

بسمه تعالی

گزارش پروژه مدولاسیون با GUI در Python

این پروژه از چند قسمت مجزا برای انجام مدولاسیون و تشکیل سیگنال های حامل و پیام متشکل می باشد:

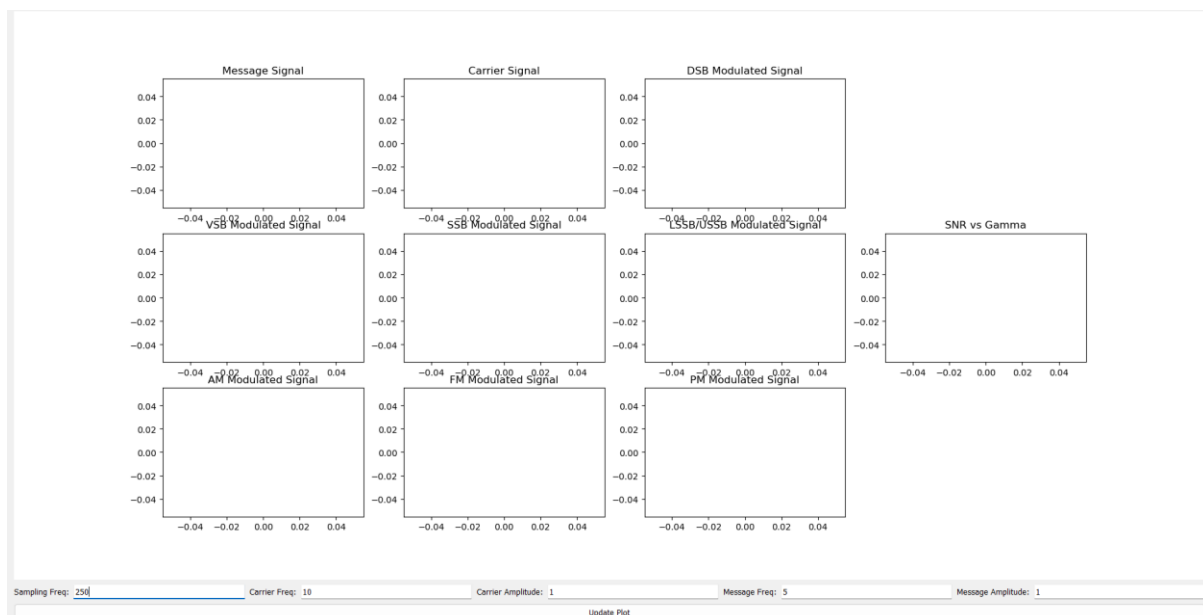
مراحل انجام:

1. ایجاد یک GUI با PyQt5

برای اینکار ابتدا کتابخانه های لازم را در پایتون با اجرای این کد در cmd نصب میکنیم:

```
>> pip install matplotlib numpy PyQt5 scipy
```

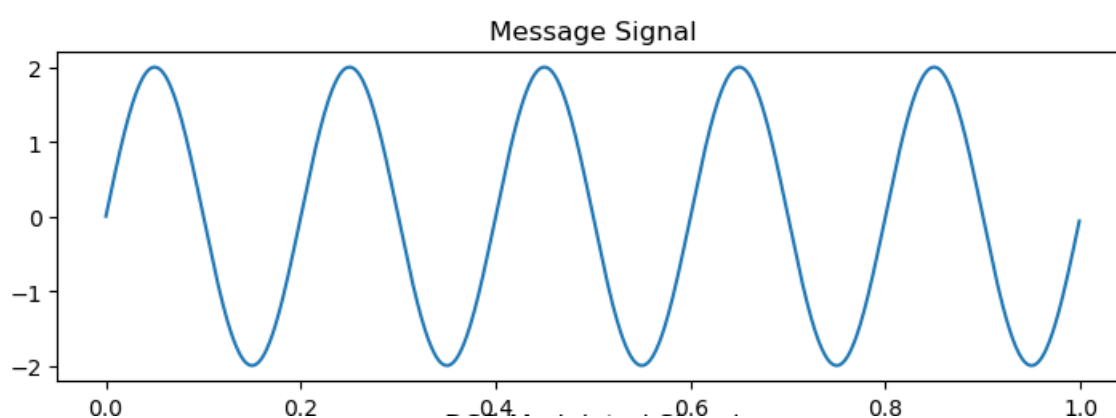
سپس با استفاده از این PyQt5 شروع به ساخت یک پنجره میکنیم و با قرار دادن پلات هایی که در matplotlib ایجاد میکنیم، صفحه را به شکل زیر پرمیکنیم و در زیر آن دکمه ها و محل ورود پارامتر ها را مشخص میکنیم.



