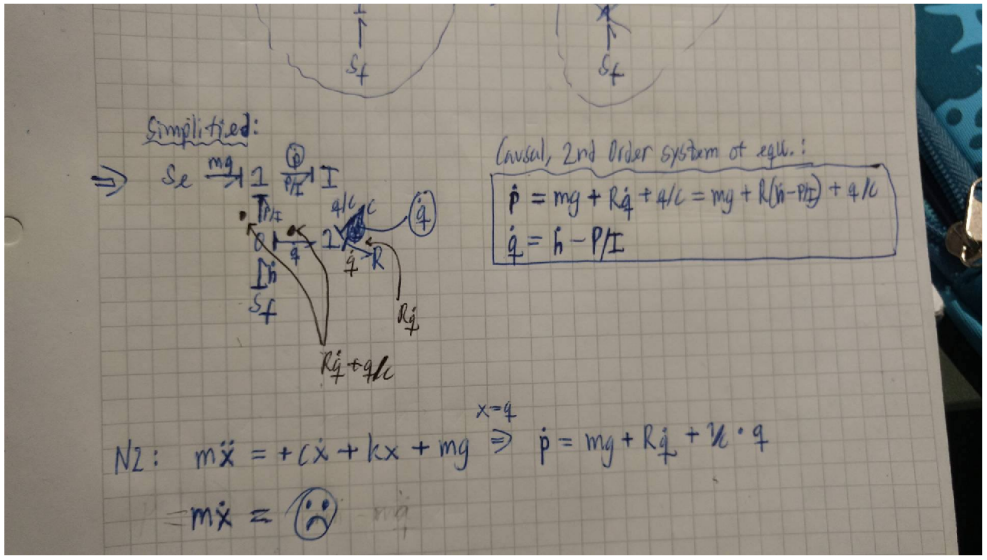
```
clear all
clc
```

# Bond Graph

```
img = imread('2.jpg');
img2 = imread('1.jpg');
figure, imshow(img);
figure, imshow(img2);
```

```
Warning: Image is too big to fit on screen; displaying at 67%
Warning: Image is too big to fit on screen; displaying at 67%
```






# Constants

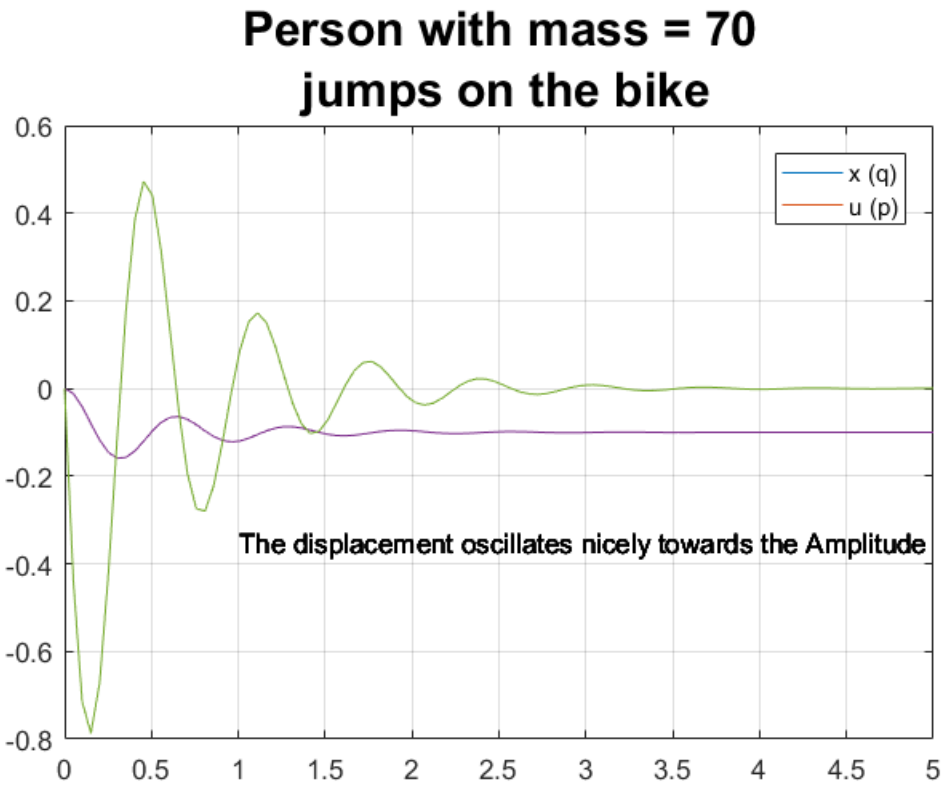
```
global h_max h_omega I C R g

I = 70;
g = -9.81;
Amptude = .1;
k = I*abs(g)/Amptude;
C= 1/k;

omega = sqrt(k/I);
period = 2*pi/omega;
R = I*omega/pi;          %R chosen such that the amplitude is reduced
 by
                            %e^-1 after t = T = 2*pi/omega
tmax = 5;

fprintf(' spring constant k = %i\n', k)
fprintf(' natural frequency omega = %i\n', omega)
fprintf(' one period = %f\n', period)
fprintf(' damping constant R = %i\n', R)
```

```
 spring constant k = 6867
 natural frequency omega = 9.904544e+00
 one period = 0.634374
 damping constant R = 2.206900e+02
```

