lab_2

March 19, 2020

In [2]: %matplotlib inline

0.1

-:

$$s(t) = A_c \cdot (1 + m \cdot \cos(\omega_m t + \phi)) \cdot \cos(\omega t)$$

, :

- $A_c $ -- ;
- ω -- ;
- ω_m -- ;
- ϕ -- ;
- m -- 0 < m<= 1.

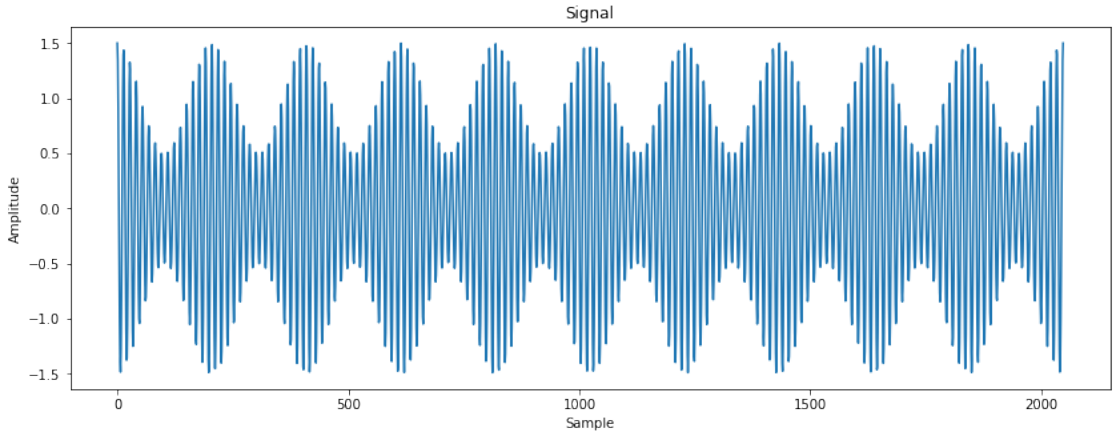
In [3]:
```
#  -
import numpy as np
import matplotlib.pyplot as plt

def get_am_signal(amp=1.0, m=0.25, f_c=10.0, f_s=2.0, period=100):
    """
    amp :
    m :  ,  0 <= m < 1
    f_c :
    f_s :
    period :
    """
    t = 2.0 * np.pi * np.linspace(0, 1, period)
    return amp * (1 + m * np.cos(f_s * t)) * np.cos(f_c * t)

signal = get_am_signal(amp=1.0, m=0.50, f_c=150.0, f_s=10.0, period=2048)

plt.figure(figsize=(14, 5))
plt.title('Signal')
plt.xlabel("Sample")
plt.ylabel("Amplitude")
plt.plot(signal)
```

Out[3]: [<matplotlib.lines.Line2D at 0x10e066da0>]




```
In [ ]: #
        import numpy as np
        from numpy import fft
        import matplotlib.pyplot as plt

        signal = get_am_signal(amp=1.0, m=5.50, f_c=100.0, f_s=30.0, period=512)

        fft_signal = fft.rfft(signal)
        m_fft_signal = np.abs(fft_signal)

        plt.figure(figsize=(14, 5))
        plt.title('AM-Signal')
        plt.xlabel("Sample")
        plt.ylabel("Amplitude")
        plt.plot(signal)

        plt.figure(figsize=(14, 5))
        plt.title('AM-Spectrum')
        plt.xlabel("Freq samples")
        plt.ylabel("Level")
        plt.plot(m_fft_signal)
        plt.show()
```

0.2

-:

$$s(t) = A_c \cdot \cos(2\pi f_c t + \frac{A_m f_\Delta}{f_m} \sin(2\pi f_s t))$$

,:

- A_c -- ;
- A_m -- ;

- f_c -- ;
- f_m -- ;
- f_Δ -- .

