# EE-435 Term Project Report

**Berkay Yaldız-2232940**

**Melih Can Zerin-2233088**

Melih Can ZERİN 2233088
Berkay YALDIZ 2232940

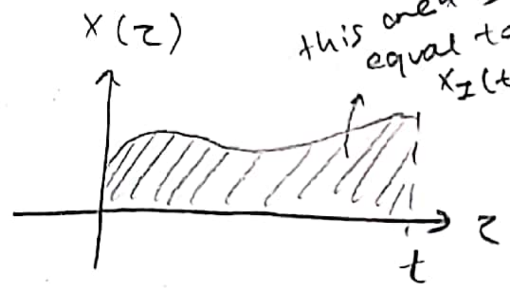
# EE 435 - TERM PROJECT

## Part 1: Discrete Representation of Continuous Signals

$x(t)$: nonzero only in the interval $0 \leq t < T$

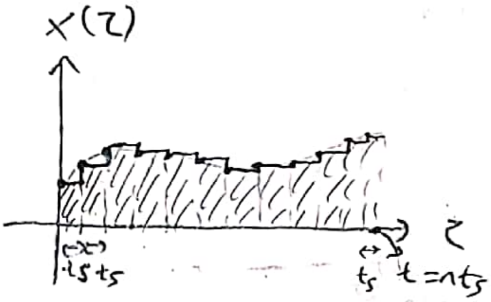
$x[n] = x(nT_s)$, $n = 0, 1, \ldots, N-1$ where $N = \frac{T}{t_s}$

a) $x_I(t) = \int_0^t x(\tau)\, d\tau$




If we divide this time interval $t$ into $n$ equal subintervals with length $t_s$, and choose an arbitrary point in each subinterval to make them rectangles (we choose the initial points of each subinterval)

$\Rightarrow$




$$\Rightarrow x_I(t) \approx \sum_{k=0}^{n-1} \underbrace{x(kt_s)}_{x[k]}\, t_s$$

$$\Rightarrow \boxed{x_I(t) \approx t_s \sum_{k=0}^{n-1} x[k]}$$

c) $$z(t) = \sum_{n=-\infty}^{\infty} \delta(t - nt_s) \xleftrightarrow{\text{CTFT}} Z(f) = \frac{1}{t_s} \sum_{k=-\infty}^{\infty} \delta\left(f - \frac{k}{t_s}\right)$$

$$x_s(t) = x(t)\, z(t) = x(t) \sum_{n=-\infty}^{\infty} \delta(t - nt_s) = \sum_{n=-\infty}^{\infty} x(nt_s)\, \delta(t - nt_s)$$

$$\xleftrightarrow{\text{CTFT}} X_s(f) = \int_{-\infty}^{\infty} x_s(t)\, e^{-j2\pi f t}\, dt = \int_{-\infty}^{\infty} \left( \sum_{n=-\infty}^{\infty} x(nt_s)\, \delta(t - nt_s) \right) e^{-j2\pi f t}\, dt$$

$$= \sum_{n=-\infty}^{\infty} x(nt_s) \int_{-\infty}^{\infty} \delta(t - nt_s)\, e^{-j2\pi f t}\, dt = \sum_{n=-\infty}^{\infty} x(nt_s)\, e^{-j2\pi f n t_s}$$

$$X(e^{jw}) = \sum_{n=-\infty}^{\infty} x[n] e^{-jwn} = \sum_{n=-\infty}^{\infty} x(nt_s) e^{-jwn}$$

$$\boxed{X_s(f) = X(e^{jw})\Big|_{w=2\pi f t_s}}$$

ii) $x_s(t) = x(t)\, z(t) \xleftrightarrow{CTFT} X_s(f) = X(f) * Z(f)$

$$\Rightarrow X_s(f) = X(f) * \frac{1}{t_s}\sum_{k=-\infty}^{\infty} \delta\left(f - \frac{k}{t_s}\right) = \frac{1}{t_s}\sum_{k=-\infty}^{\infty} X\left(f - \frac{k}{t_s}\right)$$

In order to be able to obtain $x(t)$ from $x_s(t)$ back, $x(t)$ must be bandlimited to $W$, where $\frac{1}{t_s} > 2W$. (If $X(f)=0$ at $f=\pm W$, then $\frac{1}{t_s} \geq 2W$).

iii) Assuming the condition in part ii is satisfied,

$$X_s(f) = \frac{1}{t_s}\sum_{k=-\infty}^{\infty} X\left(f - \frac{k}{t_s}\right) = X(e^{jw})\Big|_{w=2\pi f t_s}$$

Putting $f = \frac{w}{2\pi t_s}$ into the equation above,

$$\boxed{X(e^{jw}) = \frac{1}{t_s} X\left(\frac{w}{2\pi t_s} - \frac{k}{t_s}\right)}$$

Since $X(e^{jw}) = X(e^{j(w+2\pi)})$, $X(e^{jw})$ is periodic with $2\pi$ and a $2\pi$ interval of $w$ can give us every information about $X(e^{jw})$.

If the condition in part ii is satisfied, $X(f)$ is bandlimited to $|f| < \frac{1}{2t_s}$. Hence, since $w = 2\pi f t_s$, $|w| < \pi$ is the necessary and meaningful interval for $X(f)$.

$\boxed{w \in (-\pi, \pi)}$ → the interval of $w$ with maximum width that is meaningful for $X(f)$

$\boxed{f \in \left(-\frac{1}{2t_s}, \frac{1}{2t_s}\right)}$ → the corresponding interval of $f$

$\boxed{w = 2\pi f t_s}$ → the relation between $w$ and $f$

The maximum width of the $f$ interval, $F$, that we can observe $X(f)$ using $X(e^{jw})$ is

$$F = \frac{1}{2t_s} - \left(-\frac{1}{2t_s}\right) = \frac{1}{t_s} \Rightarrow \boxed{F = \frac{1}{t_s}}$$

iv) $$X[k] = X(e^{jw})\Big|_{w = k\frac{2\pi}{N_f}}, \quad k = 0, 1, \ldots, N_f - 1$$
$$(N_f \geq N)$$

$k = 0 \rightarrow w = 0$

$k = 1 \rightarrow w = \frac{2\pi}{N_f}$

$\vdots$

$k = \frac{N_f}{2} - 1 \rightarrow w = \pi - \frac{2\pi}{N_f}$

$$\left.\begin{array}{l} k = \frac{N_f}{2} \rightarrow w = \pi \rightarrow w = \pi - 2\pi = -\pi \\ k = \frac{N_f}{2} + 1 \rightarrow w = \pi + \frac{2\pi}{N_f} \rightarrow w = \pi + \frac{2\pi}{2N_f} - 2\pi = -\pi + \frac{2\pi}{N_f} \\ \vdots \\ k = N_f - 1 \rightarrow w = 2\pi - \frac{2\pi}{N_f} \rightarrow w = 2\pi - \frac{2\pi}{N_f} - 2\pi = -\frac{2\pi}{N_f} \end{array}\right\}$$

Since $X(e^{jw})$ is periodic with $2\pi$, adding or subtracting integer multiples of $2\pi$ to $w$ does not change $X(e^{jw})$. In order to get $w$ into the meaningful interval found in part iii, we subtracted $2\pi$ from $w$ values corresponding to $k$ values $\frac{N_f}{2} \leq k \leq N_f - 1$

$$\Rightarrow X[k] = \begin{cases} X(e^{jw})\Big|_{w=k\frac{2\pi}{N_f}}, & 0 \leq k \leq \frac{N_f}{2}-1 \\ X(e^{jw})\Big|_{w=(k-N_f)\frac{2\pi}{N_f}}, & \frac{N_f}{2} \leq k \leq N_f-1 \\ 0, & \text{otherwise} \end{cases}$$

The frequencies for $\frac{N_f}{2} \leq k \leq N_f - 1$ are lower than those for $0 \leq k \leq \frac{N_f}{2} - 1$. So, to sort the frequencies, we need to change the places of the left half and right half of $X[k]$.

$$\Rightarrow \boxed{X_{sorted}[k] = X\left[\left(\left(k - \frac{N_f}{2}\right)\right)_{N_f}\right], \quad 0 \leq k \leq N_f - 1}$$

v) $X_{sorted}[k] = X\left[\left(\left(k-\frac{N_f}{2}\right)\right)_{N_f}\right] = X(e^{jw})\Big|_{w=\left(k-\frac{N_f}{2}\right)\frac{2\pi}{N_f}}$

$$= \frac{1}{t_s}\sum_{l=-\infty}^{\infty} X\left(\frac{\left(k-\frac{N_f}{2}\right)/N_f}{t_s} - \frac{l}{t_s}\right)$$

$$= \frac{1}{t_s} X\left(\frac{\left(k-\frac{N_f}{2}\right)/N_f}{t_s}\right), \quad k = 0, 1, \ldots, N_f - 1$$

$$= \boxed{\frac{1}{t_s} X\left(k \cdot \frac{1}{N_f t_s} - \frac{1}{2t_s}\right)} = \frac{1}{t_s} X\left(k f_s - \frac{F}{2}\right)$$

$$\Rightarrow f_s = \frac{1}{N_f t_s} \qquad F = \frac{1}{t_s}$$

$f_s$ is dependent of $N_f$.

d) Observing a signal with limited observation time corresponds to multiplying it with a rect function. Since, in this case, we are observing the signal in the interval $0 \leq t < T$, we need to multiply the original signal expression with $\mathrm{rect}\left(\frac{t-\frac{T}{2}}{T}\right)$. The Fourier transform of

$\mathrm{rect}\left(\frac{t-\frac{T}{2}}{T}\right)$ is $T\,\mathrm{sinc}(Tf) \cdot e^{-j2\pi f \frac{T}{2}}$.

Multiplying two signals in time domain corresponds to convolution of their Fourier transforms in frequency domain. Let the signal to be observed in $0 \leq t < T$ be $x(t)$ and its Fourier transform be $X(f)$.

$$x(t)\,\mathrm{rect}(t) \xleftrightarrow{CTFT} X(f) * \mathrm{sinc}(Tf)\, e^{-j2\pi f \frac{T}{2}}$$

If $x(t) = a$, $a \in \mathbb{R}$, $X(f) = a\delta(f)$

$$\Rightarrow a\,\mathrm{rect}(t) \xleftrightarrow{CTFT} a\,\delta(f) * \mathrm{sinc}(Tf)\, e^{-j2\pi f \frac{T}{2}}$$
$$= a\,\mathrm{sinc}(Tf)\, e^{-j2\pi f \frac{T}{2}}$$

The impulses are turned into sinc functions whose main lobes are located at where the impulses are originally located.

## Part 1: Discrete Representation of Continuous Signals

Part b and c-vi are given in the code at Appendix.

