Mixed Signal Design – Assignment

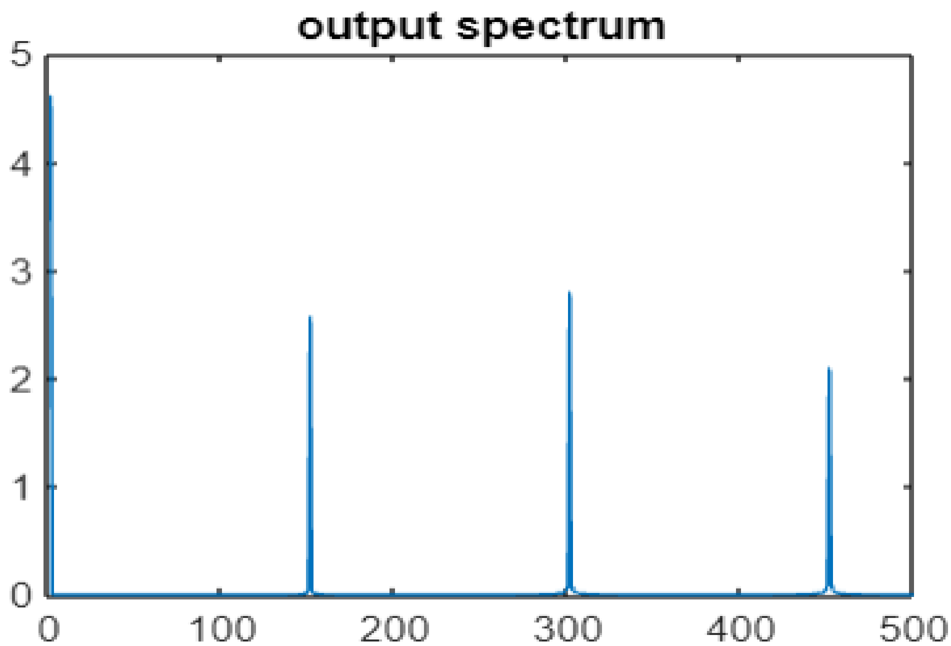
N. NIKHIL KUMAR

18BEC031 - ECE

- Plot output spectrum of nonlinear amplifier for a sinusoidal input. Quantify distortion.

