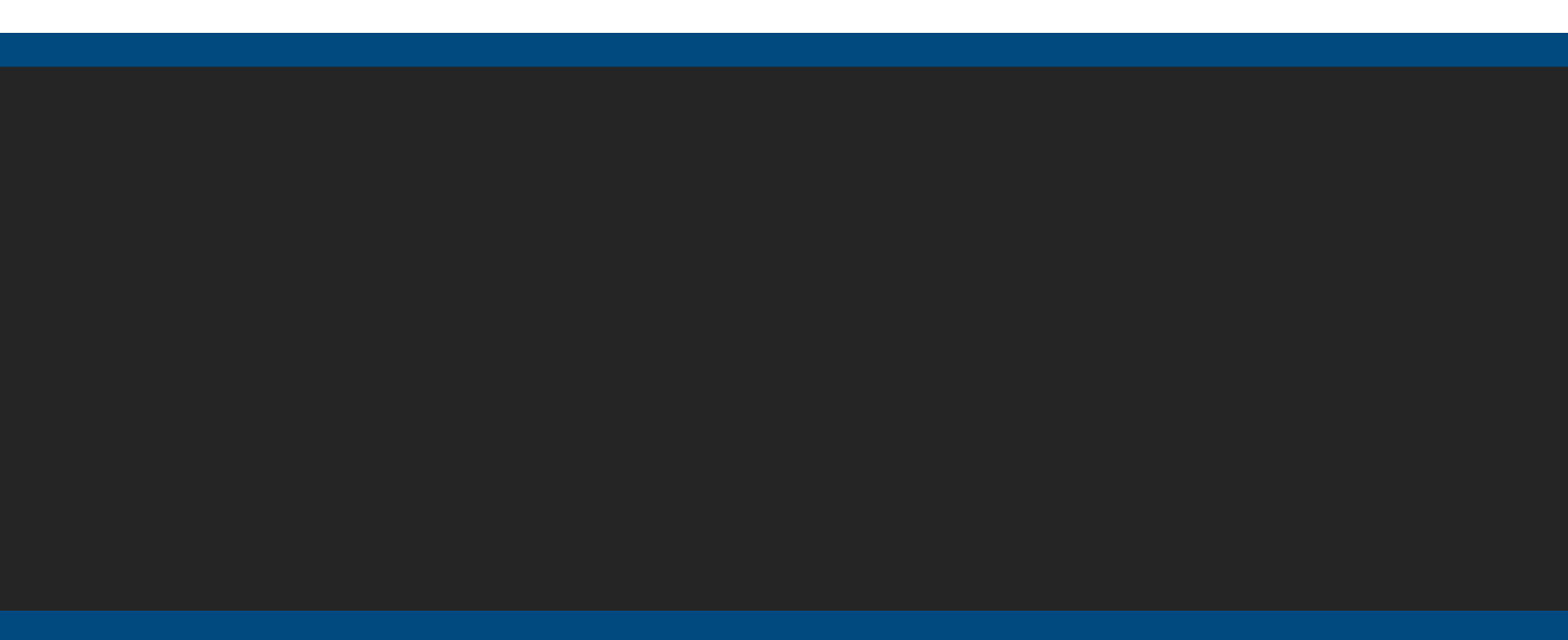


Communication Project



Ziad Sherif Muhammed BN:27 Sec:1

Eslam Ashraf Ibrahim BN:13 Sec:1

1. Our Work Explanation:

Firstly, we read three signals with different sounds and make them have the same frequency sampling =250000 by make resemble to them and find the length of each one along with max length in-order to be able to sum all modulated signal of each input. We make all audios have the same length by adjusting all of them by adding zero to make have same length. Secondly got time and frequencies intervals. Then we calculated omega **(ω1, ω2)** by **ω=2\*pi\*const** in frequency domain. Then we got carrier in cos and sin domain

**Carrier Signal One=cos (2\*pi\* ω1)**

**Carrier Signal Two=sin (2\*pi\* ω2)**

Modulation:

1. Calculate modulated signal by multiplying signal and carrier
2. Calculate Fourier transform of modulated signal returned from step 1
3. Calculate phase of modulated signal returned from step 1
4. Sum all modulated signals of all audios
5. FFT to summation of modulated signals
6. Calculate phase of modulated signals
7. Calculate frequency band pass which is used in de-modulation

De-modulation:

1. Calculate demodulated signal by multiplying carrier and modulation signal
2. Calculate low pass filter used frequency sampling.
3. Perform demodulation three times with phase shifts of 10, 30, 90 degrees for both carriers **note** **(before each phase shift calculate carrier phase)**.
4. perform demodulation two times with a local carrier frequency that is different by 2 Hz and 10 Hz from its carrier frequency.