## Part 2: MATLAB Code

```
function  m_t = message_signal_generator(varargin)
% First input is signal type, second input is f_m, third input is time vector.
signal_type = varargin{1};
f_m = varargin{2};
t = varargin{3};
switch signal_type
    case 1
        m_t = (cos(2*pi*f_m*t));
    case 2
       m_t = (sawtooth(2*pi*f_m*t, 0.5));
    case 3
        m_t = (square(2*pi*f_m*t));
    case 4
        m_t = (cos(2*pi*f_m/2*t) + cos(2*pi*f_m*t));
end
end
```

*Published with MATLAB® R2020b*

```
function [M_f, x_t, X_f, S_f, B_exp] = fm_generator(varargin)
% First variable is m_t.
% Second variable is k_f.
% Third variable is T value.
% Fourth variable is t_s.
% Fifth variable is f_c.
% Sixth variable is A_c.
% Seventh variable is N_f.


%For the purpose of project, we gave just first two inputs in our main script
%and made rest of the variables  same with given values in the project,
% but they can be changed via giving different inputs. In order to have a
% correct analysis, T and t_s values should be given accordingly. (They
% should be same when m_t was constructed, so we took it as it was stated
% in main script; however, for different size of inputs(time vectors or
% sampling rates) we still leave it as variable.
m_t = varargin{1};
k_f = varargin{2};
% These variables have default  values, but if input is given different we
% change it in switch statement.
T = 1;
f_c=20e3;
t_s=1e-6;
N_f = 5e6;
A_c = 1;
switch nargin
    case 3
        T = varargin{3};
    case 4
        T = varargin{3};
        t_s = varargin{4};
    case 5
        T = varargin{3};
        t_s = varargin{4};
        f_c = varargin{5};

    case 6
        T = varargin{3};
        t_s = varargin{4};
        f_c = varargin{5};
        A_c = varargin{6};
    case 7
        T = varargin{3};
        t_s = varargin{4};
        f_c = varargin{5};
        A_c = varargin{6};
        N_f = varargin{7};
end
N = T / t_s;
f_s=(N/N_f)*(1/T);
t = (0:t_s:T-t_s).';
F = 1/t_s;
f=((-F/2):f_s:((F/2)-f_s)).';

length_of_message_signal = length(m_t); % Length of the message signal.

m_t_integral = cumsum(m_t)*t_s; % Integrated signal.

M_f= fft(m_t,N_f)./ length_of_message_signal;
M_f= fftshift(M_f); % Two sided fft of message signal.

x_t=A_c* cos(2*pi*f_c*t+2*pi*k_f*m_t_integral); % FM modulated signal.

X_f= fft(x_t,N_f)./ length_of_message_signal;
X_f= fftshift(X_f); % Two sided fft of FM modulated signal.

S_f=(abs(X_f).^2)/T; % PSD of FM modulated signal.

P=sum(S_f)*f_s/2; % One side of the spectrum is examined.
B_exp=0; % Initial guess for Bexp is 0.
P_eff=0;
% Increase symmetrically indexes to achieve 98% power
% and find effective bandwidth.
S_b=S_f;
[~, power_frequency_fc] = min ( abs( f - f_c ) );
while (P_eff/P) < 0.98 % Power is symmetrical to fc frequency.
    s_b_1= power_frequency_fc - B_exp/ (2*f_s);
    s_b_2 = power_frequency_fc + B_exp/ (2*f_s);
    P_eff=sum(S_b(s_b_1:s_b_2))*f_s;
    B_exp=B_exp+ 1;
end
end
```

**Part 3: FM Signal and Spectrum**

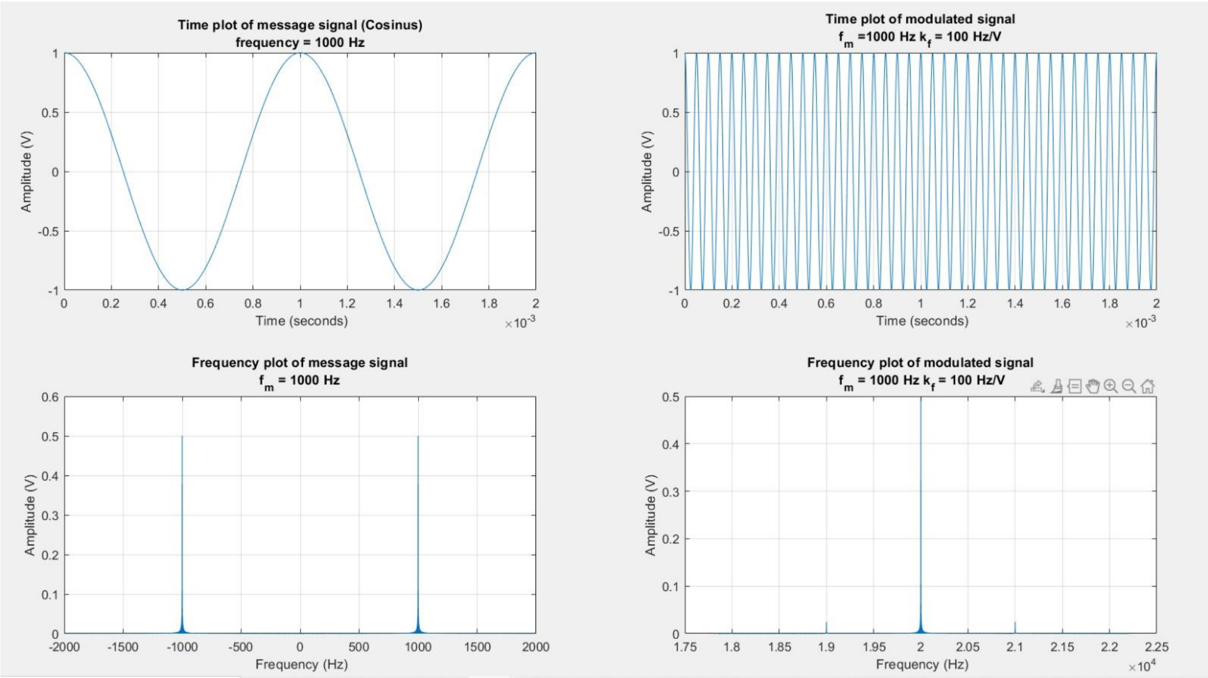
$\beta$, $B_c$ $B_{exp}$ values for each five cases are as follows:

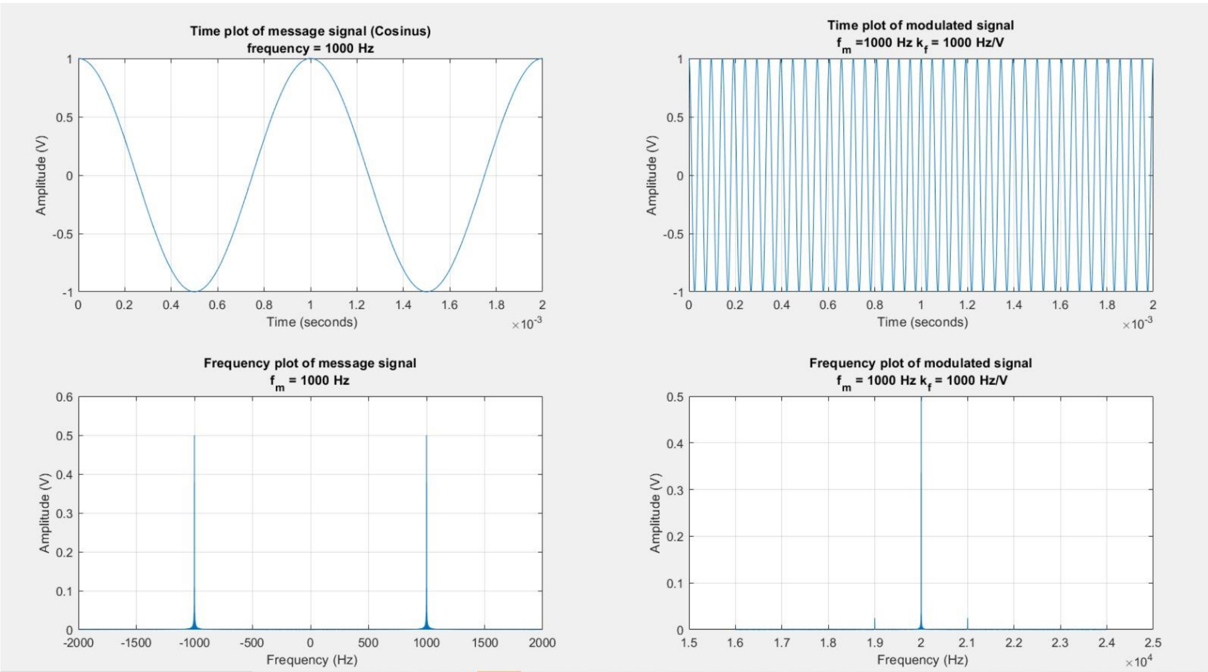
| | $\beta$ | $B_c$ (Hz) | $B_{exp}$ (Hz) |
|---|---|---|---|
| a | 0.1 | 2200 | 15 |
| b | 1 | 4000 | 4001 |
| c | 10 | 22000 | 22002 |
| d | 0.5 | 6000 | 4002 |
| e | 1 | 8000 | 8001 |

