Homework cover page

Analytic and machine learning-based modeling of dynamical systems - 036064

Dr. Maor Farid

☒ Homework no. ___2____

☐ Final project

Students details:

Chenhao Yang	941170706
Full name	ID no.
Jonathan Oh	941170383
Full name	ID no.

Before submitting your homework, you are requested to fill in the suitable survey on the course website. Submitting the survey is mandatory. Your feedback is highly important for the course's staff and serves us for optimizing the course contents for you and for maximizing the contribution of the HW and assimilation of the material learned. Of course, your answers do not influence your grade and evaluation in any way.

☒ I confirm that I have filled in and submitted the HW survey on the course website

Remarks:

Our submission files contain several solution files (solution.js etc.), it accelerates your computing speed if you wish to run .ipynb, just put them in the same directory.

Thanks for your collaboration and good luck!

Question 1

(a) Modeling

Obtain the equation of motion (EOM), for linear spring and cubic damper, characterized by the following forces: $F_k = -ky$ and $F_c = cy'^3$, respectively. Here tag represents differentiation with respect to the dimensional time scale t . Both methods Newton and Lagrange can be applied. Assume that gravitational forces are neglectable with respect to the aforementioned stiffness and damping forces.

Using Newton's method:

$$-ky - cy'^3 = my''$$

(b) Normalize the EOM

Reorganize the function:

$$my'' + ky + cy'^3 = 0$$

Define the non-dimentional time as $\tau = \omega_k t$ and natural frequency of spring as $\omega_k = \sqrt{\frac{k}{m}}$. subs in:

$$m\omega_k^2 \ddot{y}_{\tau\tau} + ky + c\omega_k^3 \dot{y}_\tau^3 = 0$$

And we have:

$$\ddot{y}_{\tau\tau} + y + \frac{c}{m}\omega_k \dot{y}_\tau^3 = 0$$

Define non-dimensionlized length $x = \frac{y}{1[m]}$:

$$\ddot{x} + x + \bar{\mu}\dot{x}^3 = 0$$

where

$$\mu = \frac{c}{m}\omega_k = \frac{c}{m} \cdot \sqrt{\frac{k}{m}}$$

μ represents how dominant the damping effect is compared to the spring.

(c) Asymptotic method

We will derive an approximate solution using the asymptotic method of singular perturbation theory. We assume that the solution comprise of two parts. A first order solution x_0 and a small correction term $x_1, \epsilon << 1$. We assume the solution happens in different time scales to avoid secular term.

$$x(\tau) = x_0(\tau) + \epsilon x_1(\tau)$$

Define different time scales as following. T_0 is the regular time scale and T_1 is the slow time scale.

$$T_0 = \tau \tag{1}$$
$$T_1 = \epsilon\tau \tag{2}$$

For each time scales, define the differtial operator as such:

$$D_0 = \frac{\partial}{\partial T_0}$$
$$D_1 = \frac{\partial}{\partial T_1}$$

italicized text Then the original equation is turned into the following form. Note that higher order term $O(\epsilon^n), n \geq 2$ is omitted. Also, we substitude μ as $\epsilon\alpha$ since we assume the damping is small.

$$Eq \equiv (D_0^2 + 2\epsilon D_0 D_1)(x_0 + \epsilon x_1) + (x_0 + \epsilon x_1) + \epsilon\alpha[(D_0 + \epsilon D_1)(x_0 + \epsilon x_1)]^3$$

If we collect the same order term, we get following set of equations.

$$O(1) : D_0^2 x_0 + x_0 = 0 \tag{3}$$
$$O(\epsilon) : 2D_0 D_1 x_0 + D_0^2 x_1 + x_1 + \alpha(D_0 x_0)^3 = 0 \tag{4}$$

First, we solve for the $O(1)$ equation.

$$D_0^2 x_0 + x_0 = 0 \tag{5}$$
$$\Rightarrow x_0 = A(T_1)e^{iT_0} + \bar{A}(T_1)e^{-iT_0} \tag{6}$$
$$\Rightarrow D_0 x_0 = Aie^{iT_0} - \bar{A}ie^{-iT_0} \tag{7}$$

Then we solve for the $O(\epsilon)$ equation using the solution we got for x_0

$$
\begin{align}
&2D_1D_0x_0 + D_0^2x_1 + x_1 + \alpha(D_0x_0)^3 = 0 \tag{8}\\
\Rightarrow &D_0^2x_1 + x_1 = 2i(D_1\bar{A}e^{-iT_0} - D_1Ae^{iT_0}) - \alpha(Aie^{iT_0} - \bar{A}ie^{-iT_0})^3 \tag{9}\\
\Rightarrow &D_0^2x_1 + x_1 = 2i(D_1\bar{A}e^{-iT_0} - D_1Ae^{iT_0}) - \alpha[-A^3ie^{3iT_0} + \bar{A}^3ie^{-3iT_0} + 3A\bar{A}(Aie^{iT_0} - \bar{A}ie^{-iT_0})] \tag{10}
\end{align}
$$