# First Simulation: Finding constants

```
%A person with mass m jumps on the bike at time t=0
%Expected behavoiur: The displacement u should level off after a while
%--------------------------------------------------------------------------
```

```
h_max = 0;
h_omega = 0;
tspan = linspace(0, tmax);

%initial values
x_0 = 0;        %displacement
u_0 = 0;        %velocity

[t, y] = ode45(@eulerStep, tspan, [x_0, u_0]);

figure(1)
plot(t, x_0 - y(:,1), t, y(:,2)/I)
legend('x (q)','u (p)')
title(sprintf('Person with mass = %i\n jumps on the bike',
 I),'FontSize',18);
text(1.0, -0.35,'The displacement oscillates nicely towards the
 Amplitude')
grid on
hold on
```

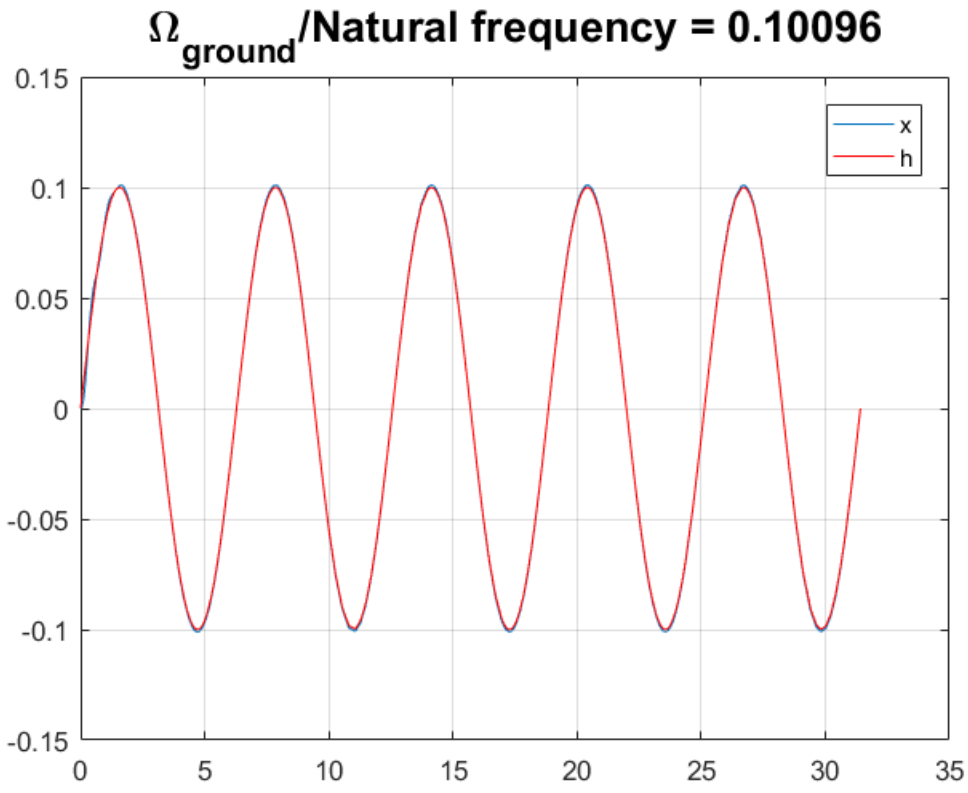