2. تشکیل سیگنال پیام

در نظر داشته باشید که کل سیگنال ها در بازه ی زمانی یک ثانیه ایجاد شده و بررسی شده اند.

برای تشکیل سیگنال پیام ابتدا از GUI و قسمت ورودی برای فرکانس و دامنه آن مقدار ورودی را میخوانیم و با تشکیل بردار زمان، یک سیگنال سینوسی برای سیگنال پیام به صورت زیر تشکیل میدهیم.



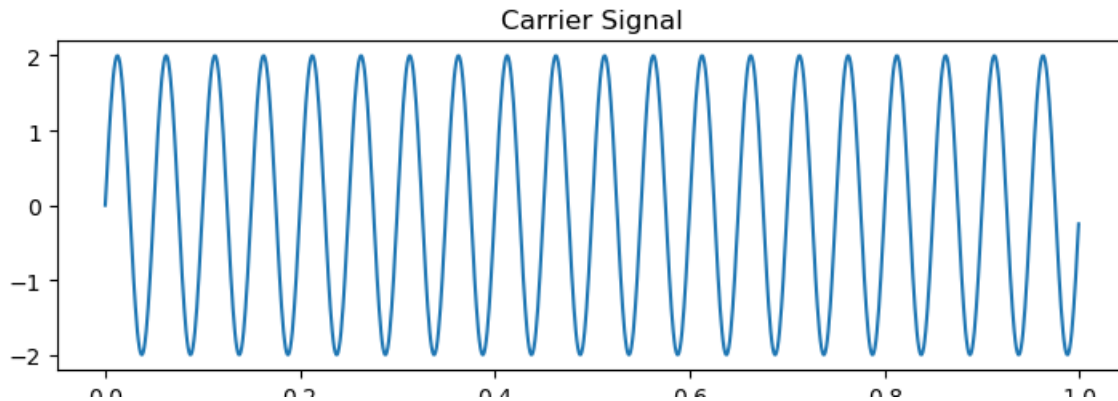
```
# Retrieve input values and save them into variables
Fs = float(self.inputs["Sampling Freq"].text())
Fm = float(self.inputs["Message Freq"].text())
Am = float(self.inputs["Message Amplitude"].text())

T = np.arange(0, 1, 1/Fs) # Time vector (1 second
duration)

# Message signal (modulating signal)
MessageSig = Am * np.sin(2 * np.pi * Fm * T)
```

3. تشکیل سیگنال حامل

همچنین برای تشکیل سیگنال حامل هم ابتدا از GUI و قسمت ورودی برای فرکانس و دامنه آن مقدار ورودی را میخوانیم و با تشکیل بردار زمان، یک سیگنال سینوسی برای سیگنال پیام به صورت زیر تشکیل میدهیم.



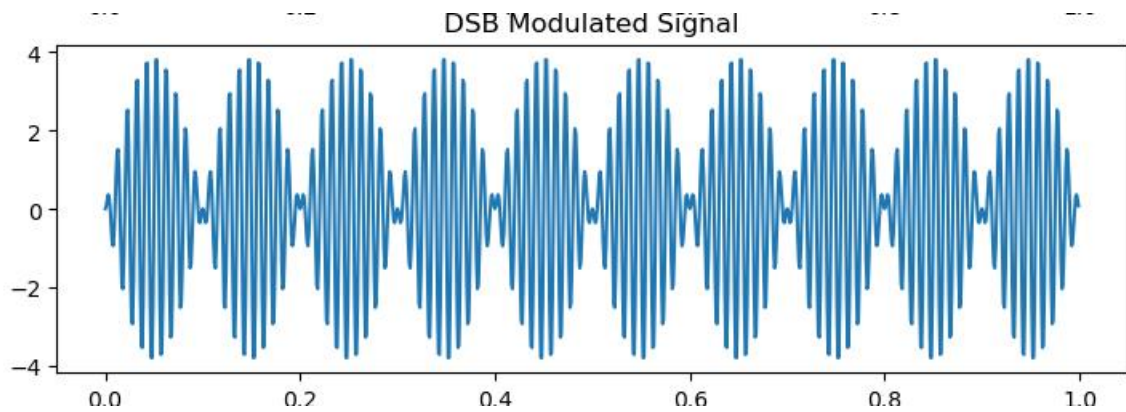
```
Fc = float(self.inputs["Carrier Freq"].text())
Ac = float(self.inputs["Carrier Amplitude"].text())

T = np.arange(0, 1, 1/Fs) # Time vector (1 second duration)

# Carrier signal
CarrSig = Ac * np.sin(2 * np.pi * Fc * T)
```