We neglect the conjugate terms and secular terms because we don't want secular terms in our solution. Then we get the solvability condition for x_1

$$2D_1A + 3\alpha A^2\bar{A} = 0$$

Where we can write the amplitude of x_0 in a polar representation. This trick is to simplify the solution later on.

$$
\begin{align}
A &= \frac{1}{2}a(T_1)e^{i\theta(T_1)} \tag{11}\\
\bar{A} &= \frac{1}{2}a(T_1)e^{-i\theta(T_1)} \tag{12}
\end{align}
$$

Substitude back into the solvability condition and we get

$$
\begin{align}
&D_1ae^{i\theta(T_1)} + \frac{3\alpha}{8}a^3e^{i\theta(T_1)} = 0 \tag{13}\\
\Rightarrow &(D_1a)e^{i\theta(T_1)} + ai(D_1\theta)e^{i\theta(T_1)} + \frac{3\alpha}{8}a^3e^{i\theta(T_1)} = 0 \tag{14}\\
\Rightarrow &(D_1a) + ai(D_1\theta) + \frac{3\alpha}{8}a^3 = 0 \tag{15}
\end{align}
$$

Collect the real term and the imaginary term respectively and we get

$$
\begin{align}
&\mathbb{R} : (D_1a) + \frac{3\alpha}{8}a^3 = 0 \tag{16}\\
&\mathbb{I} : a(D_1\theta) = 0 \tag{17}
\end{align}
$$

From the imaginary term equation we can know that θ is constant in the T_1 time scale.

$$\theta \equiv \theta(T_2, T_3 \ldots) \approx const. = \theta_0$$

As for the real term equation, we can solve the ODE by separation of variables.

$$
\begin{align}
&\frac{-8}{3\alpha}a^{-3}da = dT_1 \tag{18}\\
\Rightarrow &\frac{4}{3\alpha}a^{-3} = T_1 + c^* \tag{19}\\
\Rightarrow &\frac{4}{3\alpha}a^{-3} = T_1 + c^* \tag{20}\\
\Rightarrow &a = \pm\frac{2}{\sqrt{3\alpha T_1 + c^*}} \tag{21}
\end{align}
$$

Finally, we can substitute this back into solution $x(\tau) = x_0 + \epsilon x_1$ and we get,

$$
\begin{align}
x(\tau) &= x_0 + \epsilon x_1 \tag{22}\\
&= \frac{1}{2}a(T_1)(e^{(iT_0+\theta)} + e^{-(iT_0+\theta)}) + \epsilon x_1 \tag{23}\\
&= a(T_1)\cos(T_0 + \theta(T_1)) + \epsilon x_1 \tag{24}\\
&\approx \pm\frac{2}{\sqrt{3\alpha\epsilon\tau + c^*}}\cos(\tau + \theta_0) \tag{25}
\end{align}
$$

From the initial condition $x(\tau = 0) = 0$, we can know $\theta_0 = -\pi/2$. As for the initial condition $\dot{x}(\tau = 0) = v$ we can obtain the following.

$$
\begin{align}
&\dot{x}(\tau) = -3\alpha\epsilon(3\alpha\epsilon\tau + c^*)^{-3/2}\cos(\tau + \theta_0) - \frac{2}{\sqrt{3\alpha\epsilon\tau + c^*}}\sin(\tau + \theta_0) \tag{26}\\
&\dot{x}(0) = \frac{2}{\sqrt{3\alpha\epsilon \cdot 0 + c^*}} \equiv \bar{v} \tag{27}\\
& \tag{28}\\
&\Rightarrow c^* = 4/\bar{v}^2 \tag{29}
\end{align}
$$

Finally, we have the solution with amplitude corrected:

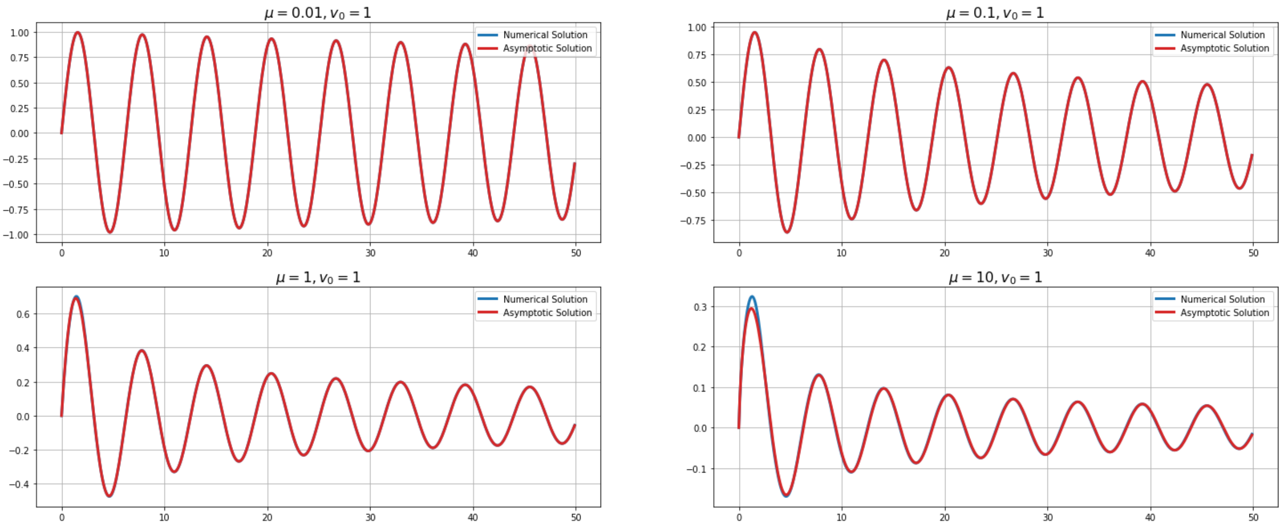
$$x(\tau) = \underbrace{\frac{2}{\sqrt{3\mu\tau + 4/\bar{v}^2}}}_{Amplitude} \bullet \underbrace{\sin(\tau)}_{Oscillation}$$

