Math 568 Homework 8
Due 03/08/23
By Marvyn Bailly

Problem 1 *Consider the inverted pendulum dynamics:*

$$y'' + (\delta + \epsilon \cos(\omega t)) \sin(y) = 0.$$

(a) Perform a Floquet analysis of the pendulum with continuous forcing $\cos(\omega t)$.

(b) Evaluate for what values of $\delta, \epsilon,$ *and* ω *the pendulum is stabilized.*

Solution.

(a) Consider the pendulum dynamics given by,

$$y'' + (\delta + \epsilon \cos(\omega t)) \sin(y) = 0.$$

When the pendulum is in the upright position, we can use Taylor expansion on the $\sin(x)$ to approximate the dynamics of the linear pendulum. In the inverted position (when $x = \pi$), we have the equation

$$y'' - (\delta + \epsilon \cos(\omega t))y = 0,$$

and in the downward (when $x = 0$) position by

$$y'' + (\delta + \epsilon \cos(\omega t))y = 0.$$

To preform Floquet analysis, recall that the Floquet discriminant is of the form

$$\Gamma = x_1\left(\frac{2\pi}{\omega}\right) + x_2'\left(\frac{2\pi}{\omega}\right).$$

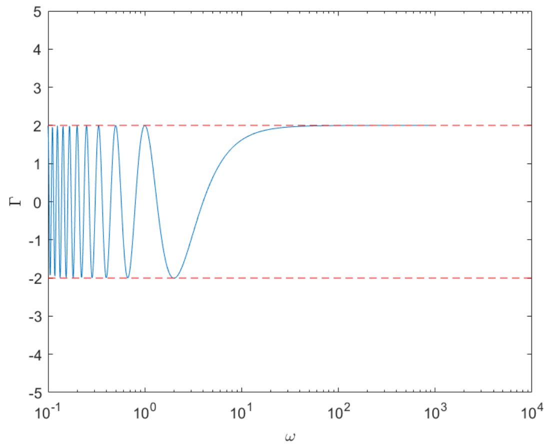
where the solutions $x_1(t)$ and $x_2(t)$ satisfy the initial conditions

$$\begin{aligned} y_1(0) &= 1, \text{ and } y_1'(0) = 0, \\ y_2(0) &= 0, \text{ and } y_2'(0) = 1. \end{aligned}$$

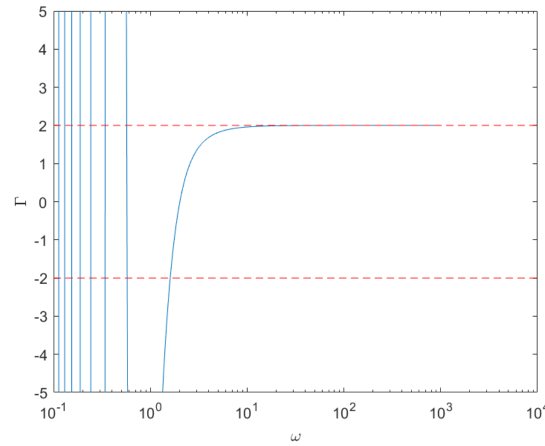
If we let $v = y'$ and $u = y$, then we have the following system of equations

$$\begin{aligned} u' &= \mp(\delta + \epsilon \cos(\omega t))v, \\ v' &= u, \end{aligned}$$

which we can analyze in MATLAB. Using the code in Listings 1, we can generate the following plots:

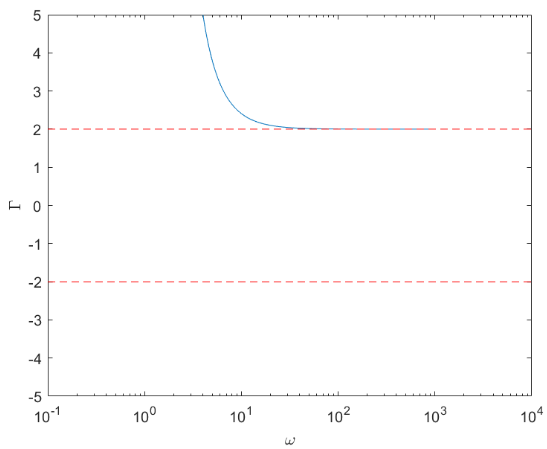


(a) Plot of downward pendulum using $\delta > \epsilon$.