4. مدولاسیون DSB

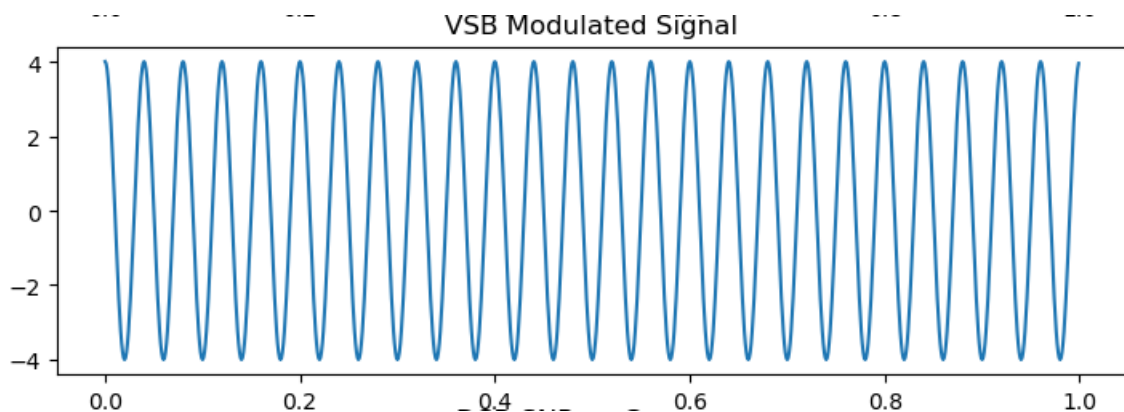
حال برای انجام مدولاسیون، با استفاده از ابزار های موجود در `scipy` میپردازیم.
برای مثال مدولاسیون DSB به این صورت قابل انجام میباشد.



```
# DSB Modulation  
DsbSig = MessageSig * CarrSig
```

5. مدولاسیون VSB

به همین منوال به مدولاسیون VSB با تابع `Hilbert` موجود در `scipy` میپردازیم.



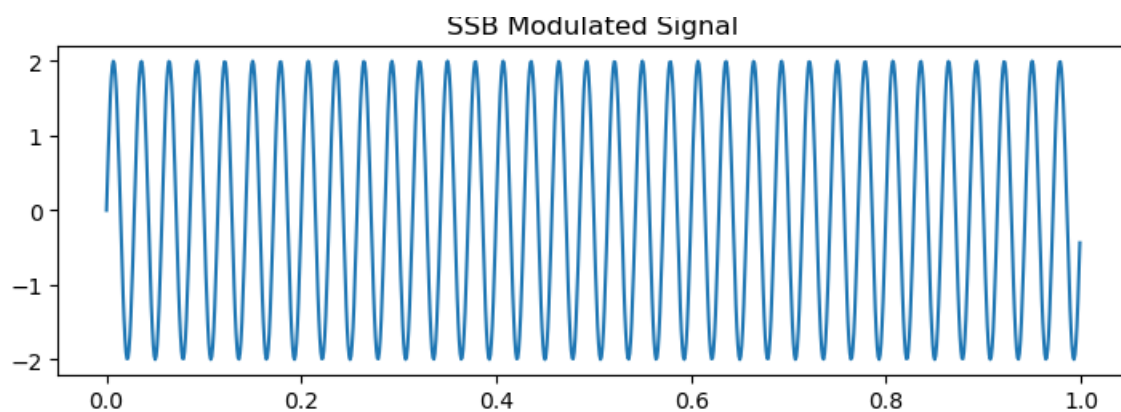
```

# VSB Modulation
# Hilbert Transform of the carrier signal
HilbertCarrSig = np.imag(hilbert(CarrSig))
HilbertMsgSig = np.imag(hilbert(MessageSig))
VsbSig = MessageSig * CarrSig + HilbertMsgSig * HilbertCarrSig

```

6. مدولاسیون SSB

به همین منوال به مدولاسیون SSB میپردازیم.