(d) Analysis

Analysis is in section (e)

In [7]:
```
# Importing relevant packages
import numpy as np
import matplotlib.pyplot as plt
```

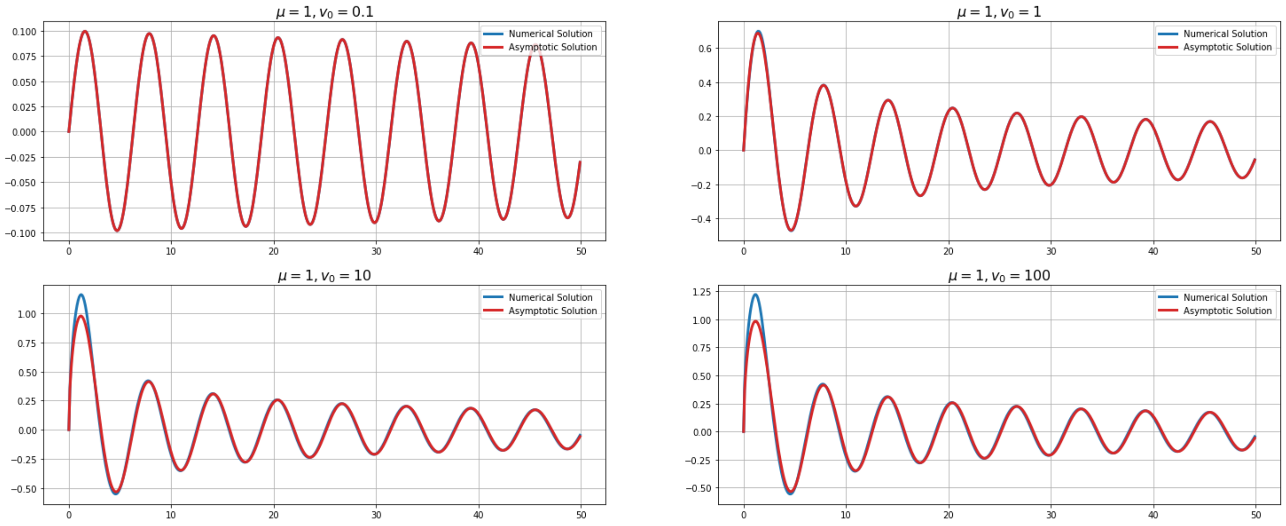
```python
import math
%matplotlib notebook
%matplotlib inline
from scipy.integrate import odeint

# Defining time vector, vector of initial conditions, and system parameter
tf, dt = 50, 0.1
t_vec = np.arange(0, tf, dt)
mu_paras = [0.01, 0.1, 1, 10]

# Defining the numerical solver for non-linear oscillator
def EXACT_EOM(x_vec, t):
    x1, x2 = x_vec
    x_vec_dot = [x2, -x1 - mu_para*math.pow(x2,3)]
    return x_vec_dot

x0 = 0
v0 = 1
x = np.zeros([4,t_vec.size]);
x_ms = np.zeros([4,t_vec.size]);

for i, mu_para in enumerate(mu_paras):
    IC_vec = [x0, v0]
    sol = odeint(EXACT_EOM, IC_vec, t_vec)
    x[i, :], x_dot = sol[:, 0], sol[:, 1]
    A = 2/np.sqrt(3*mu_para*t_vec+4/math.pow(v0, 2))
    x_ms[i, :] = np.multiply(A, np.sin(t_vec))

fig, ((ax1, ax2), (ax3, ax4)) = plt.subplots(2, 2, figsize=(25,10))
ax1.plot(t_vec, x[0,:],'tab:blue',linewidth=3)
ax1.plot(t_vec, x_ms[0,:],'tab:red',linewidth=3)
ax1.set_title('$\mu = 0.01, v_0 = 1$', fontsize=16, color='k')
ax1.legend({'Numerical Solution', 'Asymptotic Solution'}, loc ='upper right')
ax1.grid(True)

ax2.plot(t_vec, x[1,:], 'tab:blue',linewidth=3)
ax2.plot(t_vec, x_ms[1,:], 'tab:red',linewidth=3)
ax2.set_title('$\mu = 0.1, v_0 = 1$', fontsize=16, color='k')
ax2.legend({'Numerical Solution', 'Asymptotic Solution'}, loc ='upper right')
ax2.grid(True)

ax3.plot(t_vec, x[2,:], 'tab:blue',linewidth=3)
ax3.plot(t_vec, x_ms[2,:], 'tab:red', linewidth=3)
ax3.set_title('$\mu = 1, v_0 = 1$', fontsize=16, color='k')
ax3.legend({'Numerical Solution', 'Asymptotic Solution'}, loc ='upper right')
ax3.grid(True)

ax4.plot(t_vec, x[3,:], 'tab:blue',linewidth=3)
ax4.plot(t_vec, x_ms[3,:], 'tab:red',linewidth=3)
ax4.set_title('$\mu = 10, v_0 = 1$', fontsize=16, color='k')
ax4.legend({'Numerical Solution', 'Asymptotic Solution'}, loc ='upper right')
ax4.grid(True)
```