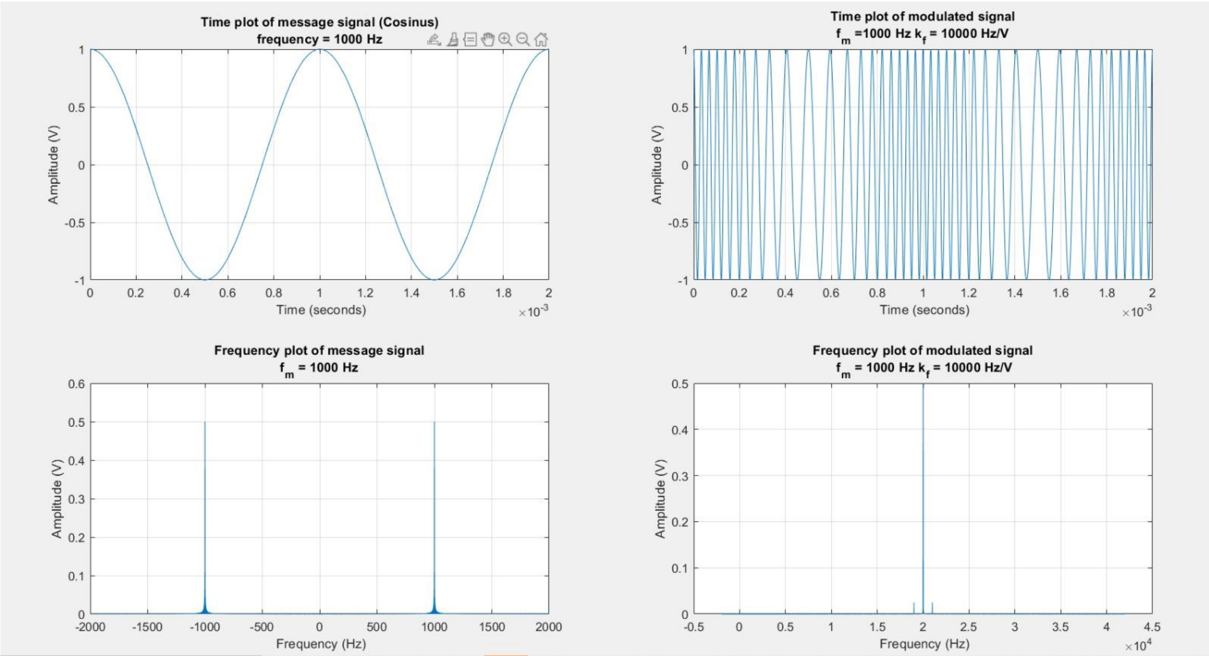
$B_c$ and $B_{exp}$ values are almost the same for the signals in b, c and e. However, for the other signals, they are not consistent. Carson's rule says that $\beta$ +1 sidebands have the significant power. For the signal in part a, $\beta$ is 0.1, which is much lower than 1, and $\beta$+1 is approximately 1. However, since the magnitudes of the impulses in the sideband are equal to $\beta/4$ for the NBFM single tone signals with amplitude 1, they do not have significant power and all the significant power is contained in the impulse at $f=f_c$. In part d, $\beta$=0.5, which is still lower than 1 but closer to 1 compared to part a, $B_c$ and $B_{exp}$ are still not close to each other but closer to each other compared to part a. For the other parts, $\beta$ values are at least 1 and the experimental bandwidths, $B_{exp}$, are almost the same as the theoretical ones, $B_c$ , found by using Carson's rule.

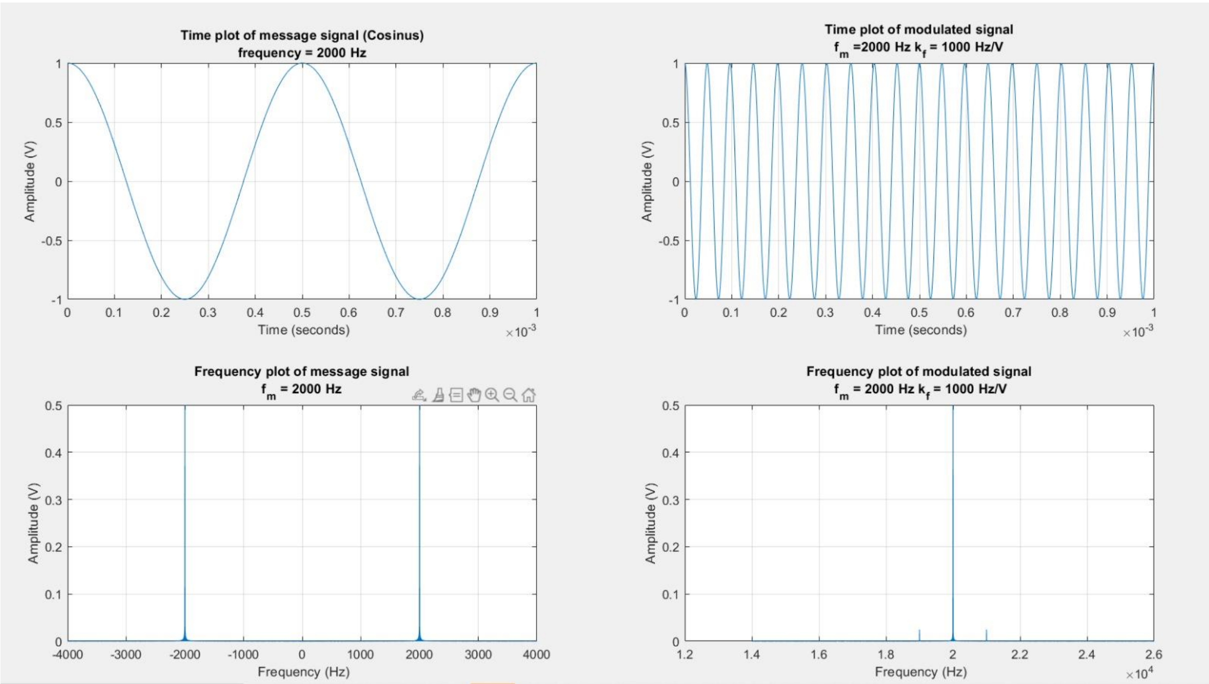
Signal with higher $k_f$ has higher amplitudes for the same sideband frequencies. Since signal with higher $k_f$ value has more bandwidth, we see a wider frequency range on the x axis. So there are more impulses for the second signal, even if we cannot see in the plot.

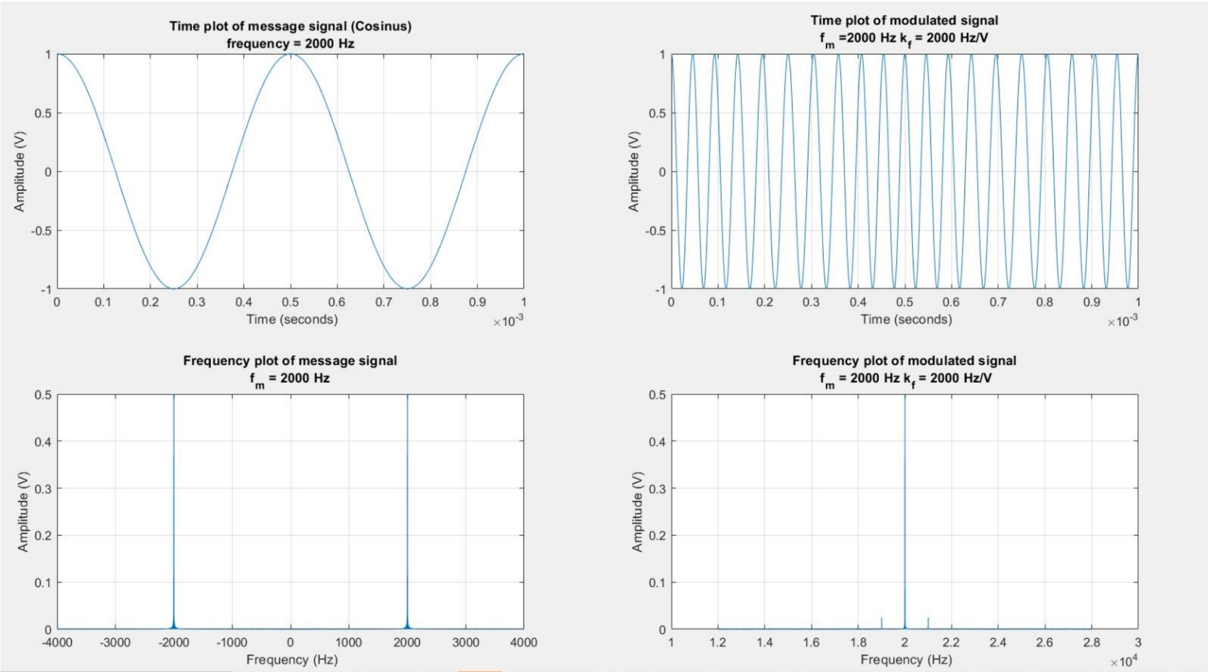
As we can see from the plots, higher $k_f$ valued signals have a more spread bandwidth for the same $f_m$ values, but $\beta$ values are also important for the calculation of effective bandwidth. For the same beta valued signals whose $k_f$ is higher than the other has a higher frequency (actually in this case it is two times higher) spectrum as expected.

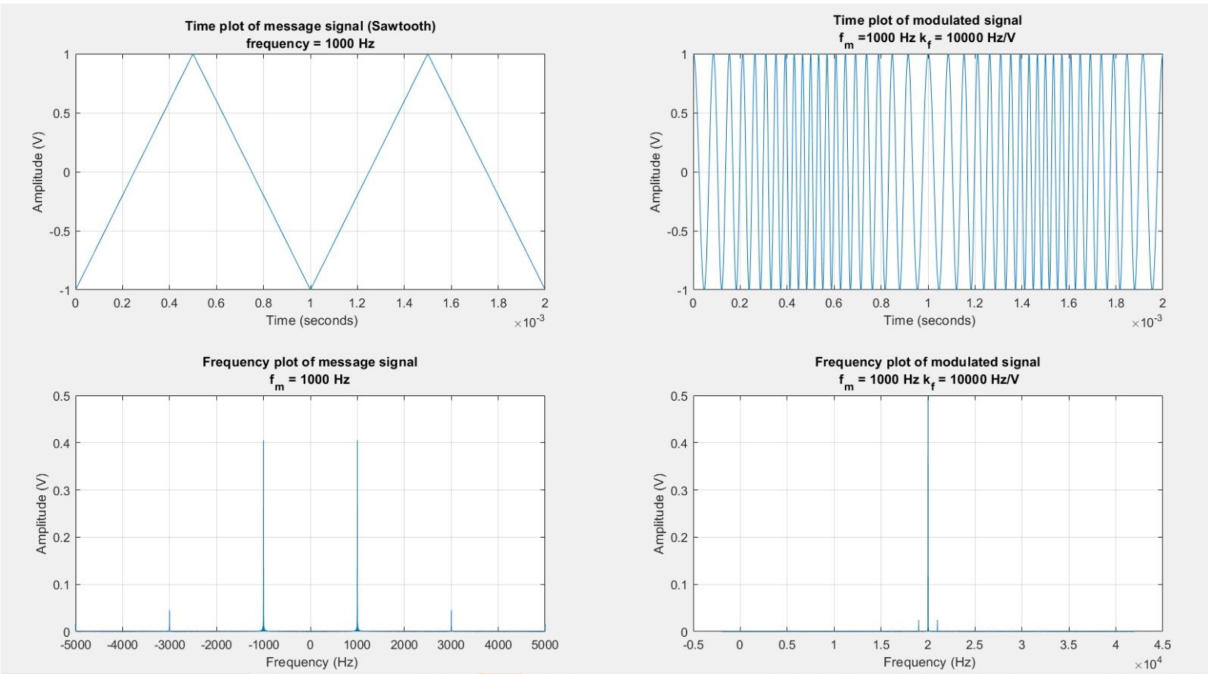

Time plot of message signal (Cosinus)
frequency = 1000 Hz
Amplitude (V)
Time (seconds)
Time plot of modulated signal
f_m =1000 Hz k_f = 100 Hz/V
Amplitude (V)
Time (seconds)
Frequency plot of message signal
f_m = 1000 Hz
Amplitude (V)
Frequency (Hz)
Frequency plot of modulated signal
f_m = 1000 Hz k_f = 100 Hz/V
Amplitude (V)
Frequency (Hz)