Required results and answers to questions:

**Input signals**

|  |  |
| --- | --- |
| Signal 1 |  |
| Signal 2 |  |
| Signal 3 |  |

**Modulation:**

w1 = 2\*pi\*500000

w1 = 2\*pi\*1000000

|  |  |
| --- | --- |
| Signal 1 |  |
| Signal 2 |  |
| Signal 3 |  |

Synchronous Demodulation:

|  |  |
| --- | --- |
| Signal 1 |  |
| Signal 2 |  |
| Signal 3 |  |

Comment:

Multiply output \*2 because **M(t)[output] =0.5\*M(t)[original]**

**Output is less weaker than original sound**

**Demodulation with phase shift 10:**

|  |  |
| --- | --- |
| Signal 1 |  |
| Signal 2 |  |
| Signal 3 |  |

Comment:

**Interference in sounds between signal 2 and signal 3 when make demodulation to them.**

**Signal 1 become little weaker.**

**Demodulation with phase shift 30:**

|  |  |
| --- | --- |
| Signal 1 |  |
| Signal 2 |  |
| Signal 3 |  |

Comment:

**More Interference in sounds between signal 2 and signal 3 when make demodulation to them.**

**Signal 1 become Weaker.**

**Demodulation with phase shift 90:**

|  |  |
| --- | --- |
| Signal 1 |  |
| Signal 2 |  |
| Signal 3 |  |

Comment:

**When make demodulation to signal 3 output is signal 2 and make demodulation to signal 2 output is signal3**

**Signal 1 be equal zero (no sound)**

**Demodulation with a local carrier frequency that is different by 2 Hz:**

|  |  |
| --- | --- |
| Signal 1 |  |
| Signal 2 |  |
| Signal 3 |  |

Comment:

**Interference in sounds between signal 2 and signal 3 when make demodulation to them.**

**All signals have been occurred distortion on them**

**Demodulation with a local carrier frequency that is different by 10 Hz:**

|  |  |
| --- | --- |
| Signal 1 |  |
| Signal 2 |  |
| Signal 3 |  |

Comment:

**More Interference in sounds between signal 2 and signal 3 when make demodulation to them.**

**All signals have been occurred more distortion on them**

**Codes:**

%% Read Voice

[signal\_1, frequency\_sampling1] = audioread('signals/ziad.wav');

[signal\_2, frequency\_sampling2] = audioread('signals/esoo.wav');

[signal\_3, frequency\_sampling3] = audioread('signals/mohey.wav');

%% Resampling the signals

FsNew = 250000;

frequency\_sampling=FsNew;

ratio=FsNew/frequency\_sampling1;

[temp\_P, temp\_Q] = rat(ratio);

signal\_1 = resample(signal\_1, temp\_P, temp\_Q);

ratio=FsNew/frequency\_sampling2;

[temp\_P, temp\_Q] = rat(ratio);

signal\_2 = resample(signal\_2, temp\_P, temp\_Q);

ratio=FsNew/frequency\_sampling3;

[temp\_P, temp\_Q] = rat(ratio);

signal\_3 = resample(signal\_3, temp\_P, temp\_Q);

%% Sum first and second channel

signal\_1 = signal\_1(:, 1)+signal\_1(:, 2);

signal\_2 = signal\_2(:, 1)+ signal\_2(:, 2);

signal\_3 = signal\_3(:, 1)+signal\_3(:, 2);

%% set maxuim length between 3 signals

length\_1 = length(signal\_1);

length\_2 = length(signal\_2);

length\_3 = length(signal\_3);

N = max([length\_1, length\_2, length\_3]);

%% make 3 signals be same

signal\_1 = [signal\_1;zeros(N-length\_1, 1)];

signal\_2 = [signal\_2;zeros(N-length\_2, 1)];

signal\_3 = [signal\_3;zeros(N-length\_3, 1)];

%% get time and frequencyuency interval

t = linspace(0, N/frequency\_sampling, N);

frequency\_d = frequency\_sampling/2;

frequency = -frequency\_d : frequency\_sampling/N: frequency\_d - frequency\_sampling/N;