In [8]:

```python
x0 = 0
v0s = [0.1, 1, 10, 100]
x = np.zeros([4,t_vec.size])
x_ms = np.zeros([4,t_vec.size])
mu_para = 1

for i, v0 in enumerate(v0s):
    IC_vec = [x0, v0]
    sol = odeint(EXACT_EOM, IC_vec, t_vec)
    x[i, :], x_dot = sol[:, 0], sol[:, 1]
    A = 2/np.sqrt(3*mu_para*t_vec+4/math.pow(v0, 2))
    x_ms[i, :] = np.multiply(A, np.sin(t_vec))

fig, ((ax1, ax2), (ax3, ax4)) = plt.subplots(2, 2, figsize=(25,10))
ax1.plot(t_vec, x[0,:],'tab:blue',linewidth=3)
```

```
ax1.plot(t_vec, x_ms[0,:],'tab:red',linewidth=3)
ax1.set_title('$\mu = 1, v_0 = 0.1$', fontsize=16, color='k')
ax1.legend({'Numerical Solution', 'Asymptotic Solution'}, loc ='upper right')
ax1.grid(True)

ax2.plot(t_vec, x[1,:], 'tab:blue',linewidth=3)
ax2.plot(t_vec, x_ms[1,:], 'tab:red',linewidth=3)
ax2.set_title('$\mu = 1, v_0 = 1$', fontsize=16, color='k')
ax2.legend({'Numerical Solution', 'Asymptotic Solution'}, loc ='upper right')
ax2.grid(True)

ax3.plot(t_vec, x[2,:], 'tab:blue',linewidth=3)
ax3.plot(t_vec, x_ms[2,:], 'tab:red', linewidth=3)
ax3.set_title('$\mu = 1, v_0 = 10$', fontsize=16, color='k')
ax3.legend({'Numerical Solution', 'Asymptotic Solution'}, loc ='upper right')
ax3.grid(True)

ax4.plot(t_vec, x[3,:], 'tab:blue',linewidth=3)
ax4.plot(t_vec, x_ms[3,:], 'tab:red',linewidth=3)
ax4.set_title('$\mu = 1, v_0 = 100$', fontsize=16, color='k')
ax4.legend({'Numerical Solution', 'Asymptotic Solution'}, loc ='upper right')
ax4.grid(True)
```



(e). Describe $\bar{\mu}$

Answers for (e) & (d) :

As clearly illustrated from graph, the asymptotic solutions using multiple scale method are good approximations of numerical solution. They almost have no periodical error, but minor amplitude difference in the beginning.

Plus, we found that the asymptotic solutions are **more** valid when μ and $\bar{v}$ is **small**. This is because we assumed small damping. So when the μ is too large, the solution might have undershoot. Nonetheless, two solutions are almost perfectly synchronized after 10 [sec] no matter what initial conditions and damping parameter.

We can draw conclusion that the asymptotic solutions are valid when damping parameter $\bar{\mu}$ and initial velocity $\bar{v}$ are small.

When the damping parameter μ increases, damping effect on the system is more dominant compared to other forces. As shown in figures above, when $\bar{v} = 1$, larger μ makes the amplitude of oscillation smaller.

Question 2

(a) slow evolution equations

Rescaled EOM:

$$E.O.M. = \ddot{x} + 2\epsilon^2\mu\dot{x} + x + \epsilon a_2 x^2 + \epsilon^2 a_3 x^3 = \epsilon^2 F\cos(\Omega\tau)$$

with different time scales:

$$x(\tau) = x_0(T_0, T_1, T_2) + \epsilon x_1(T_0, T_1, T_2) + \epsilon^2 x_2(T_0, T_1, T_2)$$

Define different time scales as following. T_0 is the regular time scale and T_1 is the slow time scale, T_2 is the super-slow time scale.

$$T_0 = \tau \tag{30}$$
$$T_1 = \epsilon\tau \tag{31}$$
$$T_2 = \epsilon^2\tau \tag{32}$$