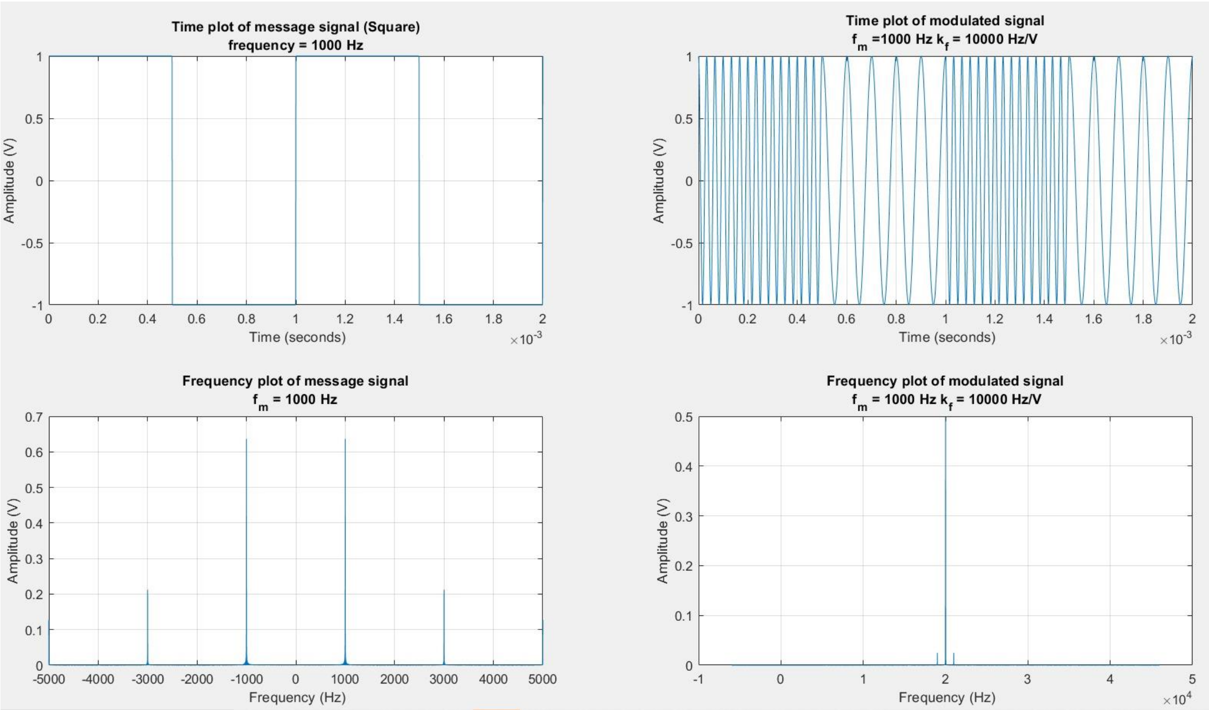

Time plot of message signal (Cosinus)
frequency = 1000 Hz
Amplitude (V)
Time (seconds)
Time plot of modulated signal
f_m =1000 Hz k_f = 1000 Hz/V
Amplitude (V)
Time (seconds)
Frequency plot of message signal
f_m = 1000 Hz
Amplitude (V)
Frequency (Hz)
Frequency plot of modulated signal
f_m = 1000 Hz k_f = 1000 Hz/V
Amplitude (V)
Frequency (Hz)


Time plot of message signal (Cosinus)
frequency = 1000 Hz
Amplitude (V)
Time (seconds)
×10^-3
Time plot of modulated signal
f_m =1000 Hz k_f = 10000 Hz/V
Amplitude (V)
Time (seconds)
×10^-3
Frequency plot of message signal
f_m = 1000 Hz
Amplitude (V)
Frequency (Hz)
Frequency plot of modulated signal
f_m = 1000 Hz k_f = 10000 Hz/V
Amplitude (V)
Frequency (Hz)
×10^4


Time plot of message signal (Cosinus)
frequency = 2000 Hz
Amplitude (V)
Time (seconds)
×10^-3
Time plot of modulated signal
f_m =2000 Hz k_f = 1000 Hz/V
Amplitude (V)
Time (seconds)
×10^-3
Frequency plot of message signal
f_m = 2000 Hz
Amplitude (V)
Frequency (Hz)
Frequency plot of modulated signal
f_m = 2000 Hz k_f = 1000 Hz/V
Amplitude (V)
Frequency (Hz)
×10^4


Time plot of message signal (Cosinus)
frequency = 2000 Hz
Amplitude (V)
Time (seconds)
Time plot of modulated signal
f_m =2000 Hz k_f = 2000 Hz/V
Amplitude (V)
Time (seconds)
Frequency plot of message signal
f_m = 2000 Hz
Amplitude (V)
Frequency (Hz)
Frequency plot of modulated signal
f_m = 2000 Hz k_f = 2000 Hz/V
Amplitude (V)
Frequency (Hz)


Time plot of message signal (Sawtooth)
frequency = 1000 Hz
Amplitude (V)
Time (seconds)
Time plot of modulated signal
f_m =1000 Hz k_f = 10000 Hz/V
Amplitude (V)
Time (seconds)
Frequency plot of message signal
f_m = 1000 Hz
Amplitude (V)
Frequency (Hz)
Frequency plot of modulated signal
f_m = 1000 Hz k_f = 10000 Hz/V
Amplitude (V)
Frequency (Hz)




## Part 4: Wideband FM

An FM signal is of the following form:

$$u(t) = A_c.\cos(2\pi f_c t + 2\pi k_f \int_{-\infty}^{t} m(\tau)d\tau)$$

where $A_c$ is the amplitude of the carrier, $f_c$ is the carrier frequency, $k_f$ is the frequency sensitivity of the modulator and m(t) is the message signal.

Instantaneous frequency of this FM signal is given by the following equation:

$$f_i(t) = f_c + k_f.m(t)$$

According to Woodward's theorem, the PSD of u(t) is given as follows:

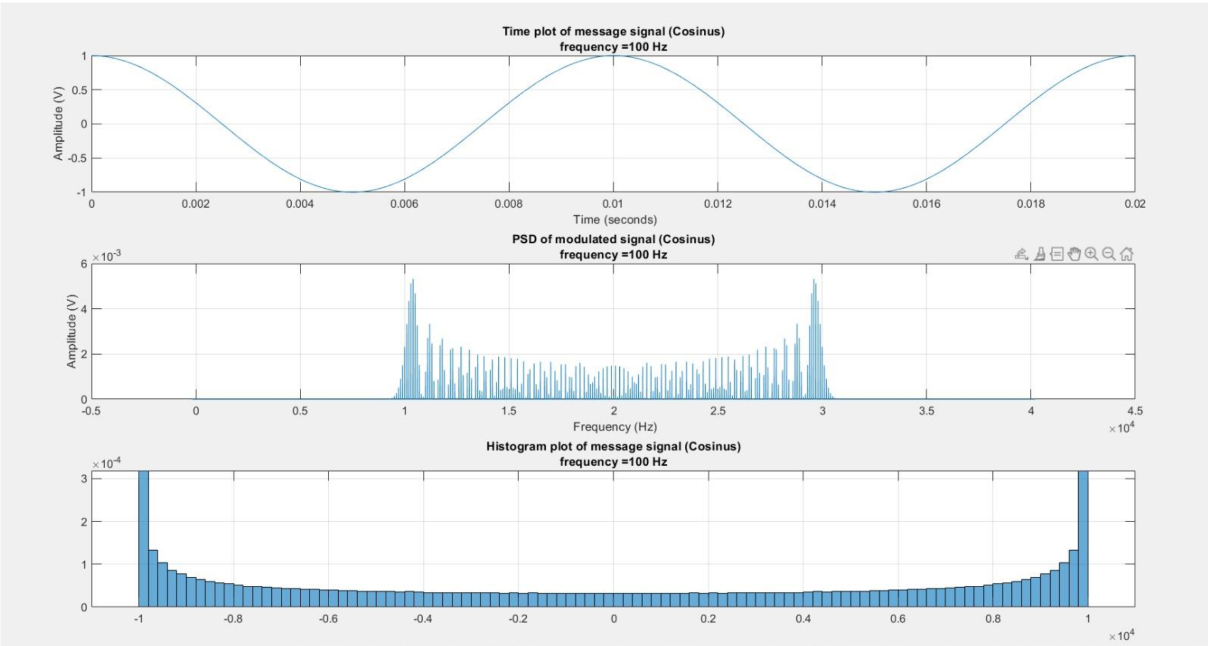
$$S_U(f) = \frac{{A_c}^2}{4}.[p_f(f) + p_f(-f)]$$

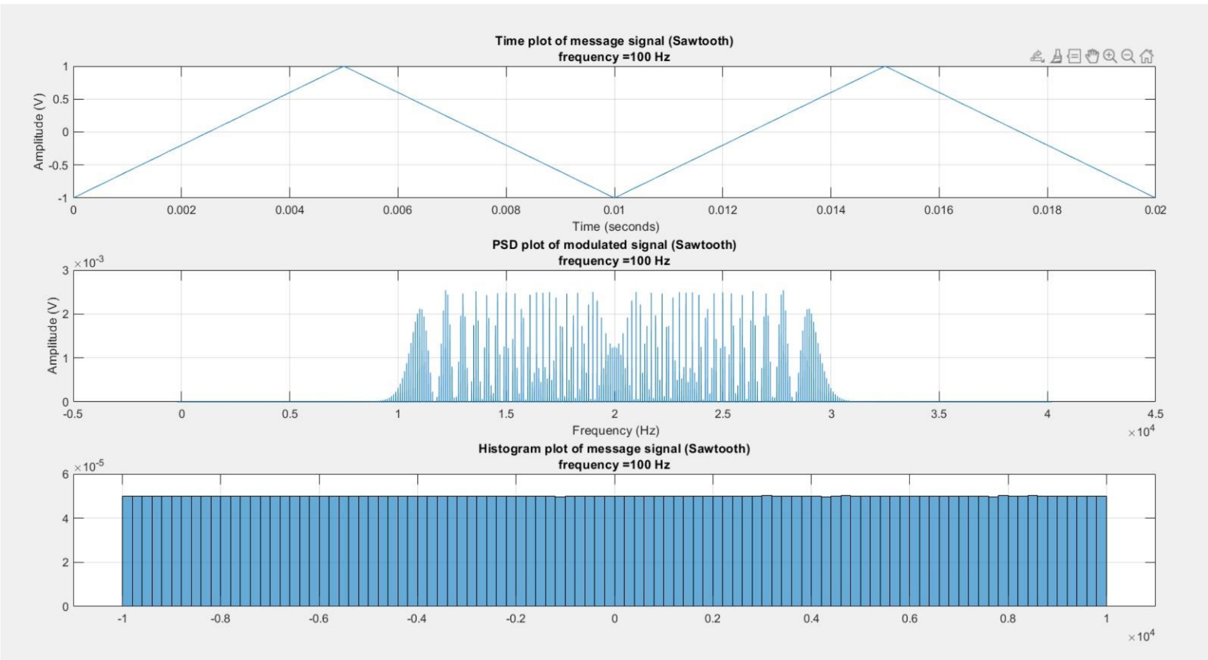
where $p_f$ is the probability density of $f_i$(t). Since $f_i$(t) has the same probability distribution with $k_f$.m(t) with the shifted mean by $f_c$ , $S_U(f)$ has the same shape with $k_f$.m(t) at the center frequency of f=$f_c$=20 kHz and f=-$f_c$=-20 kHz with a scaling factor of $\frac{{A_c}^2}{4}$.