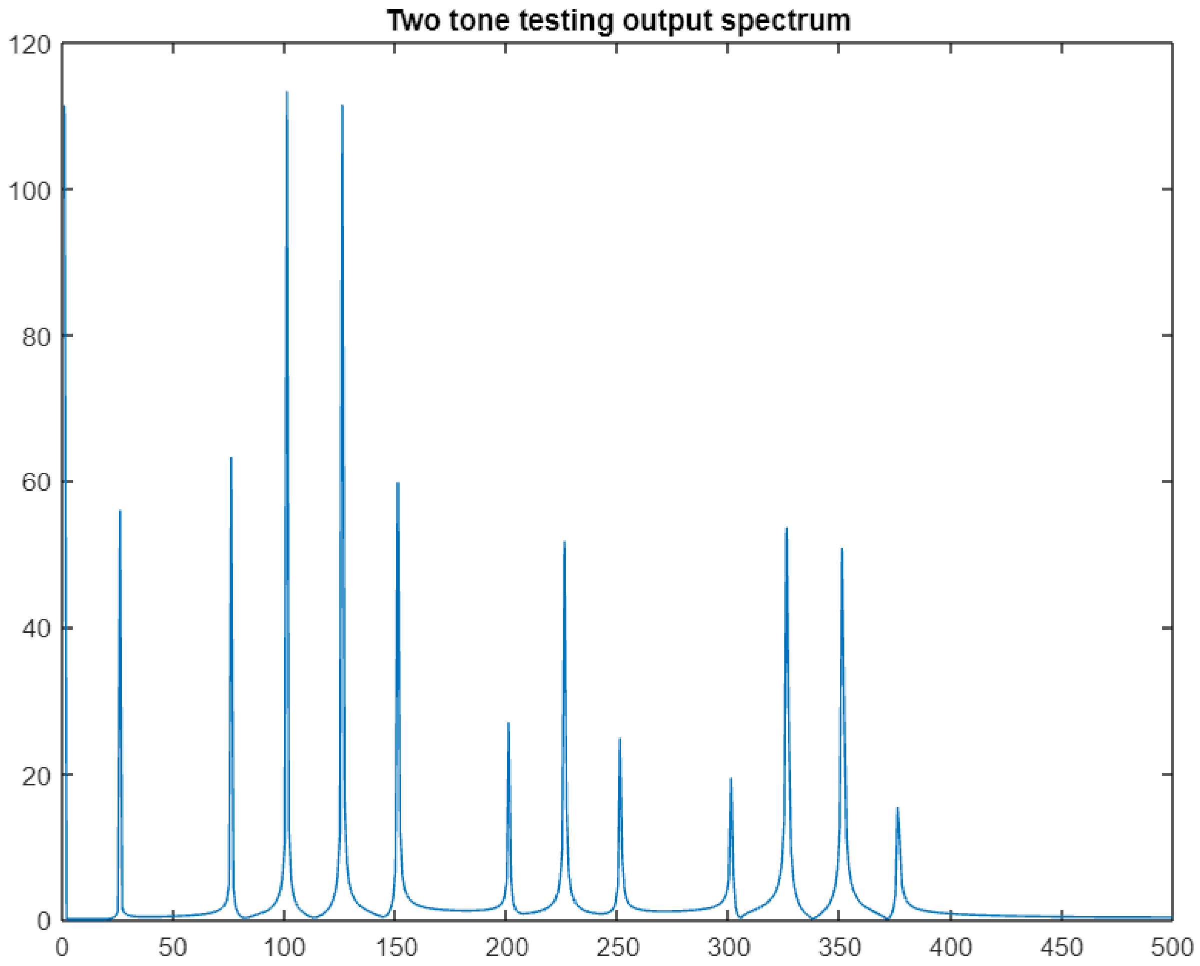
Matlab Code –

```
t=0:0.0001:1;
Vm=15;
k1=1;
k2=0.5;
k3=0.05;
k4=0.005;
Vin=Vm*sin(300*pi*t);
Vout=k1+(k2*Vin)-(k3*(Vin.^2))-(k4*(Vin.^3));
hd2=thd(Vout);
hd=(((k2*Vm).^2+((k3*(Vm.^2)).^2)+((k4*(Vm.^3)).^2)).^(1/2))/k1;
V=(fft(Vout))/length(Vout);
plot(abs(V));
xlim([0 500]);
title('output spectrum');
```



- Two tone testing of nonlinear amplifier. Plot output spectrum and quantify distortion.
  Matlab Code –

```
t=0:0.001:1;
vm=15;
k0=1;      %offset
k1=10;     %fundamental gain
k2=0.5;
k3=0.05;
vin2=vm*(sin(200*pi*t)+sin(250*pi*t))
vout2=k0+(k1*vin2)-(k2*(vin2.^2))-(k3*(vin2.^3));
v2=(fft(vout2))/length(vout2)
plot(abs(v2))
xlim([0 500])
title('Two tone testing output spectrum')
```



- Graphically compute IIP3

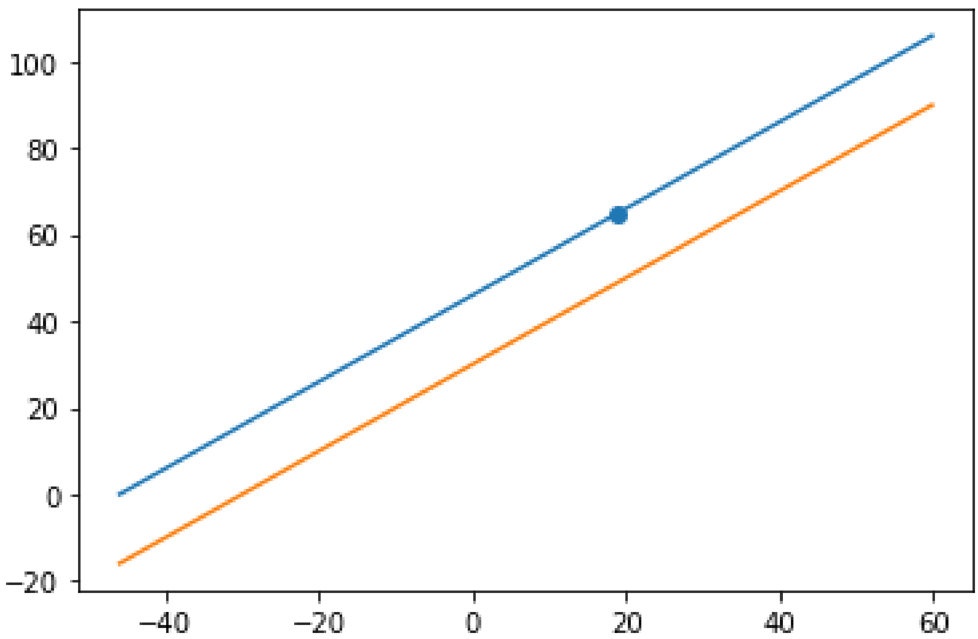
  Code –

```
import numpy as np
import matplotlib.pyplot as plt
import scipy
import math
Vm = np.arange(0.1,20,0.1)

a0 = 1
a1 = 10
a2 = -1
a3 = -2
Vout_fund_20log = 20*np.log(Vm) + 20*np.log(a1)
Vm_IM3 = (3/4)*a3*Vm*3
Vm_IM3_20log = 20*np.log(abs(Vm_IM3))

plt.plot(20*np.log(Vm),Vout_fund_20log)
plt.plot(20*np.log(Vm),Vm_IM3_20log)

Vm_form = np.sqrt((4/3)*abs(a1/a3))
plt.scatter(20*np.log(Vm_form),20*np.log(Vm_form) + 20*np.log(a1))
```