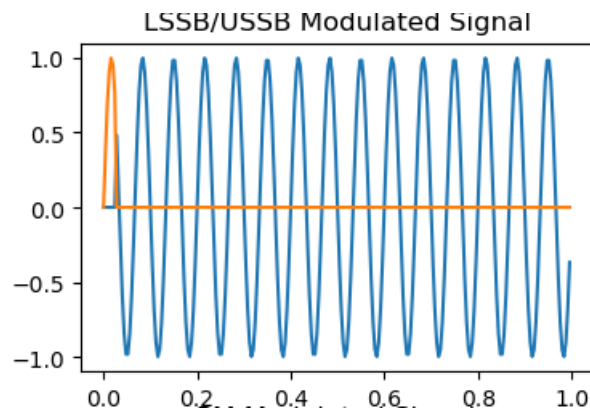
```

# SSB Modulation
SsbSig = MessageSig * np.cos(2 * np.pi * Fc * T) - \
        HilbertMsgSig * np.sin(2 * np.pi * Fc * T)

```

7. مدولاسیون LSSB/USSB

به همین منوال به مدولاسیون SSB میپردازیم.



LSSB Modulation

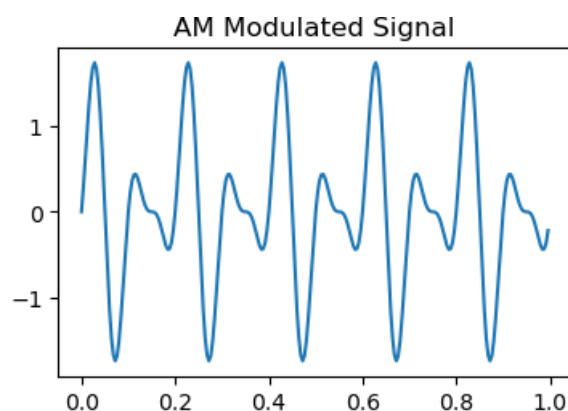
```
LSSBModSig = MessageSig * \
    np.cos(2 * np.pi * Fc * T) - HilbertMsgSig * \
    np.sin(2 * np.pi * Fc * T)
LSSBModSig = LSSBModSig * np.heaviside(2 * np.pi * Fc * T - np.pi/2,
0)
```

USSB Modulation

```
USSBModSig = MessageSig * \
    np.cos(2 * np.pi * Fc * T) - HilbertMsgSig * \
    np.sin(2 * np.pi * Fc * T)
USSBModSig = USSBModSig * np.heaviside(np.pi/2 - 2 * np.pi * Fc * T,
0)
```

8. مدولاسیون AM

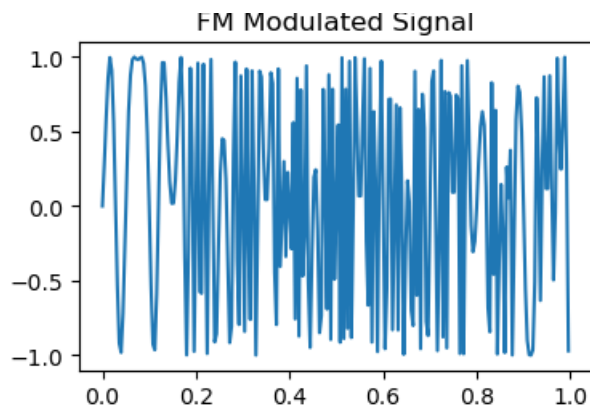
به همین منوال به مدولاسیون AM میپردازیم.



```
# AM modulation  
AMModSig = (1 + MessageSig) * CarrSig
```