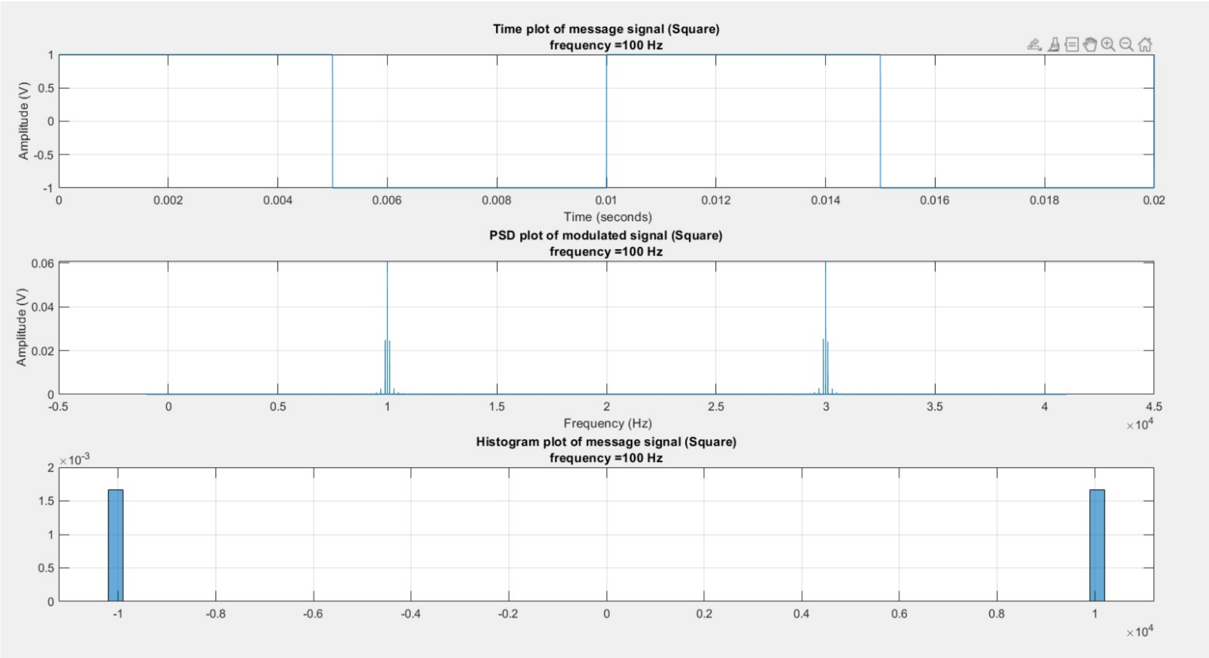
Since the histogram of $k_f$.m(t) shows the probability distribution of it without normalizing, we normalized the histogram to obtain the probability density of $k_f$.m(t).

As can be seen from S(f) plots, since they are plotted for the positive frequencies, they have the same shape with the normalized histogram of $k_f.m(t)$ at a center frequency $f=f_c=20$ kHz.

Histogram shows distribution of data and gives a clue about for pdf. As we can see for the cosine signal, its shape is U (as expected). For the sawtooth wave, we see a triangular wave in the time domain which resembles to a linear line and from its histogram it can be derived that data is distributed uniformly (uniform pdf). For the square wave, histogram output is only at the edges, we would expect that because square wave takes only two values as -1 and 1.






## Part 5: Armstrong's Method for FM Generation

**a)** An FM signal is in the following form:

$$A_c.\cos(2\pi f_c t + 2\pi k_f \int_{-\infty}^{t} m(\tau)d\tau)$$

where $A_c$ is the amplitude of the carrier, $f_c$ is the carrier frequency, $k_f$ is the frequency sensitivity of the modulator and m(t) is the message signal. This expression can be written as follows:

$$A_c.\cos(2\pi f_c t)\cos\left(2\pi k_f \int_{-\infty}^{t} m(\tau)d\tau\right) - A_c.\sin(2\pi f_c t)\sin\left(2\pi k_f \int_{-\infty}^{t} m(\tau)d\tau\right)$$

In narrowband FM signals, the modulation index , which is $\beta = \frac{k_f.\max(|m(t)|)}{f_m}$, where $f_m$ is the bandwidth of m(t), is significantly small and this yields to the following:

$$\cos\left(2\pi k_f \int_{-\infty}^{t} m(\tau)d\tau\right) \approx 1$$

$$\sin\left(2\pi k_f \int_{-\infty}^{t} m(\tau)d\tau\right) \approx 2\pi k_f \int_{-\infty}^{t} m(\tau)d\tau$$

Therefore, for NBFM signals, the FM signal expression becomes:

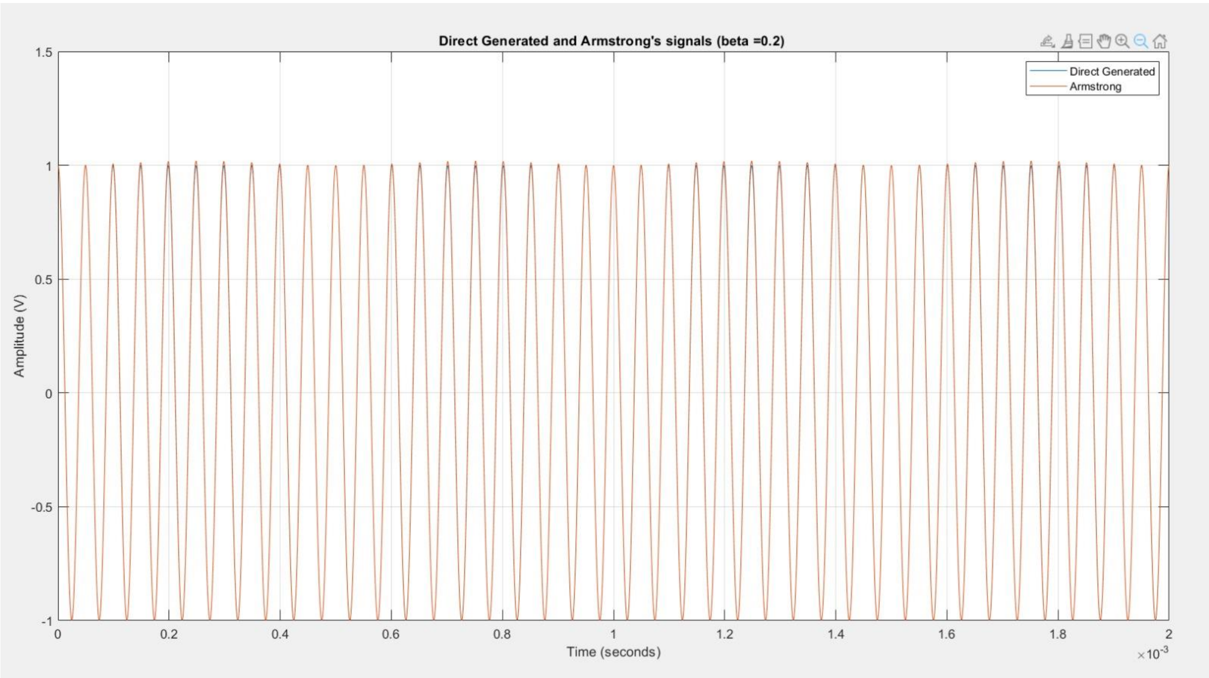
$$A_c.\cos(2\pi f_c t) - A_c.\sin(2\pi f_c t).\,2\pi k_f \int_{-\infty}^{t} m(\tau)d\tau$$

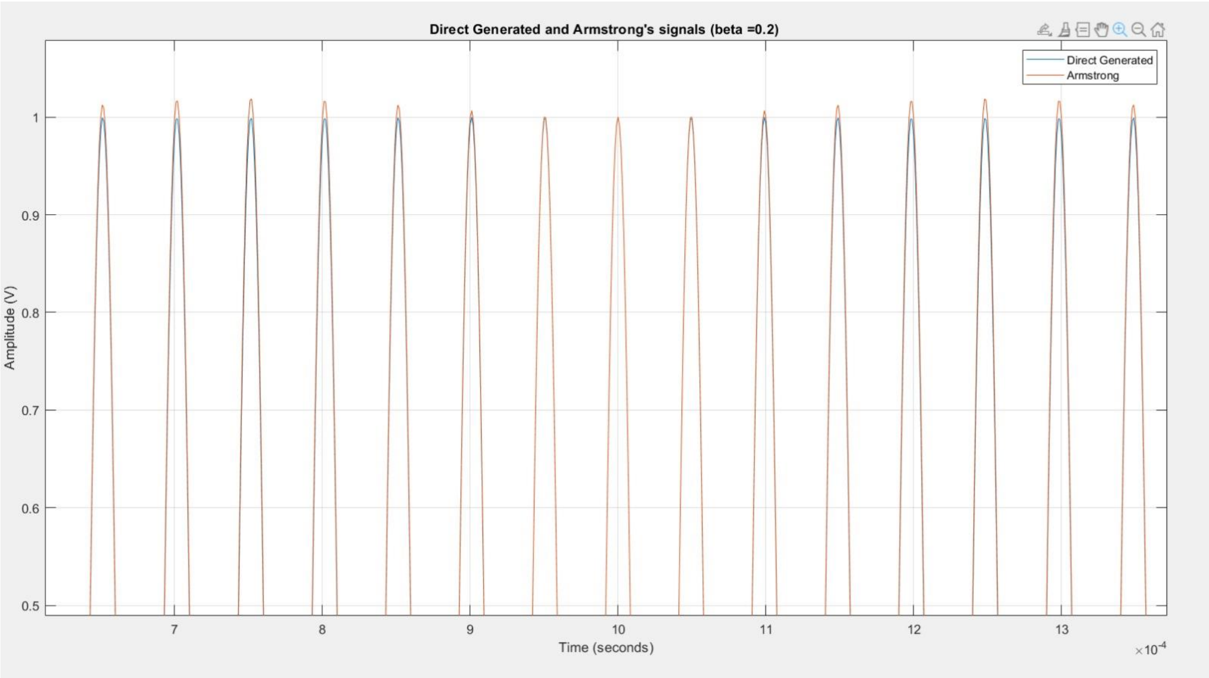
For the single tone cosine NBFM signal with frequency $f_m$ and amplitude 1, this expression is as follows:

$$A_c.\cos(2\pi f_c t) - A_c.\sin(2\pi f_c t).\,\beta.\sin(2\pi f_m t)$$

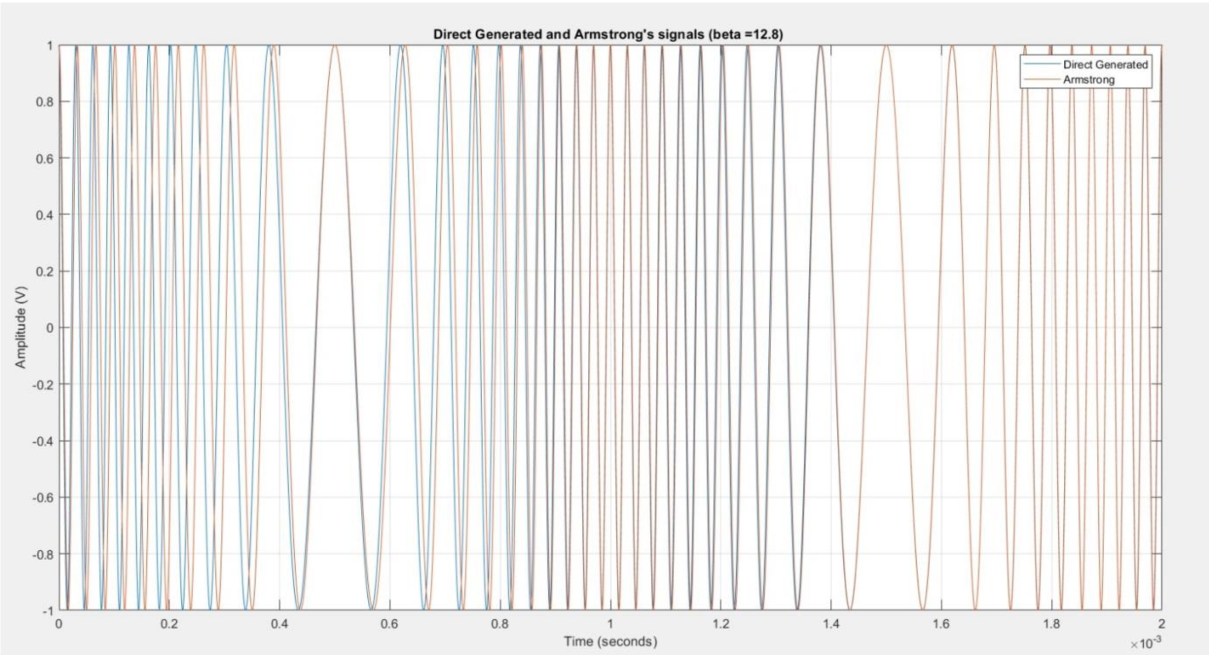
**b)** Since Armstrong's method uses PLL circuit, it has to lock in to the frequency after a while when it started to work. Since we use the doubler for once , the output is very close to the output of the direct generation FM.

**c)** We can see the difference between x(t) and $x_{Armstrong}$(t) more clear when β is much more higher (12.8 in this case). This is due to the fact that we used the doubler 7 times in this case. For the small values of t, there is a phase difference and after a while the outputs become almost the same. This means that the stable equilibrium point is delayed. The delay is 1 ms in this case.

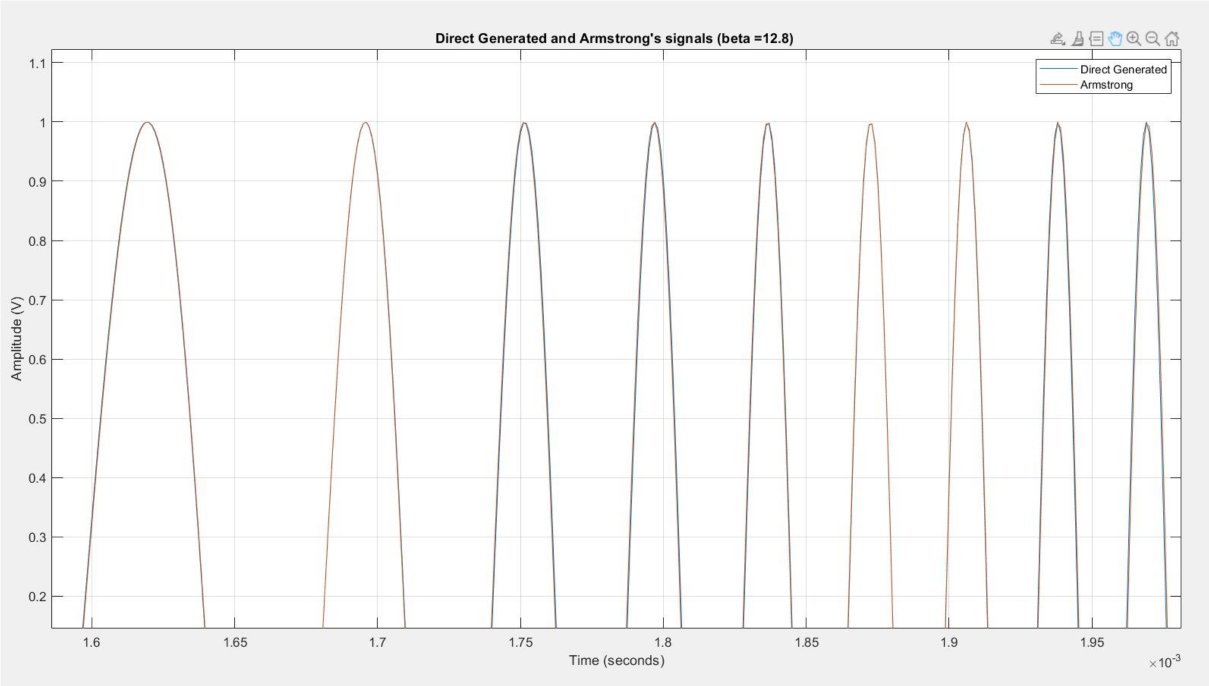


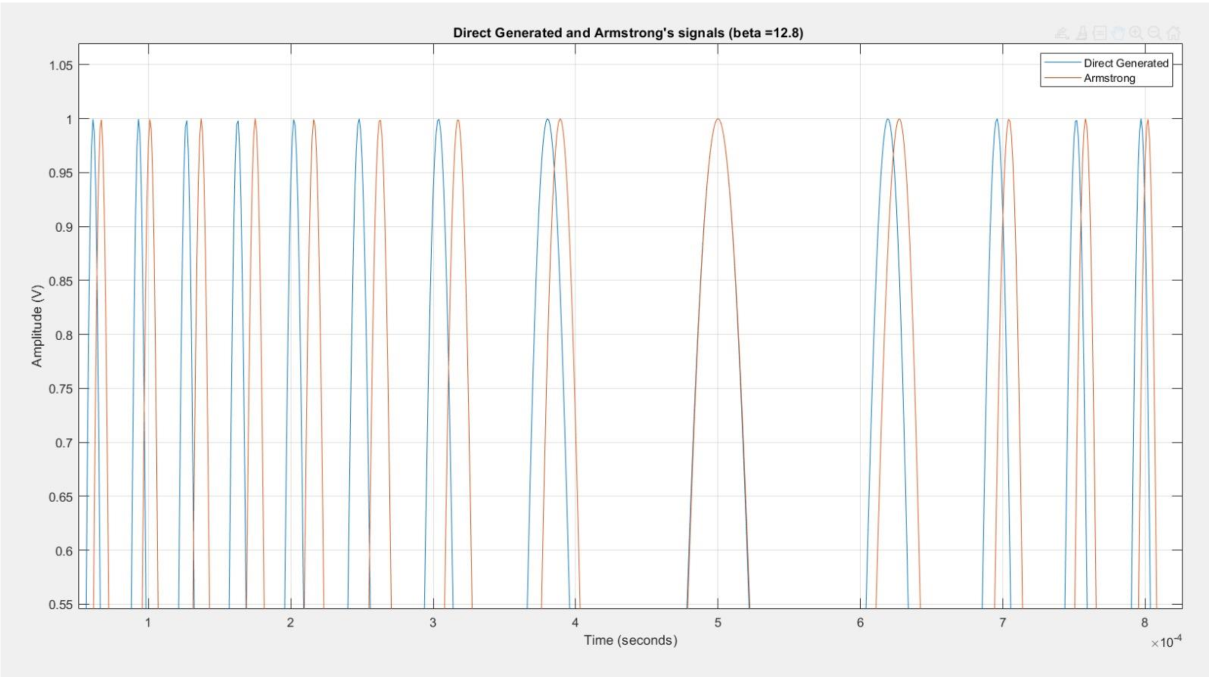


Direct Generated and Armstrong's signals (beta =0.2)
Direct Generated
Armstrong
Amplitude (V)
Time (seconds)
1
0.9
0.8
0.7
0.6
0.5
7
8
9
10
11
12
13
×10-4

Direct Generated and Armstrong's signals (beta =12.8)
Direct Generated
Armstrong
Amplitude (V)
Time (seconds)
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1
0
0.2
0.4
0.6
0.8
1
1.2
1.4
1.6
1.8
2
×10-3




## Part 6: Digital Modulation

**a)** The instantaneous frequency of an FM signal is given by the following expression:

$$f_i(t) = f_c + k_f . m(t)$$

The instantaneous frequency of the modulated signal x(t) is as follows:

$$f_i(t) = \begin{cases} f_1 \, , if \; a_k = -1 \\ f_2 \, , if \; a_k = 1 \end{cases}$$

By equating these two equations above, m(t) is obtained as follows:

$$m(t) = \begin{cases} \dfrac{f_1 - f_c}{k_f}, if\ a_k = -1 \\ \dfrac{f_2 - f_c}{k_f}, if\ a_k = 1 \end{cases}$$

To make m(t) zero mean, $f_c$ should be $\frac{f_1+f_2}{2} = \frac{18\ kHz+22kHz}{2} = 20\ kHz$. For m(t) to have values between +1 and -1, $k_f$=2 kHz/V should be taken.

With these values, m(t) is as follows:

$$m(t) = \begin{cases} -1, if\ a_k = -1 \\ +1, if\ a_k = 1 \end{cases}$$

**b) iii)** The experimental bandwidth, $B_{exp}$ , is measured as 7566 Hz. This value is not equal to $f_2$-$f_1$=4000 Hz. This is due to the fact that the effective bandwidth of an FM signal is found by Carson's Rule as $B_c = 2(\Delta f + W)$ , where Δf=$k_f$.$A_m$ and W is the bandwidth of the message signal m(t). In this case, Δf=(2 kHz/V).1V=2 kHz and W is approximately equal to $\frac{1}{T_s} = \frac{1}{250\mu s} = 4\ kHz$ , where $T_s$ is one symbol time, which is given as 250 µs. Therefore, by Carson's Rule, effective bandwidth $B_c$ is found approximately as 8 kHz. The experimental bandwidth, $B_{exp}$ is close to that value.