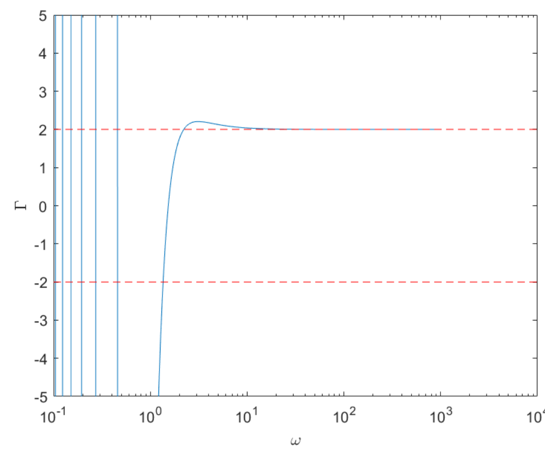


(b) Plot of downward pendulum using $\delta < \epsilon$

Figure 1: Plots of Γ (blue solid line) for different ω values in the downward pendulum. We also plot the stability boundaries where $\Gamma = \pm 2$ in the dotted red lines. In (a) we use $\delta = 1$ and $\epsilon = 0.1$ and (b) we use $\delta = 0.1$ and $\epsilon = 1$.



(a) Plot of upward pendulum using $\delta > \epsilon$.



(b) Plot of upward pendulum using $\delta < \epsilon$

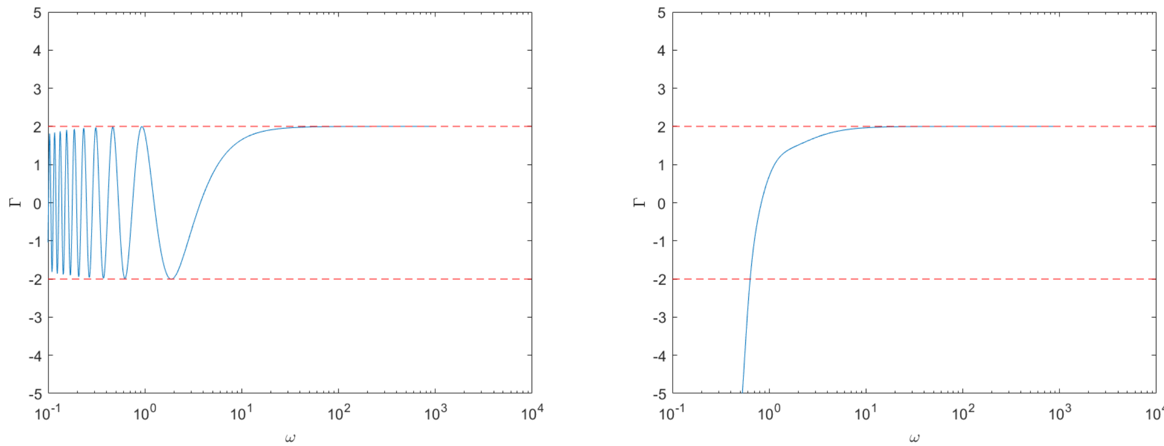
Figure 2: Plots of Γ (blue solid line) for different ω values in the upward pendulum. We also plot the stability boundaries where $\Gamma = \pm 2$ in the dotted red lines. In (a) we use $\delta = 1$ and $\epsilon = 0.1$ and (b) we use $\delta = 0.1$ and $\epsilon = 1$.