In [ ]:
```
#  FM-
import numpy as np
import matplotlib.pyplot as plt

def get_fm_signal(amp=1.0, f_d=0.25, f_c=10.0, f_s=2.0, period=100):
    """
    amp :
    f_d :   f_d < period/4
    f_c :
    f_s :
    period :
    """
    t = 2.0 * np.pi * np.linspace(0, 1, period)
    return amp * np.cos(f_c * t + f_d/f_s * np.sin(f_s * t))

signal = get_fm_signal(amp=1.0, f_d=20, f_c=60.0, f_s=5.0, period=256)

plt.figure(figsize=(14, 5))
plt.title('Signal')
plt.xlabel("Sample")
plt.ylabel("Amplitude")
plt.plot(signal)
```

In [ ]:
```
#  FM-
signal = get_fm_signal(amp=1.0, f_d=15, f_c=30.0, f_s=5.0, period=256)

fft_signal = fft.rfft(signal)
m_fft_signal = np.abs(fft_signal)

plt.figure(figsize=(14, 5))
plt.title('FM-Signal')
plt.xlabel("Sample")
plt.ylabel("Amplitude")
plt.plot(signal)

plt.figure(figsize=(14, 5))
plt.title('FM-Spectrum')
plt.xlabel("Freq samples")
plt.ylabel("Level")
plt.plot(m_fft_signal)
plt.show()
```

In [ ]:
```
#
def get_ask_signal(mod_sequence,carirer_frequency=64, width=64):
```

```
        mod_signal = np.repeat(mod_sequence, repeats=width)

        carirer_period = mod_signal.size
        return mod_signal * np.sin(width * 2.0 * np.pi * np.linspace(0, 1, carirer_period))

    mod_sequence = np.array(
        [1, 1, 1, 0, 1, 0, 0, 0, 1, 1, 0, 1, 0, 0, 0, 0, 1, 1, 0, 1, 1, 1, 1, 0, 1,
         1, 1, 0, 1, 0, 1, 0, 1, 1, 0, 0, 1, 1, 1, 0, 1, 1, 0, 1, 0, 0, 0, 0, 1, 1,
         1, 0, 1, 0, 0])

    ask_signal = get_ask_signal(mod_sequence, 32, 64)
    mod_signal = np.repeat(mod_sequence, repeats=64)

    plt.figure(figsize=(14, 5))
    plt.title('ASK-Signal')
    plt.xlabel("Sample")
    plt.ylabel("Amplitude")
    plt.plot(ask_signal)
    plt.plot(mod_ask, '--')
```

In [ ]:
```
    #
    mod_sequence = np.array(
        [1, 1, 1, 0, 1, 0, 0, 0, 1, 1, 0, 1, 0, 0, 0, 0, 1, 1, 0, 1, 1, 1, 1, 0, 1,
         1, 1, 0, 1, 0, 1, 0, 1, 1, 0, 0, 1, 1, 1, 0, 1, 1, 0, 1, 0, 0, 0, 0, 1, 1,
         1, 0, 1, 0, 0])

    def get_ask_signal(mod_sequence, width=64, mod_0_freq = 32, mod_1_freq = 128):
        mod_signal = np.repeat(mod_sequence, repeats=width)
        carirer_period = mod_signal.size
        mod_frequencies = np.zeros(carirer_period)
        mod_frequencies[mod_signal == 0] = mod_0_freq
        mod_frequencies[mod_signal == 1] = mod_1_freq
        return np.sin(mod_frequencies *  2.0 * np.pi * np.linspace(0, 1, carirer_period))

    fsk_signal = get_ask_signal(mod_sequence, width=32, mod_0_freq = 32, mod_1_freq = 128)
    mod_signal = np.repeat(mod_sequence, repeats=32)

    plt.figure(figsize=(14, 5))
    plt.title('FSK-Signal')
    plt.xlabel("Sample")
    plt.ylabel("Amplitude")
    plt.plot(fsk_signal)
    plt.plot(mod_signal, '--')
```

In [ ]:
```
    #
    mod_sequence = np.array(
        [1, 1, 1, 0, 1, 0, 0, 0, 1, 1, 0, 1, 0, 0, 0, 0, 1, 1, 0, 1, 1, 1, 1, 0, 1,
         1, 1, 0, 1, 0, 1, 0, 1, 1, 0, 0, 1, 1, 1, 0, 1, 1, 0, 1, 0, 0, 0, 0, 1, 1,
```

```
     1, 0, 1, 0, 0])


def get_psk_signal(mod_sequence, carirer_frequency=64, width=64):
    mod_signal = np.repeat(mod_sequence, repeats=width)
    carirer_period = mod_signal.size
    return np.sin(carirer_frequency * 2.0 * np.pi * np.linspace(0, 1, carirer_period) +

psk_signal = get_psk_signal(mod_sequence, carirer_frequency=16, width=32)
mod_signal = np.repeat(mod_sequence, repeats=32)

plt.figure(figsize=(14, 5))
plt.title('PSK-Signal')
plt.xlabel("Sample")
plt.ylabel("Amplitude")
plt.plot(psk_signal)
plt.plot(mod_signal, '--')
```