For each time scales, define the differtial operator as such:

$$D_i = \frac{\partial}{\partial T_i}$$

The equation can be rewritten as,

$$(D_0^2 + 2\epsilon D_1 D_0 + \epsilon^2(D_1^2 + 2D_0 D_2))x + 2\epsilon^2\mu(D_0 + \epsilon D1 + \epsilon^2 D_2)x + x + \epsilon a_2 x^2 + \epsilon^2 a_3 x^3 = \epsilon^2 F\cos(\Omega\tau)$$

Collect the same order term up to first three order then we get three equations

$$O(1): D_0^2 x_0 + x_0 = 0 \tag{33}$$
$$O(\epsilon): D_0^2 x_1 + x_1 + 2D_1 D_0 x_0 + a_2 x_0^2 = 0 \tag{34}$$
$$O(\epsilon^2): (D_1^2 + 2D_0 D_2)x_0 + 2D_0 D_1 x_1 + D_0^2 x_2 + 2\mu D_0 x_0 + x_2 + 2a_2 x_0 x_1 + a_3 x_0^3 = Fcos(\Omega\tau) \tag{35}$$

Solve for $O(1)$ equation, same as before.

$$x_0 = A(T_1, T_2)e^{iT_0} + \bar{A}(T_1, T_2)e^{-iT_0} \tag{36}$$
$$D_0 x_0 = Aie^{iT_0} - \bar{A}ie^{-iT_0} \tag{37}$$

Substitute x_0 to $O(\epsilon)$ equation,

$$D_0^2 x_1 + x_1 = -a_2 x_0^2 - 2D_1 D_0 x_0 \tag{38}$$
$$= -a_2(A^2 e^{2iT_0} + \bar{A}^2 e^{-2iT_0} + 2A\bar{A}) - 2i[(D_1 A)e^{iT_0} - (D_1\bar{A})e^{-iT_0}] \tag{39}$$

Neglect the non-secular term and the conjugate term, then we can obtain the solvability condition for x_0

$$-2iD_1 A(T_1, T_2)e^{iT_0} = 0 \tag{40}$$
$$\Rightarrow D_1 A = 0 \tag{41}$$
$$\Rightarrow A \equiv A(T_2) \tag{42}$$

We found out that $A, \bar{A}$ is only a function of T_2. Thus the $O(\epsilon)$ equation is reduced as follows.

$$D_0^2 x_1 + x_1 = -a_2(A^2 e^{2iT_0} + \bar{A}^2 e^{-2iT_0} + 2A\bar{A}) \tag{43}$$

We can solve for x_1 using variation of parameters principle.

$$x_1 = Be^{iT_0} + \bar{B}e^{-iT_0} + Ce^{2iT_0} + \bar{C}e^{-2iT_0} - 2a_2 A\bar{A}$$

Subs back,

$$-3(Ce^{2iT_0} + \bar{C}e^{-2iT_0}) - 2a_2 A\bar{A} = -a_2(A^2 e^{2iT_0} + \bar{A}^2 e^{-2iT_0} + 2A\bar{A})$$

Then we get $C, \bar{C}$ and expression for x_1

$$C = \frac{a_2}{3}A^2 \tag{44}$$
$$\bar{C} = \frac{a_2}{3}\bar{A}^2 \tag{45}$$
$$x_1 = Be^{iT_0} + \bar{B}e^{-iT_0} + \frac{a_2}{3}A^2 e^{2iT_0} + \frac{a_2}{3}\bar{A}^2 e^{-2iT_0} - 2a_2 A\bar{A} \tag{46}$$

Since we have expression for x_0, x_1, we can substitute them back into equation of $O(\epsilon^2)$. It's a rather long solution. We will just show the secular term.

$$e^{iT_0}(2D_2 Ai + 2D_1 Bi + 2\mu Ai - \frac{10}{3}a_2^2 A^2\bar{A} + 3a_3 A^2\bar{A}) = F\cos(\Omega T_0) \tag{47}$$

Since the forcing frequency is in the vicinity of the natural frequency $\Omega \approx 1$, we can write it in the second order representation.

$$\Omega \approx 1 = 1 + \epsilon\sigma_1 + \epsilon^2\sigma_2$$

As for the cosine term, we can expand it with the Euler equation.

$$F\cos(\Omega\tau) = \frac{F}{2}(e^{i\Omega T_0} + e^{-i\Omega T_0}) = \frac{F}{2}(e^{i(1+\epsilon\sigma+\epsilon^2\sigma_2)T_0} + e^{-i(1+\epsilon\sigma+\epsilon^2\sigma_2)T_0}) \tag{48}$$
$$= \frac{F}{2}(e^{iT_0}e^{i\sigma_1 T_1}e^{i\sigma_2 T_2} + e^{-iT_0}e^{-i\sigma_1 T_1}e^{-i\sigma_2 T_2}) \tag{49}$$

Solvability Condition:

$$2D_2 Ai + 2D_1 Bi + 2\mu Ai - \frac{10}{3}a_2^2 A^2\bar{A} + 3a_3 A^2\bar{A} - \frac{F}{2}e^{i\sigma T_1}e^{i\sigma_2 T_2} = 0 \tag{50}$$