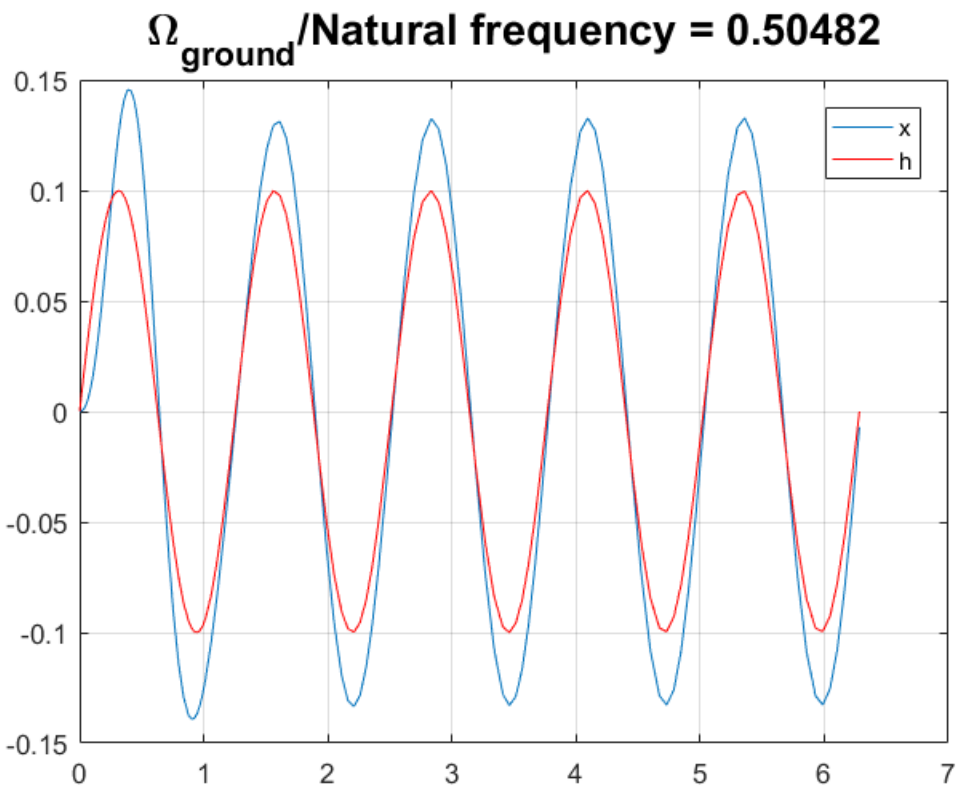

# Second simulation

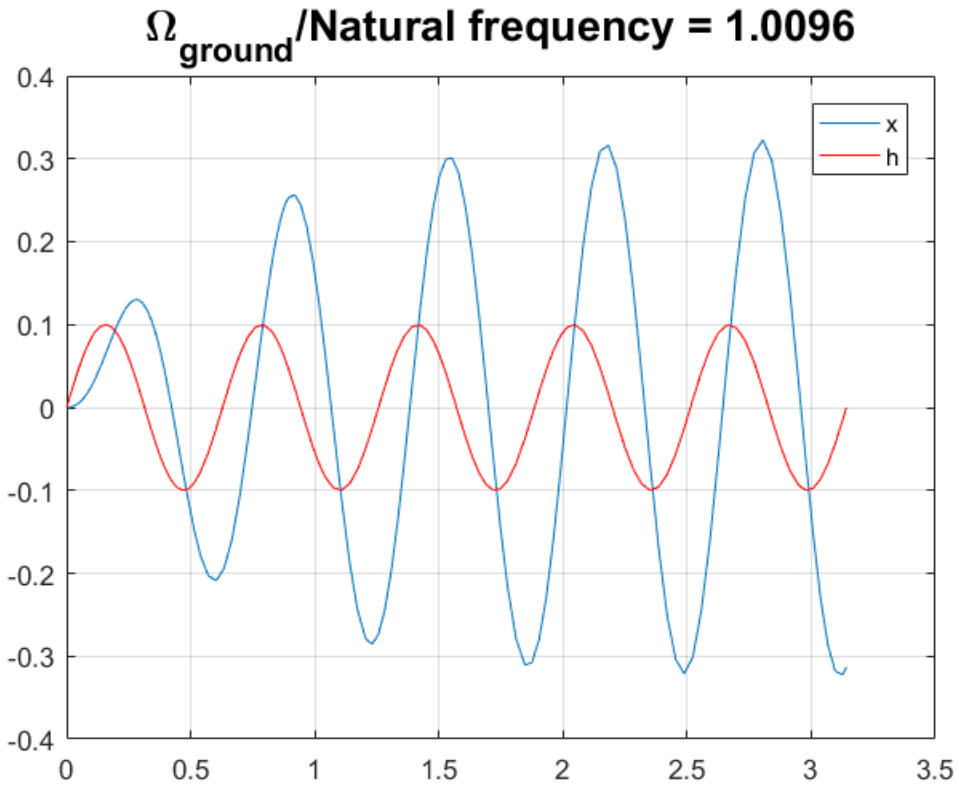
Simulates the vertical motion of when he's cycling at constant speed and the ground level changes as h(y)

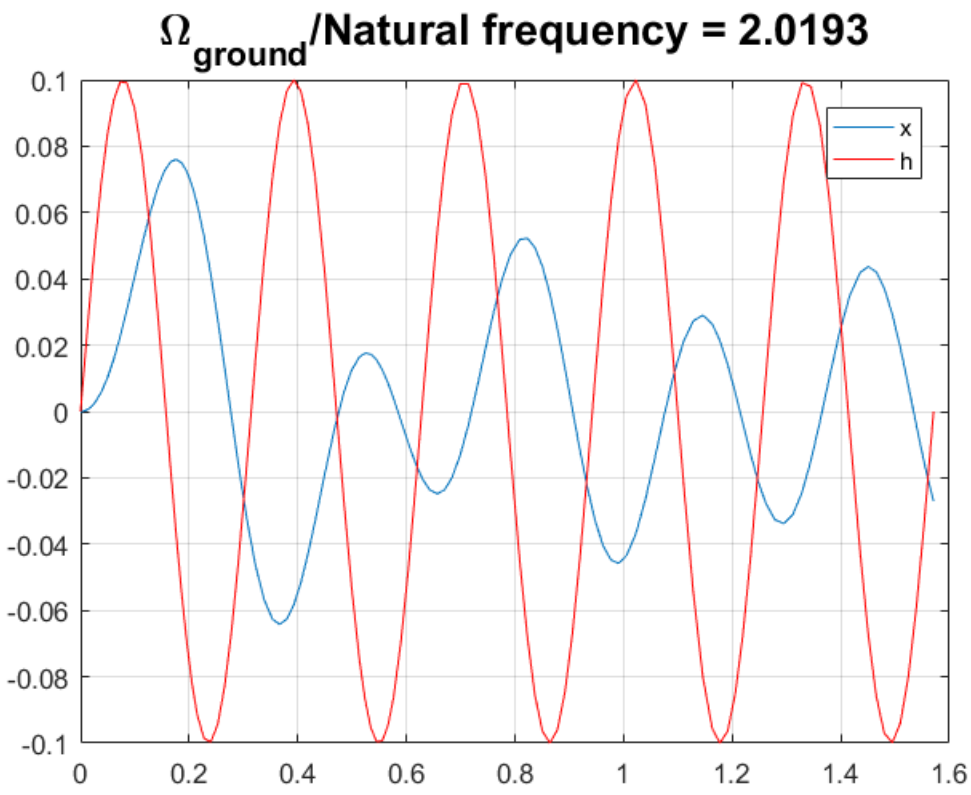
```
%-----------------------------------------------------------------------
```