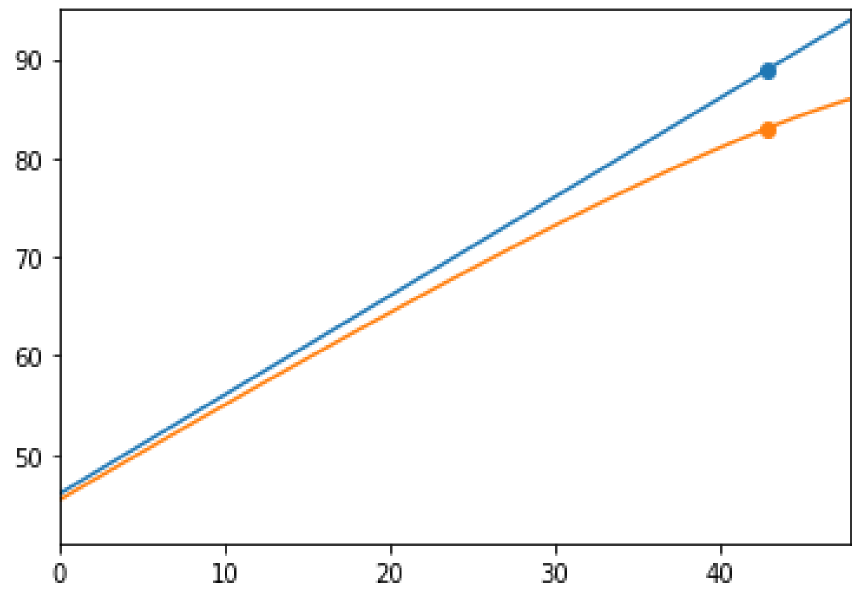


- Plot Pout vs Pin. Locate 1dB compression point.

  Code –

```
import numpy as np
import matplotlib.pyplot as plt
import scipy
import math
Vm = np.arange(0.01,2,0.01)

a0 = 1
a1 = 10
a2 = -1
a3 = -2
R = 5
Pin = 10*np.log(Vm**2/(2*R*0.001))
Pout_ideal = Pin + 20*np.log(a1)
Pout_actual = Pin + 20*np.log(a1 + ((3/4)*a3*Vm*2))
N = 1
alog = 10**(N/20)
Comp_Vm = np.sqrt((a1*(1-alog))/((3/4)*alog*a3))
Pin_comp_pt = 10*np.log(Comp_Vm**2/(2*R*0.001))
print("Compression point Vm = ",Comp_Vm)
print("Pin at Compression pt = ",Pin_comp_pt)
p1 = Pin_comp_pt + 20*np.log(a1)
p2 = Pin_comp_pt + 20*np.log(a1 + ((3/4)*a3*Comp_Vm*2))
plt.xlim([0,Pin_comp_pt +5])
plt.ylim([20*np.log(a1) - 5,10*math.ceil(p1/10) +5])
plt.plot(Pin,Pout_ideal)
plt.plot(Pin,Pout_actual)
plt.scatter(Pin_comp_pt,p1)
plt.scatter(Pin_comp_pt,p2)
plt.show()
```



Compression point Vm = 0.8514656456810393
Pin at Compression pt = 42.83577935306219