For the nonlinear case, we follow a similar process but now using the equations

$$u' = \mp(\delta + \epsilon \cos(\omega t)) \sin(v),$$

$$v' = u.$$

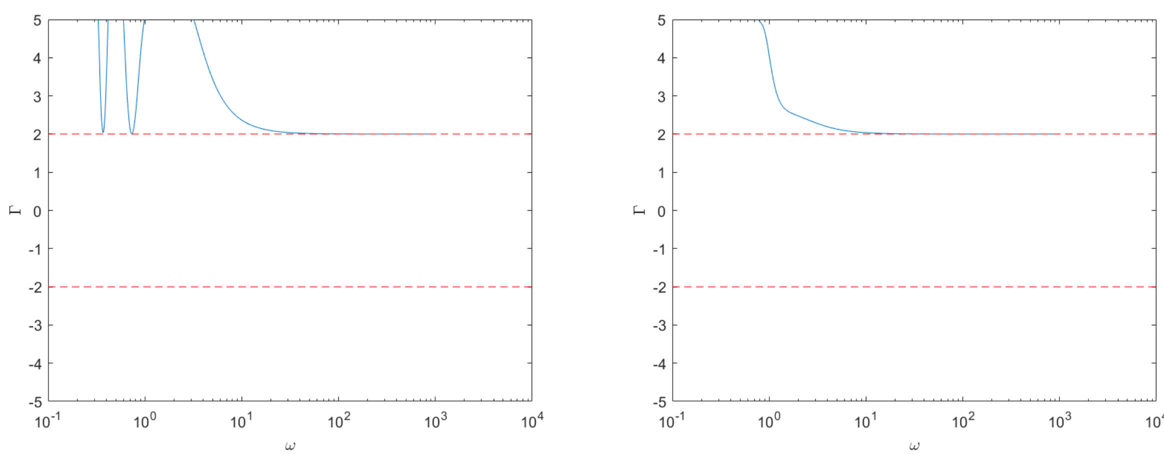
Using MATLAB, we can generate the following plots:



(a) Plot of downward pendulum using $\delta > \epsilon$.

(b) Plot of downward pendulum using $\delta < \epsilon$

Figure 3: Plots of Γ (blue solid line) for different ω values in the downward pendulum. We also plot the stability boundaries where $\Gamma = \pm 2$ in the dotted red lines. In (a) we use $\delta = 1$ and $\epsilon = 0.1$ and (b) we use $\delta = 0.1$ and $\epsilon = 0.2$.



(a) Plot of upward pendulum using $\delta > \epsilon$.

(b) Plot of upward pendulum using $\delta < \epsilon$

Figure 4: Plots of Γ (blue solid line) for different ω values in the upward pendulum. We also plot the stability boundaries where $\Gamma = \pm 2$ in the dotted red lines. In (a) we use $\delta = 1$ and $\epsilon = 0.1$ and (b) we use $\delta = 0.1$ and $\epsilon = 1$.

(b) To find where the pendulum is stable, we can look at the Figures 1,2,3, and 4 and see when Γ is less than $|2|$.

Listing 1: **MATLAB code used to generated plots.**

```
%Switch these to match which situation you are tring to simulate
%d = 1;
%e = .1;
d = .1;
e = .2;
w = 10e2;


wVals = 10.^linspace(-0.5,3,1000);
GVals = zeros(1,length(wVals));


for i = 1:length(wVals)
    w = wVals(i);
    T = 2*pi/w;
    tSpan = [0,T];

    y01 = [1,0];
    %--switch these to match which sitaution you are trying to simulate
    [t1,y1] = ode45(@(t,y) upNonLinear(t,y,d,e,w), tSpan, y01);
    %[t1,y1] = ode45(@(t,y) downNonLinear(t,y,d,e,w), tSpan, y01);
    %[t1,y1] = ode45(@(t,y) up(t,y,d,e,w), tSpan, y01);
    %[t1,y1] = ode45(@(t,y) down(t,y,d,e,w), tSpan, y01);


    %--switch these to match which sitaution you are trying to simulate
    y02 = [0,1];
    [t2,y2] = ode45(@(t,y) upNonLinear(t,y,d,e,w), tSpan, y02);
    %[t1,y1] = ode45(@(t,y) downNonLinear(t,y,d,e,w), tSpan, y01);
    %[t1,y1] = ode45(@(t,y) up(t,y,d,e,w), tSpan, y01);
    %[t1,y1] = ode45(@(t,y) down(t,y,d,e,w), tSpan, y01);


    GVals(i) = y1(end,1) + y2(end,2);
end

figure(1)
```

```
semilogx(wVals,GVals);
hold on
semilogx([10^(-1),10e3],[2,2],"--r")
semilogx([10^(-1),10e3],[-2,-2],"--r")
axis([10^(-1),10e3,-5,5])
xlabel('\omega')
ylabel('\Gamma')
hold off

function dydt = down(t,y,d,e,w)
    dydt = [y(2); -(d + e*cos(w*t))*y(1)];
end
function dydt = up(t,y,d,e,w)
    dydt = [y(2); (d+e*cos(w*t))*y(1)];
end
function dydt = downNonLinear(t,y,d,e,w)
    dydt = [y(2); -(d+e*cos(w*t))*sin(y(1))];
end
function dydt = upNonLinear(t,y,d,e,w)
    dydt = [y(2); (d+e*cos(w*t))*sin(y(1))];
end
```

$\square$