Where we can write the amplitude of x_0 in a polar representation. This trick is to simplify the solution later on.

$$A = \frac{1}{2}a(T_2)e^{i\theta(T_2)} \tag{51}$$
$$\bar{A} = \frac{1}{2}a(T_2)e^{-i\theta(T_2)} \tag{52}$$
$$B = \frac{1}{2}b(T_1, T_2)e^{i\phi(T_1, T_2)} \tag{53}$$

Substitute the polar representaion back to the solvability condition and we get the following equations.

$$D_2a(T_2)e^{i\theta(T_2)}i + D_1b(T_1,T_2)e^{i\phi(T_1,T_2)}i + \mu a(T_2)e^{i\theta(T_2)}i + \frac{1}{8}a^3(T_2)e^{i\theta(T_2)}(\frac{-10}{3}a_2^2 + 3a_3) - \frac{F}{2}e^{i\sigma_1 T_1}e^{i\sigma_2 T_2} = 0 \tag{54}$$

$$a_{,2}i - a\theta_{,2} + b_{,1}e^{i(\phi-\theta)}i - b\phi_{,1}e^{i(\phi-\theta)} + \mu ai + \frac{1}{8}a^3(\frac{-10}{3}\alpha_2^2 + 3\alpha_3) - \frac{F}{2}e^{i(\sigma T_1 + \sigma_2 T_2 - \theta)} = 0 \tag{55}$$

For simplicity, we define:

$$\lambda \equiv \phi - \theta$$

$$\Theta \equiv \sigma_1 T_1 + \sigma_2 T_2 - \theta$$

And the equation becomes:

$$a_{,2}i - a(\sigma_2 - \Theta_{,2}) + b_{,1}e^{i\lambda}i - b\lambda_{,1}e^{i\lambda} + \mu ai + \frac{1}{8}a^3(\frac{-10}{3}\alpha_2^2 + 3\alpha_3) - \frac{F}{2}e^{i\Theta} = 0 \tag{56}$$

$$a_{,2}i - a(\sigma - \Theta_{,2}) + b_{,1}(\cos\lambda + i\sin\lambda)i - b\lambda_{,1}(\cos\lambda + i\sin\lambda) + \mu ai + \frac{1}{8}a^3(\frac{-10}{3}\alpha_2^2 + 3\alpha_3) - \frac{F}{2}(\cos\Theta + i\sin\Theta) = 0 \tag{57}$$

$$\begin{cases} \mathbb{R}: \quad -a(\sigma_2 - \Theta_{,2}) - b_{,1}\sin\lambda - b\lambda_{,1}\cos\lambda + \frac{1}{8}a^3(\frac{-10}{3}\alpha_2^2 + 3\alpha_3) - \frac{F}{2}\cos\Theta = 0 \\ \mathbb{I}: \quad a_{,2} + b_{,1}\cos\lambda - b\lambda_{,1}\sin\lambda + \mu a - \frac{F}{2}\sin\Theta = 0 \end{cases} \tag{58}$$

The solvbility function is still too intricate to solve, however, we can take a glimpse at the steady state:

$$a_{,2} = b_{,1} = 0, \quad \Theta_{,i} = \lambda_{,i} = 0$$

And since $\theta = \theta(T_2)$,

$$\Theta_{,1} = \sigma_1 - \theta_1 \Rightarrow \sigma_1 = 0 \tag{59}$$

Thus, we obtain the slow evolution equations.

$$\begin{cases} \mathbb{R}: \quad -a\sigma_2 + \frac{1}{8}a^3(\frac{-10}{3}\alpha_2^2 + 3\alpha_3) - \frac{F}{2}\cos\Theta = 0 \\ \mathbb{I}: \quad \mu a - \frac{F}{2}\sin\Theta = 0 \end{cases} \tag{60}$$

(b) $a(\Omega)$

Obtain a close expression for the system's frequency response, i.e. the relation between the steady-state response amplitude of the system and the forcing frequency $a(\Omega)$, for a given set of parameters:$\bar{\alpha}_2, \bar{\alpha}_3, \bar{\mu}, \bar{F}$.

First of all, recap solvbility funtion

$$\begin{cases} -a\sigma_2 + \frac{1}{8}a^3(\frac{-10}{3}\alpha_2^2 + 3\alpha_3) - \frac{F}{2}\cos\Theta = 0 \\ \mu a - \frac{F}{2}\sin\Theta = 0 \end{cases} \tag{61}$$

Using trig identity, $\cos^2 + \sin^2 = 1$,

$$\frac{F^2}{4} = (-a\sigma_2 + \frac{1}{8}a^3(\frac{-10}{3}\alpha_2^2 + 3\alpha_3))^2 + (\mu a)^2 \tag{62}$$

$$\Rightarrow \sigma_2 = \pm\frac{1}{\epsilon^2}\sqrt{\frac{\bar{F}^2}{4a^2} - \bar{\mu}^2} + \frac{1}{8\epsilon^2}a^2(\frac{-10}{3}\bar{\alpha}_2^2 + 3\bar{\alpha}_3) \tag{63}$$