%% frequencyuency Domin

signal\_1\_fft = fftshift(fft(signal\_1));

signal\_2\_fft = fftshift(fft(signal\_2));

signal\_3\_fft = fftshift(fft(signal\_3));

phase\_signal\_1 = unwrap(angle(signal\_1\_fft));

phase\_signal\_2 = unwrap(angle(signal\_2\_fft));

phase\_signal\_3 = unwrap(angle(signal\_3\_fft));

%% Carrie in frequencyuency Domin

Carrier\_frequencyuency\_1 = 50000;

Carrier\_frequencyuency\_2 = 2\*50000;

WC\_1 = 2\*pi \* Carrier\_frequencyuency\_1;

WC\_2 = 2\*pi \* Carrier\_frequencyuency\_2;

%% Carrier in time Domain

carrier\_signal\_1 = cos(WC\_1 \* t);

carrier\_signal\_2 = cos(WC\_2 \* t);

carrier\_signal\_3 = sin(WC\_2 \* t);

phase\_carrier\_signal\_1 = unwrap(angle(fftshift(fft(carrier\_signal\_1))));

phase\_carrier\_signal\_2 = unwrap(angle(fftshift(fft(carrier\_signal\_2))));

phase\_carrier\_signal\_3 = unwrap(angle(fftshift(fft(carrier\_signal\_3))));

%% Modulation

modulatedSignal\_t1 = signal\_1' .\* carrier\_signal\_1;

modulatedSignal\_t2 = signal\_2' .\* carrier\_signal\_2;

modulatedSignal\_t3 = signal\_3' .\* carrier\_signal\_3;

modulated\_signal\_f1 = fftshift(fft(modulatedSignal\_t1));

modulated\_signal\_f2 = fftshift(fft(modulatedSignal\_t2));

modulated\_signal\_f3 = fftshift(fft(modulatedSignal\_t3));

phase\_modulation\_1 = unwrap(angle(modulated\_signal\_f1));

phase\_modulation\_2 = unwrap(angle(modulated\_signal\_f2));

phase\_modulation\_3 = unwrap(angle(modulated\_signal\_f3));

%% Add modulated signals

Modulated\_Signal = modulatedSignal\_t1 + modulatedSignal\_t2 + modulatedSignal\_t3;

Modulated\_Signal\_f = fftshift(fft(Modulated\_Signal));

phase\_mod = unwrap(angle(Modulated\_Signal\_f));

frequncy\_BandPass =24000;

frequncy\_BandPass2 =24000;

%% Synchronous Modulation for 3 signals

FileNames=["Out\_1" "Out\_2" "Out\_3"];

Carries=[carrier\_signal\_1 ;carrier\_signal\_2; carrier\_signal\_3];

for i = 1:3

if i == 1

LowPass=demodulation(Modulated\_Signal,frequncy\_BandPass,FileNames(i), Carries(i,:,:), frequency\_sampling,i+6,t);

else

LowPass=demodulation(Modulated\_Signal,frequncy\_BandPass2,FileNames(i), Carries(i,:,:), frequency\_sampling,i+6,t);

end

end

%%

% Phase Shift: 10

Carries\_phase\_10 = CarriersPhase(Carrier\_frequencyuency\_1, Carrier\_frequencyuency\_2, 10,t);

FileNames\_phases\_10=["Out\_1\_phase\_10" "Out\_2\_phase\_10" "Out\_3\_phase\_10"];

for i = 1:3

demodulation(Modulated\_Signal,frequncy\_BandPass,FileNames\_phases\_10(i), Carries\_phase\_10(i,:,:), frequency\_sampling,i+9,t);

end

% Phase Shift: 30

Carries\_phase\_30 = CarriersPhase(Carrier\_frequencyuency\_1, Carrier\_frequencyuency\_2, 30,t);

FileNames\_phases\_30=["Out\_1\_phase\_30" "Out\_2\_phase\_30" "Out\_3\_phase\_30"];

for i = 1:3

demodulation(Modulated\_Signal,frequncy\_BandPass,FileNames\_phases\_30(i), Carries\_phase\_30(i,:,:), frequency\_sampling,i+13,t);

end

% Phase Shift: 90

Carries\_phase\_90 = CarriersPhase(Carrier\_frequencyuency\_1, Carrier\_frequencyuency\_2, 90,t);