**c)** QPSK modulated signal expression is given in the question as follows:

$$x(t) = \cos\left(2\pi f_c t + a_k \frac{\pi}{2}\right), for\ kT_s < t < (k+1)T_s$$

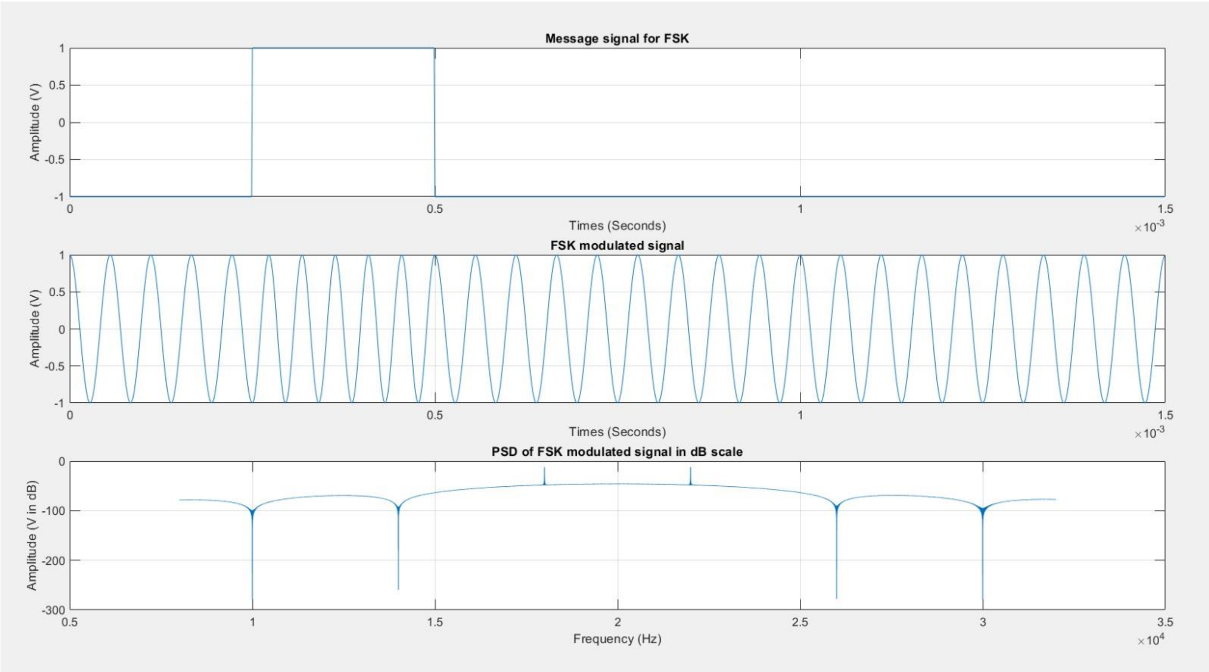
The general PM signal expression is as follows:

$$x(t) = \cos\left(2\pi f_c t + k_p . m(t)\right)$$

Equating these two expressions above, we obtain $k_p = \frac{\pi}{2}$ and m(t) as follows:

$$m(t) = a_k\ , for\ kT_s < t < (k+1)T_s$$

where $a_k$ takes values of 0,1,2 and 3 with equal probabilities.


Message signal for FSK
Amplitude (V)
Times (Seconds)
×10^-3
FSK modulated signal
Amplitude (V)
Times (Seconds)
×10^-3
PSD of FSK modulated signal in dB scale
Amplitude (V in dB)
Frequency (Hz)
×10^4

# EE-435 Term Project, Berkay Yaldiz 2232940, Melih Can Zerin 2233088

## Table of Contents



# Part 1

Part b We found at part a) that we need to sum all the rectangle heights and multiply it with interval delta Tau. Let x(t) = t^2 and T = 10.

```
clear
close all
T = 10;
a = 0;
b = T;
n = 1000000;% As n increases error gets smaller.
delta = (b-a)/n;
t = a:delta:b;
X = t.*t;
Area = cumsum(X) *delta ;
p=cumtrapz(t,X); % Matlab's built in function.
error=Area-p;
clearvars -except error
% Part c, vi
% Since we just need a 2pi interval for DFT frequencies, after taking
 fft; fftshift can be
% used to represent DFT outputs between -pi and pi intervals (Sampling
 frequency normalized to 2*pi).
```

# Part 2

We wrote according to given conventions. We used column vectors for signals. message_signal_generator.m and fm_generator.m functions are the answers for this part.

```
Ac = 1;
T = 1;
t_s = 1e-6;

N = T / t_s;
```

```
N_f = 5e6;

f_s=(N/N_f)*(1/T);
F = 1/t_s;

f=((-F/2):f_s:((F/2)-f_s)).';

t = (0:t_s:T-t_s).';

f_c = 20e3;
```

# Part 3

```
signal_number = 7; % Number of signals can be changed here, but f_m
 and k_f inputs should be entered by hand!
signal_type_part3 = 1; % Signal type input ( 1 for cos, 2 for
 sawtooth, 3 for square, 4 for sum of 2 cosine signals)
f_m = [ repmat(1e3,1,3) repmat(2e3, 1, 2) repmat(1e3,1,2) ]; % 7
 different signal parameters in part 3 (a-b-c-d-e-f-g)
k_f  = [ 0.1e3 1e3 10e3 1e3 2e3 10e3 10e3];
theoretical_beta = zeros(1, signal_number); % Theoritical beta values.
B_c = zeros(1,signal_number);
% Now we will construct signals with our function in a for loop, each
% column represents a, b, c, d, e, f, g signals respectively.

% Initialize used variables for speed.
length_of_signal = length(t);
m_t = zeros(length_of_signal ,signal_number);
x_t = zeros(length_of_signal , signal_number);

M_f = zeros(N_f, signal_number);
X_f = zeros(N_f, signal_number);
S_f = zeros( N_f, signal_number);
P = zeros(1, signal_number);
P_eff= zeros(1, signal_number);
B_exp=zeros(1, signal_number);
for ii=1: signal_number
    if isequal(ii,6)
        signal_type_part3 = 2; % Make message signal sawtooth.
    elseif isequal(ii,7)
        signal_type_part3 = 3; % Make message signal square.
    end
    m_t(:,ii) =
 message_signal_generator(signal_type_part3,f_m(ii),t); % First input
 is signal type, second input is f_m, third input is time vector.
    [M_f(:,ii), x_t(:,ii), X_f(:,ii), S_f(:,ii), B_exp(:,ii)] =
 fm_generator(m_t(:,ii), k_f(ii) );

    theoretical_beta(ii) = k_f(ii) / f_m(ii); % A_m is 1.
    B_c(ii) = 2* f_m(ii) * (theoretical_beta(ii)  + 1);

    % Take interval between  -2*fm and 2*fm for plotting M(f) and X(f)
 and
```

```
   % 0 - 2/f_m for time plot index.
   [~, index_time] = min( abs ( t - ( 2 / f_m(ii) ) ) );
   if  ii < 6
       [~, M_f_index1] = min( abs ( f - (-2*f_m(ii) ) ) ); % We used
min-abs approach because there was a rounding-precision problem while
determining the index.
       [~, M_f_index2] = min( abs ( f - (2*f_m(ii) ) ) ) ;
   else
       [~, M_f_index1] = min ( abs( f - (-5 * f_m(ii) ) ) );
       [~, M_f_index2] = min ( abs ( f - (5*f_m(ii) ) ) );
   end

   [~, X_f_index1] = min ( abs( f - ( f_c - max( B_exp(ii),
B_c(ii) ) ) ) );
   [~, X_f_index2] = min ( abs ( f - ( f_c + max( B_exp(ii),
B_c(ii) ) ) ) );

   figure('Position',[0 0 1920 1080]) % In order to plot figures with
full size.
   subplot(2,2,1)
   plot(t(1:index_time), m_t(1:index_time,ii) )
   grid on
   if ii < 6
       title({'Time plot of message signal (Cosinus)', ['frequency =
',num2str(f_m(ii)) ,' Hz'] } )
   elseif isequal(ii,6)
       title({'Time plot of message signal (Sawtooth)', ['frequency =
',num2str(f_m(ii)) ,' Hz'] } )
   else
       title({'Time plot of message signal (Square)', ['frequency =
',num2str(f_m(ii)) ,' Hz'] } )
   end
   ylabel('Amplitude (V)')
   xlabel(' Time (seconds)')

   subplot(2,2,2)
   plot(t(1:index_time), x_t(1:index_time,ii) )
   grid on
   title({ 'Time plot of modulated signal', [ 'f_m
=',num2str(f_m(ii)) ,' Hz', ' k_f = ',num2str(k_f(ii)), ' Hz/V' ] } )
   ylabel('Amplitude (V)')
   xlabel(' Time (seconds)')

   subplot(2,2,3)
   plot( f ( M_f_index1:M_f_index2), abs( M_f
( M_f_index1:M_f_index2,ii) ) )
   grid on
   if ii < 6
       title( { 'Frequency plot of message signal' , ['f_m =
',num2str(f_m(ii)) ,' Hz'] } )
   elseif isequal(ii,6)
       title( { 'Frequency plot of message signal' , ['f_m =
',num2str(f_m(ii)) ,' Hz'] } )
   else
```

```
        title( { 'Frequency plot of message signal' , ['f_m =
 ',num2str(f_m(ii)) ,' Hz'] } )
    end
    ylabel('Amplitude (V)')
    xlabel(' Frequency (Hz)')

    subplot(2,2,4)
    plot( f ( X_f_index1:X_f_index2), abs(X_f
 ( X_f_index1:X_f_index2) ) ) ;
    grid on
    title( { 'Frequency plot of modulated signal',[' f_m =
 ',num2str(f_m(ii)) ,' Hz k_f = ', num2str(k_f(ii)), ' Hz/V'] } )
    ylabel('Amplitude (V)')
    xlabel(' Frequency (Hz)')
end
```