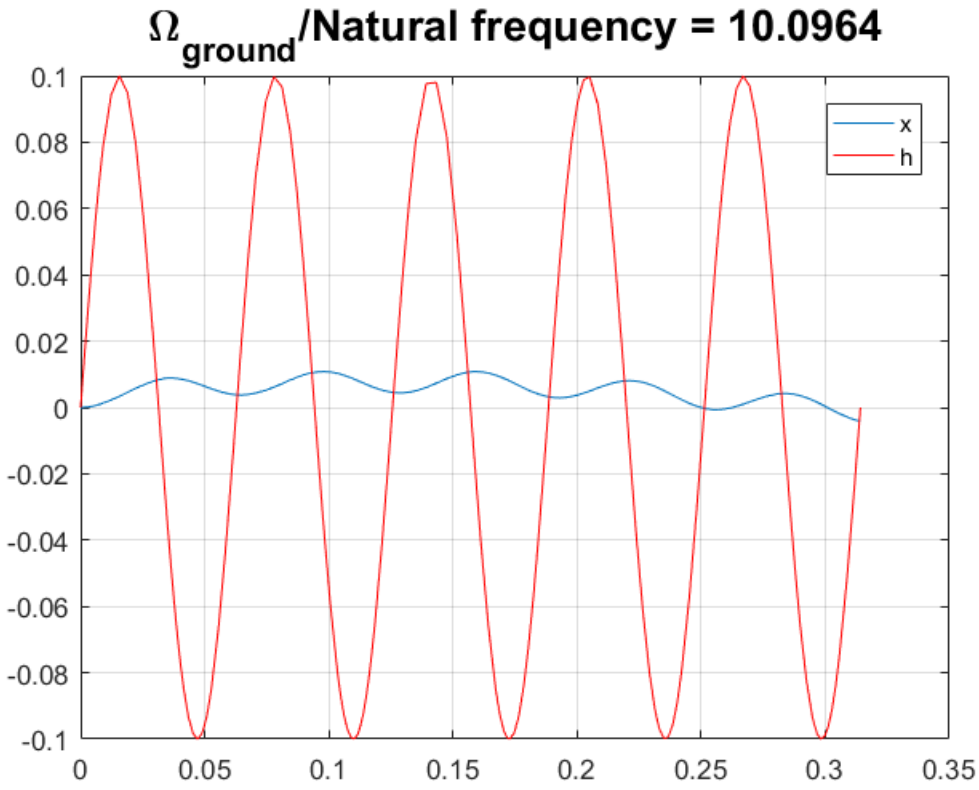
```
%initial values
x_0 = y(end, 1); %displacement
u_0 = 0;         %velocity

h_max = Amptude;
h_0 = [1 5 10 20 100];
k = 1;
for h_omega = h_0
    k = k + 1;
    tmax = 10*pi/h_omega;
    [t, y] = ode15s(@eulerStep, [0 tmax], [x_0, u_0]);
    figure(k)
    plot(t,x_0-
y(:,1)+h_max*sin(t.*h_omega),t,h_max*sin(t.*h_omega),'r')
    legend('x','h')
    title(['\Omega_{ground}/Natural frequency = ',num2str(h_omega/
omega)],'FontSize',16);
    grid on
    drawnow
    hold off
end
```




$\Omega_{ground}$/Natural frequency = 0.50482
x
h
0.15
0.1
0.05
0
-0.05
-0.1
-0.15
0
1
2
3
4
5
6
7