9. مدولاسیون FM

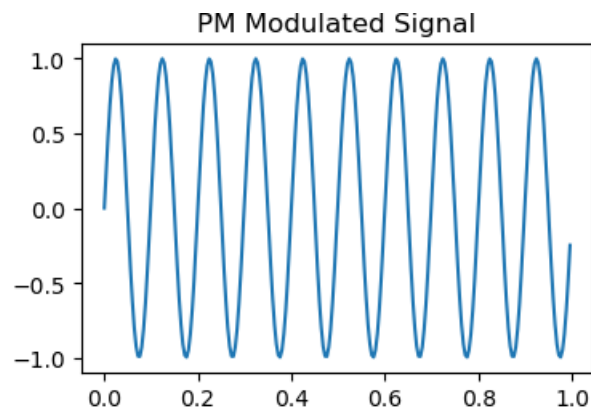
به همین منوال به مدولاسیون FM با داشتن Deviation Factor برای فرکانس میپردازیم.



```
# FM modulation  
Kf = 10.0 # Frequency deviation factor  
FMModSig = np.sin(2 * np.pi * (Fc + Kf * MessageSig) * T)
```

0 1 . مدولاسیون PM

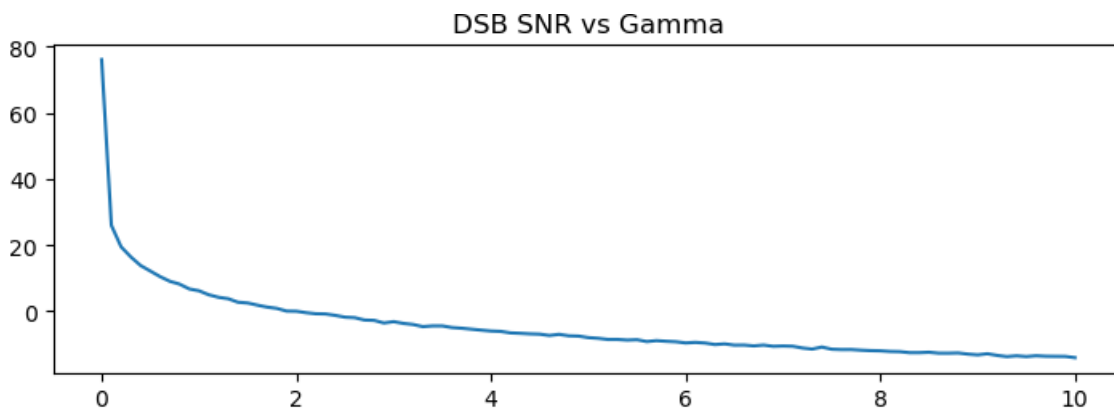
به همین منوال به مدولاسیون PM با داشتن Deviation Factor برای فاز میپردازیم.



```
# PM modulation
Kp = 1.0 # Phase deviation factor
Pm = np.cumsum(MessageSig) / Fs
PMModSig = np.sin(2 * np.pi * Fc * T + Kp * Pm)
```

1 1 . محاسبه SNR برای یکی از مدولاسیون ها براساس گاماهای متفاوت

برای محاسبه SNR با ایجاد های سیگنال نویز رندوم گاوسی با واریانس های گاما، به ازای گاماهای متفاوت، آنها را به صورت نسبت توان سیگنال نویز به سیگنال اصلی به صورت زیر حساب میکنیم و در کل منحنی آن را رسم میکنیم.




```
# SNR Calculation
# Different levels of gamma (ratio of signal amplitude to noise
amplitude)
GammaVals = np.arange(0, 10.1, 0.1)
SNRVals = np.zeros(len(GammaVals)) # Initialize SNR values

for i, gamma in enumerate(GammaVals):
    # Generate Gaussian noise
    Noise = gamma * np.random.randn(len(DsbSig))
    NoisySig = DsbSig + Noise # Add noise to modulated signal

    # Calculate signal power and noise power
    SigPow = np.mean(DsbSig ** 2)
    NoisePow = np.mean(Noise ** 2)

    # Calculate SNR
    SNRVals[i] = 10 * np.log10(SigPow / (NoisePow + 1e-7))
```

2.1. ظاهر نهایی پروژه

در نهایت با قرار دادن منطق توضیح داده شده در بستر PyQt5 پایتون میتوان آن را به صورت یک اپلیکیشن مستقل ارائه داد.