# Part 4

```
k_f4 = 10e3;
f_m4 = 100;
signal_number_part4 = 3;

m_t_part4 = zeros(length_of_signal, signal_number_part4);
B_c_part4 = zeros(1,signal_number_part4);
theoretical_beta_part4 = zeros(1,signal_number_part4);
S_f_part4 = zeros(N_f, signal_number_part4);
B_exp_part4 = zeros(1,signal_number_part4);
for ii=1 : signal_number_part4
    m_t_part4(:,ii) = message_signal_generator(ii, f_m4, t); % ii=1 -
 cos,  ii=2 -sawtooth, ii=3 square. If signal number changes, please
 change the input to message signal genetor accordingly.
    [~,~,~,S_f_part4(:,ii), B_exp_part4(ii)] =
 fm_generator(m_t_part4(:,ii), k_f4);% We do not need x_t, X_f, M_f
 for this part.

    theoretical_beta_part4(ii) = k_f4 / f_m4; % A_m is 1.
    B_c_part4(ii) = 2* f_m4 * (theoretical_beta_part4(ii)  + 1);

    figure('Position',[0 0 1920 1080])
    subplot(3,1,1)
    [~,time_index] =min ( abs ( t - 2 ./ f_m4) );
    plot( t (1 : time_index), m_t_part4 (1: time_index,ii) ) % Plot
 between 0 and 2/f_m
    if isequal(ii,1)
        title( { 'Time plot of message signal (Cosinus)', ['frequency
 =',num2str(f_m4) ,' Hz'] } )
    elseif isequal(ii,2)
        title( { 'Time plot of message signal (Sawtooth)', ['frequency
 =',num2str(f_m4) ,' Hz'] } )
    else
        title( { 'Time plot of message signal (Square)', ['frequency
 =',num2str(f_m4) ,' Hz'] } )
    end
```

```
   ylabel('Amplitude (V)')
   xlabel(' Time (seconds)')
   grid on
   subplot(3,1,2)
   [~, freq_index_part4_1] = min ( abs( f - ( f_c -
max( B_exp_part4(ii), B_c_part4(ii) ) ) ) );
   [~, freq_index_part4_2] = min ( abs( f - ( f_c +
max( B_exp_part4(ii), B_c_part4(ii) ) ) ) );
   plot( f (freq_index_part4_1 :
freq_index_part4_2 ),S_f_part4(freq_index_part4_1 :
freq_index_part4_2,ii)  )
   if isequal(ii,1)
       title( { 'PSD of modulated signal (Cosinus)', ['frequency
=',num2str(f_m4) ,' Hz'] } )
   elseif isequal(ii,2)
       title( { 'PSD plot of modulated signal (Sawtooth)',
['frequency =',num2str(f_m4) ,' Hz'] } )
   else
       title( { 'PSD plot of modulated signal (Square)', ['frequency
=',num2str(f_m4) ,' Hz'] } )
   end
   ylabel('Amplitude (V)')
   xlabel(' Frequency (Hz)')
   grid on
   subplot(3,1,3)
   myhistogram=histogram( k_f4 * m_t_part4(:,ii));
   myhistogram.Normalization='pdf'; % Since histogram is generally
used to calculate pdf of a signal, we normalized it.
   grid on
   if isequal(ii,1)
       title( { 'Histogram plot of message signal (Cosinus)',
['frequency =',num2str(f_m4) ,' Hz'] } )
   elseif isequal(ii,2)
       title( { 'Histogram plot of message signal (Sawtooth)',
['frequency =',num2str(f_m4) ,' Hz'] } )
   else
       title( { 'Histogram plot of message signal (Square)',
['frequency =',num2str(f_m4) ,' Hz'] } )
   end

end
```

# Part 5

Part a We can write FM signal expression for NBFM as follows ( EE435_angle modulation) x_t_NBFM_c=cos(2*pi*f_c*t)-((sin(2*pi*f_c*t))*2*pi*k_f_5_NBFM.*m_t_5_integral);

```
% Part b
f_m_5=1e3;
m_t_5 = message_signal_generator(1,f_m_5,t);
beta_5_b=0.2;
k_f_5_b=beta_5_b*f_m_5;
[~,x_t_5_b,~,~,~]=fm_generator(m_t_5,k_f_5_b);
```

```
beta_5_NBFM=0.1; % Since NBFM beta is half of the required beta, we
 selected NBFM_fc as fc/2 (10kHz).
f_c_NBFM1 = f_c/2;
k_f_5_NBFM=beta_5_NBFM*f_m_5;
x_t_NBFM_b=cos(2*pi*f_c_NBFM1*t)-
beta_5_NBFM*(sin(2*pi*f_c_NBFM1*t)) .* sin(2*pi*f_m_5*t);
x_armstrong_t_b=2*((x_t_NBFM_b.^2)-0.5);
[~,time_index_part5] = min( abs ( t - ( 2 / f_m_5 ) ) );
figure
plot( t (1:time_index_part5), x_t_5_b( 1:time_index_part5) )
hold on;
plot(t (1:time_index_part5) , x_armstrong_t_b(1:time_index_part5));
title('Direct Generated and Armstrong''s signals (beta =0.2) ')
ylabel('Amplitude (V)')
xlabel('Time (seconds)')
legend('Direct Generated','Armstrong')
hold off;
grid on
error_for_NBFM1 = mean((abs(x_t_5_b-x_armstrong_t_b)));
% Mean error for part b can be seen above (beta = 0.2).

% Part c
beta_5_c=12.8;
k_f_5_c=beta_5_c*f_m_5;
[~,x_t_5_c,~,~,~]=fm_generator(m_t_5,k_f_5_c);

beta_5_NBFM2 = 0.1;
f_c_NBFM2 = f_c/(2^7); % % Since NBFM beta is 1/128 of the required
 beta, we selected NBFM_fc as fc/128 (0.1563) kHz.
k_f_5_NBFM2=beta_5_NBFM2*f_m_5;
x_t_NBFM_c=cos(2*pi*f_c_NBFM2*t)-
beta_5_NBFM2*(sin(2*pi*f_c_NBFM2*t)) .* sin(2*pi*f_m_5*t);
x_armstrong_t_c = x_t_NBFM_c;
for ii = 1:7
    x_armstrong_t_c=2*((x_armstrong_t_c.^2)-0.5);
end
figure
plot( t(1:time_index_part5), x_t_5_c(1:time_index_part5) ) % Plot from
 0 to 2/fm.
hold on;
plot( t(1:time_index_part5) , x_armstrong_t_c(1:time_index_part5) )
hold off;
title('Direct Generated and Armstrong''s signals (beta =12.8) ')
ylabel('Amplitude (V)')
xlabel('Time (seconds)')
hold off;
grid on
legend('Direct Generated','Armstrong')
error_for_NBFM2 = mean((abs((x_t_5_c-x_armstrong_t_c))));
% Mean error for part b can be seen above (beta = 12.8).
% As we can see from the plot, Armstrong's method catches the original
% signal after nearly 1ms.
```

# Part 6

Part a

```
f1_part6 = 18e3;
f2_part6 = 22e3;
Ts = 250e-6;
A_c_6 = 1;
f_c_6 = 20e3;
kf_6 = 2e3;
t_6 = (0:t_s:(Ts-t_s)).';
N_s = 4e3;
realization_number = 1000;
X_f_part6 = zeros( N_f,1);
symbol_length=length(t_6);
x_t_6 = zeros( N_s*length(t_6),1);
m_t_6 = zeros( N_s*length(t_6), 1);
X_f_6 = zeros( N_f, 1); % FFT length is N_f = 5e6.
S_f_6_temp = zeros( N_f, 1);
a_k = randi( [0 1], N_s, realization_number);
zero_indexes = a_k ==0;
a_k(zero_indexes) = -1;
time_vector_part6 = zeros(N_s*length(t_6),1);
% Consider the following lines for speed.
for kk = 1: N_s
time_vector_part6( symbol_length*(kk-1) + 1 : (kk) *symbol_length) =
 ( kk-1).*Ts + t_6; % construct time vector for each a_k.
end % Construct time vector just once.

symbol_1 = (f1_part6 - f_c_6) ./ kf_6 *ones(symbol_length,1);
symbol_2 = (f2_part6 - f_c_6) ./ kf_6 *ones(symbol_length,1);
length_of_signal_part6 = length(x_t_6);
% Calculate just one time these three parameters.

for ii = 1:realization_number
    for kk = 1: N_s
        if a_k(kk,ii) == -1
            x_t_6(symbol_length*(kk-1) + 1 : (kk) *symbol_length) =
 cos( 2 * pi * f1_part6 * time_vector_part6(symbol_length*(kk-1) + 1 :
 (kk) *symbol_length) );
            if ii == 1000 % Consider only one realization of message
 signal. (Last one, also true for modulated signal.)
                m_t_6(symbol_length*(kk-1) + 1 : (kk) *symbol_length)
 = symbol_1 ;
            end
        else
            x_t_6(symbol_length*(kk-1) + 1 : (kk) *symbol_length) =
 cos( 2 * pi * f2_part6 * time_vector_part6(symbol_length*(kk-1) + 1 :
 (kk) *symbol_length) );
            if ii == 1000
                m_t_6(symbol_length*(kk-1) + 1 : (kk) *symbol_length)
 = symbol_2;
            end
```

```
        end
    end
    X_f_part6 = fft(x_t_6,N_f) ./ length_of_signal_part6 ;  % Since
 trials will be long, we did not use our function and directly
 calculated N_f points FFT.
    S_f_6_temp = S_f_6_temp + (abs(X_f_part6).^2);
end
time_index_part6 = 1.5e-3 / t_s;
figure
subplot(3,1,1)
plot( t(1:time_index_part6), m_t_6(1:time_index_part6) )
title('Message signal for FSK')
xlabel('Times (Seconds)')
ylabel('Amplitude (V)')
grid on

subplot(3,1,2)
plot( t(1:time_index_part6), x_t_6(1:time_index_part6) )
title('FSK modulated signal')
xlabel('Times (Seconds)')
ylabel('Amplitude (V)')
grid on

[~, freq_index_part6_1] = min ( abs( f - 8e3 ) );
[~, freq_index_part6_2] = min ( abs( f -  32e3  ) );
S_f_6_temp = fftshift(S_f_6_temp);
S_f_6 = S_f_6_temp / T / realization_number;
subplot(3,1,3)
plot( f ( freq_index_part6_1: freq_index_part6_2),
 10*log10(S_f_6(freq_index_part6_1:freq_index_part6_2)))
title('PSD of FSK modulated signal in dB scale')
xlabel('Frequency (Hz)')
ylabel('Amplitude (V in dB)')
grid on

% Part iii- Calculation of Bexp:
P_6 = sum(S_f_6 )*f_s /2; % One sided Power.
P_eff_6 = 0;
B_exp_6 = 0;
[~, power_frequency_fc] = min ( abs( f - f_c ) );
S_b=S_f_6;
while (P_eff_6/P_6)<0.98 % Power is symmetrical to fc frequency.
    s_b_part6_1= power_frequency_fc - B_exp_6/(2*f_s);
    s_b_part6_2 = power_frequency_fc + B_exp_6 / (2*f_s);
    P_eff_6=sum(S_b(s_b_part6_1:s_b_part6_2))*f_s;
    B_exp_6=B_exp_6+1;
end
```

*Published with MATLAB® R2020b*