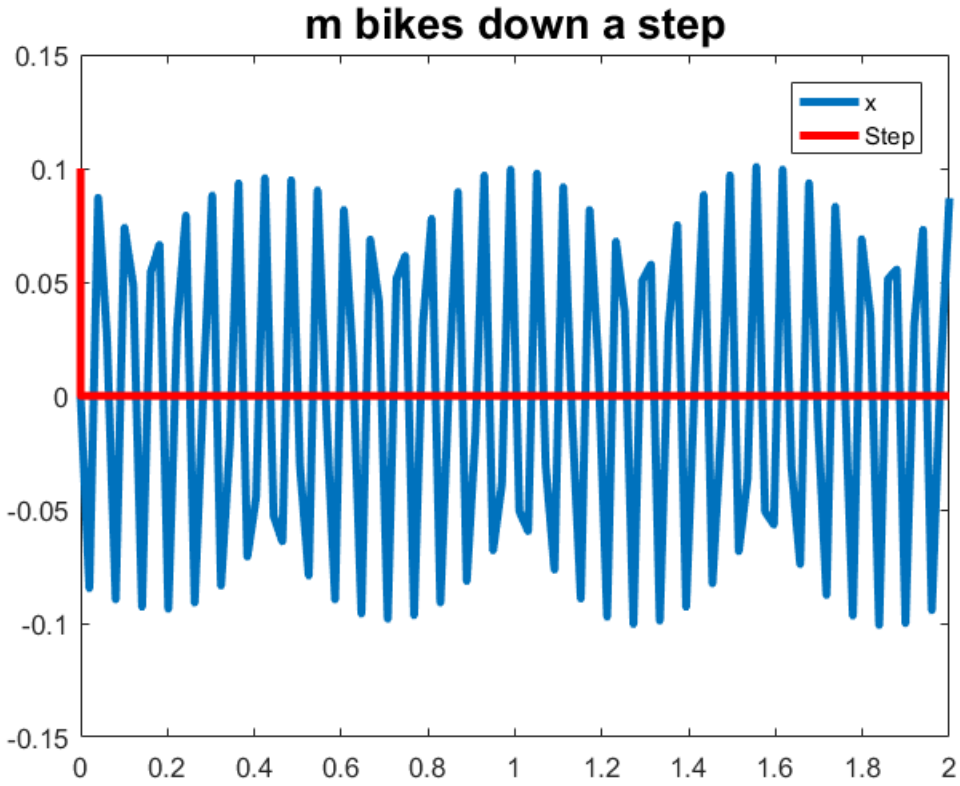

$\Omega_{ground}$/Natural frequency = 1.0096
x
h
0.4
0.3
0.2
0.1
0
-0.1
-0.2
-0.3
-0.4
0
0.5
1
1.5
2
2.5
3
3.5


$\Omega_{ground}$/Natural frequency = 2.0193
x
h
0.1
0.08
0.06
0.04
0.02
0
-0.02
-0.04
-0.06
-0.08
-0.1
0
0.2
0.4
0.6
0.8
1
1.2
1.4
1.6


$\Omega_{ground}$/Natural frequency = 10.0964
x
h
0.1
0.08
0.06
0.04
0.02
0
-0.02
-0.04
-0.06
-0.08
-0.1
0
0.05
0.1
0.15
0.2
0.25
0.3
0.35

# Third simulation:

```
%-------------------------------------------------------------------------

x_0 = Amptude;    % Step height
u_0 = 0;
hmax = 0;
tmax = 2;
tspan = linspace(0, tmax);

[t,y]=ode45(@eulerStep,tspan,[x_0; u_0]);

% Plotting: The biker falls 50% lower than the step!
k = k+1;
figure(k)
plot(t,x_0-y(:,1),[0 0 tmax],[x_0 0 0],'r','LineWidth',3)
legend('x','Step')
title('m bikes down a step','FontSize',16);
```




# Function file:

```
function d_dt = eulerStep(t, y)
global h_max h_omega I C R g
```

```
h_dot = h_max*h_omega*cos(h_omega*t);
d_dt=zeros(2,1);

q = y(1);
p = y(2);

q_dot = h_dot - p/I;
d_dt(1) = q_dot;
d_dt(2) = I*g + R*q_dot + q/C;
```

*Published with MATLAB® R2016b*