Recall that $\Omega = 1 + \epsilon\sigma_1 + \epsilon^2\sigma_2, \quad \sigma_1 = 0$, thus,

$$\Omega(a) = 1 \pm \sqrt{\frac{\bar{F}^2}{4a^2} - \bar{\mu}^2} + \frac{1}{8}a^2(\frac{-10}{3}\bar{\alpha}_2^2 + 3\bar{\alpha}_3) \tag{64}$$

$$a(\Omega) = \Omega^{-1}(a) \tag{65}$$

(c) $\bar{\beta}$

From $\Omega(a)$, we can see there's a non-dimensional parameter that represents the effect of the nonlinear terms on the shape of the frequency response curve which is

$$\bar{\beta} = \frac{1}{8}(\frac{-10}{3}\bar{\alpha}_2^2 + 3\bar{\alpha}_3) \tag{66}$$

So the frequency response $\Omega(a)$ is then,

$$\Omega(a) = 1 \pm \sqrt{\frac{\bar{F}^2}{4a^2} - \bar{\mu}^2} + \bar{\beta}a^2 \tag{67}$$

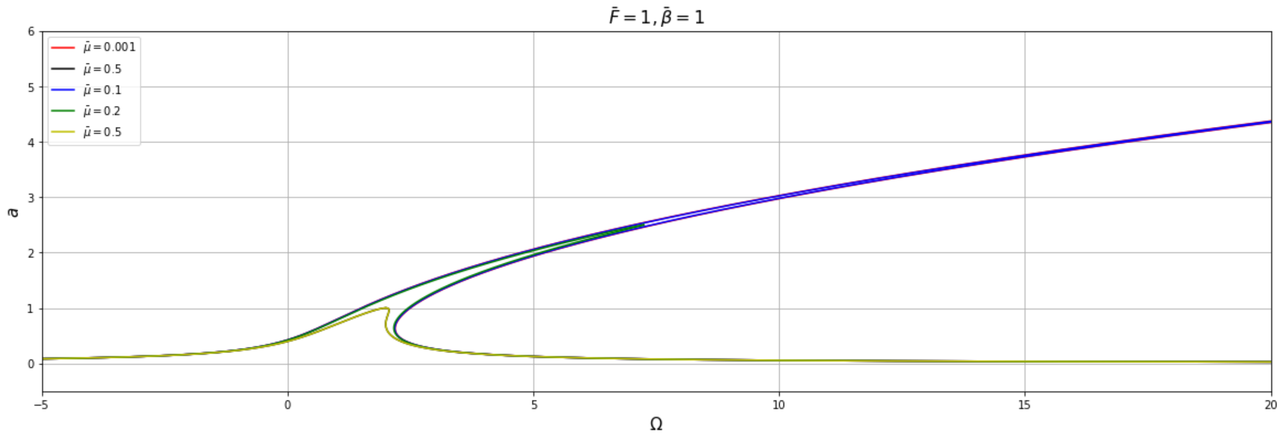
(d) damping effect

For the given parameters, our equation is then:

$$\Omega(a) = 1 \pm \sqrt{\frac{1}{4a^2} - \bar{\mu}^2 + a^2} \tag{68}$$

In [9]:

```
# Defining a, and system parameter
a = np.arange(0.01, 10, 0.001)
mu_paras = [0.001, 0.5, 0.1, 0.2, 0.5]
color = ['r','k','b','g','y']
# Defining the function
def f(a, mu):
    a_valid = a[1/2/mu>a]
    sol1 = 1 + np.sqrt((1/(4*a_valid**2)-mu**2)) + a_valid**2
    sol2 = 1 - np.sqrt((1/(4*a_valid**2)-mu**2)) + a_valid**2
    return a_valid, sol1, sol2

plt.figure(figsize=(20,6))
legendlist = []
for idx, mu_para in enumerate(mu_paras):
    plt.plot(f(a,mu_para)[1], f(a,mu_para)[0], color[idx])
    plt.plot(f(a,mu_para)[2], f(a,mu_para)[0], color[idx])
    legendlist.append(r'$\bar \mu=$' + str(mu_para))

plt.legend(legendlist,loc='best')
plt.title(r'$\bar F = 1, \bar \beta =1 $', fontsize=16, color='k')
plt.xlabel(r'$\Omega$', fontsize=15)
plt.ylabel(r'$a$', fontsize=15)
plt.xlim((-5,20))
plt.ylim((-0.5,6))
# plt.legend({'Numerical Solution', 'Asymptotic Solution'}, loc ='best')
plt.grid(True)

leg = plt.gca().get_legend()
for idx in range(len(mu_paras)):
    leg.legendHandles[idx].set_color(color[idx])
```



Analysis

As the figure above dipicted, we noticed that as $\bar{\mu}$ increases, the 'peak' of the amplitude of the solution a decreases correspondly and it gets smoother, the multiple-solutions region is narrower and nonlinearity is less dominant. To conlcude, if nonlinearity parameter α_2, α_3 is large, high damping μ helps alleviate the nonlinear phenomena.

