

ECOMMS Homework 2 - Aidan Sharpe

Problem 1

```
import numpy as np

f_c = 1250
f_m = 125
A_c = 10
a = 1

def g(t):
    return A_c * (1 + a*(0.2*np.cos(2*np.pi*f_m*t) + 0.5*np.sin(2*np.pi*f_c*t)))

# First derivative of g(t)
def dg_dt(t):
    return -0.2*2*f_m*np.pi*A_c*a*np.sin(2*np.pi*f_m*t) \
        + 2*np.pi*0.5*f_c*A_c*np.cos(2*np.pi*f_c*t)

# Second derivative of g(t)
def ddg_dtt(t):
    return -0.2*(2*np.pi*f_m)**2*A_c*a*np.cos(2*np.pi*f_m*t) \
        - 0.5*(2*np.pi*f_c)**2*A_c*a*np.sin(2*np.pi*f_c*t)

# Use Newton's method to find the maximum of the function
def newton_method(t):
    for i in range(10):
        t = t - dg_dt(t)/ddg_dtt(t)
    print("Newton method:", t, np.max(g(t)))
    return np.max(g(t))

def dc_dt(t):
    return A_c*a*np.pi*f_c*np.cos(2*np.pi*f_c*t)

# Sample g(t) at the maxima and minima of the carrier signal
def sample_method():
    samples = f_c / f_m
    n = np.arange(samples)
    t = (2*n + 1) / (4*f_c)
    t = t[np.argmax(g(t))]
    print("Sampling method:", t, g(t))
    return t

if __name__ == '__main__':
    t_max = sample_method()
    A_max = newton_method(t_max)
```

```
a_coeff = (A_max - A_c) / A_c  
print("Value of a where positive modulation is 90%:", 0.9/a_coeff)
```

Sampling Method

$t_{max} = 0.0002$, $g_{max} = 16.9754$ ### Newton Method $t_{max} = 0.000199206$,
 $g_{max} = 16.9755$

Positive modulation is 90% when $a = 1.2902$