(e) Effect of nonlinearities

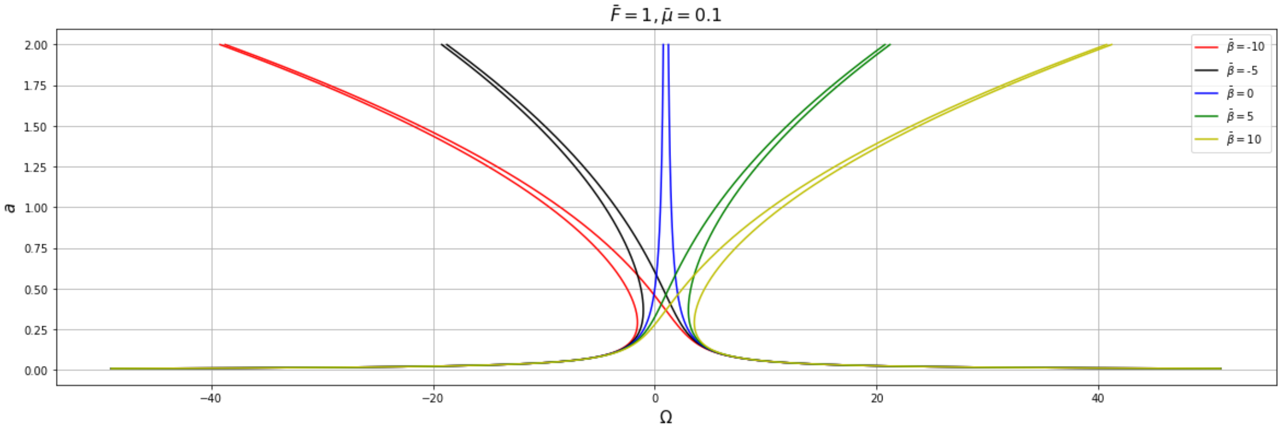
For the given parameters, our equation is then:

$$\Omega(a) = 1 \pm \sqrt{\frac{1}{4a^2} - 0.01 + \bar{\beta} a^2} \tag{69}$$

In [10]:

```
# Defining a, and system parameter
a = np.arange(0.01, 2, 0.001)
beta_paras = [-10, -5, 0, 5, 10]
color = ['r','k','b','g','y']
# Defining the function
def f(a, beta):
    sol1 = 1 + np.sqrt((1/(4*a**2)-0.01)) + beta * a**2
    sol2 = 1 - np.sqrt((1/(4*a**2)-0.01)) + beta * a**2
    return sol1, sol2

plt.figure(figsize=(20,6))
legendlist = []
for idx, beta_para in enumerate(beta_paras):
    plt.plot(f(a,beta_paras[idx])[0], a, color[idx])
    plt.plot(f(a,beta_paras[idx])[1], a, color[idx])
    legendlist.append(r'$\bar \beta=$' + str(beta_para))

plt.legend(legendlist,loc='best')
plt.title(r'$\bar F = 1, \bar \mu =0.1 $', fontsize=16, color='k')
plt.xlabel(r'$\Omega$', fontsize=15)
plt.ylabel(r'$a$', fontsize=15)
```

```python
# plt.legend({'Numerical Solution', 'Asymptotic Solution'}, loc ='best')
plt.grid(True)

leg = plt.gca().get_legend()
for idx in range(len(mu_paras)):
    leg.legendHandles[idx].set_color(color[idx])
```



Analysis

When $\bar{\beta} > 0$, we have a hardening non-linearity as definition because the peak of amplitude falls on the right; when β is around zero, our system is close to linearity; when $\beta < 0$, we have softening non-linearity because the peak of amplitude bends to the left.

Recall that:

$$\bar{\beta} = \frac{1}{8}(\frac{-10}{3}\bar{\alpha}_2^2 + 3\bar{\alpha}_3) \tag{70}$$

The non-linear parameter $\bar{\alpha}_2$ causes the system to be softening, because it always depreciates the value of $\bar{\beta}$, as explained the system goes to softening on-linearity; The non-linear parameter $\bar{\alpha}_3$ causes the system to be hardening. If it is a positive real number, it will enlarge the value of $\bar{\beta}$, as explained the system goes to hardening non-linearity and if it is negative, it causes the system to have softening non-linearity.

As for explaining how this happens, we can turn to duffing equation. In duffing equation, α_3 describes the amouont of non-linearity in the restoring force. Thus, when α_3 increases, as if the spring is getting stiffer.

(f) jump

Yes. Jumping effect is plausible. When the system is hardened, or softened, it might bend the curve too much and causes the graph to have multiple solutions. When we change the frequency of the forcing Ω in muliple solutions domain, it might cause a jumping effect (drop from larger amplitude to smaller amplitude).

Also, when damping coefficient μ is small, the graph is sharp, and the system's nonlinear phenomena appears. Again, we will have multiple solutions on the frequency response because of the sharp bump, and when we change the frequency of the forcing Ω, it might cause a jumping effect.