FileNames\_phases\_90=["Out\_1\_phase\_90" "Out\_2\_phase\_90" "Out\_3\_phase\_90"];

for i = 1:3

if i == 1

demodulation(Modulated\_Signal,frequncy\_BandPass,FileNames\_phases\_90(i), Carries\_phase\_90(i,:,:), frequency\_sampling,i+17,t);

else

demodulation(Modulated\_Signal,frequncy\_BandPass2,FileNames\_phases\_90(i), Carries\_phase\_90(i,:,:), frequency\_sampling,i+17,t);

end

end

%%

% local carrier frequencyuency different than Fc by 2 Hz in Demodulation

Carries\_shift\_2 = CarriersDifferentFc(Carrier\_frequencyuency\_1, Carrier\_frequencyuency\_2, 2,t);

FileNames\_shift\_2=["Out\_1\_shift\_2" "Out\_2\_shift\_2" "Out\_3\_shift\_2"];

for i = 1:3

demodulation(Modulated\_Signal,frequncy\_BandPass,FileNames\_shift\_2(i), Carries\_shift\_2(i,:,:), frequency\_sampling,i+21,t);

end

% local carrier frequencyuency different than Fc by 10 Hz in Demodulation

Carries\_shift\_10 = CarriersDifferentFc(Carrier\_frequencyuency\_1, Carrier\_frequencyuency\_2, 10,t);

FileNames\_shift\_10=["Out\_1\_shift\_10" "Out\_2\_shift\_10" "Out\_3\_shift\_10"];

for i = 1:3

demodulation(Modulated\_Signal,frequncy\_BandPass,FileNames\_shift\_10(i), Carries\_shift\_10(i,:,:), frequency\_sampling,i+25,t);

end

% Ploting Signals

plot\_signal(t, frequency.', phase\_signal\_1,signal\_1, abs(signal\_1\_fft),1, 'Signal 1');

plot\_signal(t, frequency.', phase\_signal\_2,signal\_2, abs(signal\_2\_fft),2, 'Signal 2');

plot\_signal(t, frequency.', phase\_signal\_3,signal\_3, abs(signal\_3\_fft),3, 'Signal 3');

% Ploting Modulated Signals

plot\_signal(t, frequency.',phase\_modulation\_1,modulatedSignal\_t1, abs(modulated\_signal\_f1),4, 'Modulated Signal 1');

plot\_signal(t, frequency.', phase\_modulation\_2,modulatedSignal\_t2, abs(modulated\_signal\_f2),5, 'Modulated Signal 2');

plot\_signal(t, frequency.', phase\_modulation\_3,modulatedSignal\_t3, abs(modulated\_signal\_f3),6, 'Modulated Signal 3');

function plot\_signal(length\_t, f,angle, time, freq,counter, Time)

figure(counter)

subplot(3, 1, 1)

plot(length\_t, time)

xlabel('time')

%title of time Domain

title(strcat(Time, ' signal in time Domin'))

subplot(3, 1, 2)

plot(f, freq)

xlabel('f')

%title of frequency Domain

title(strcat(Time, ' Signal in frequency Domain'))

subplot(3, 1, 3)

plot(f, angle)

ylabel('phase')

xlabel('f')

%title of phase

title(strcat(Time, ' Phase'))

end

function lpf=demodulation(signal\_modulation, frequncy\_BandPass, out\_name, carrier\_signal, frequency\_signal\_sampling,figure,t)

demodulationdSignal = 2\* (signal\_modulation .\* carrier\_signal);

lpf = lowpass(demodulationdSignal, frequncy\_BandPass, frequency\_signal\_sampling);

fftSignal = fftshift(fft(lpf));

N=(length(fftSignal)/2);

frequency = (-1\*N:N-1);

phase=unwrap(angle(fftSignal));

plot\_signal(t, frequency.',phase,lpf, abs(fftSignal),figure, out\_name);

audiowrite(strcat("Output\_signals/", out\_name, '.wav'), lpf, frequency\_signal\_sampling);

end

function Array\_of\_carries = CarriersPhase(Carrier\_frequencyuency\_1, Carrier\_frequencyuency\_2, deg,time)

carry\_1 = cos((2\*pi \* Carrier\_frequencyuency\_1 \* time) + ((deg \* pi) / 180));

carry\_2 = cos((2\*pi \* Carrier\_frequencyuency\_2 \* time) + ((deg \* pi) / 180));

carry\_3 = sin((2\*pi \* Carrier\_frequencyuency\_2 \* time) + ((deg \* pi) / 180));

Array\_of\_carries=[carry\_1; carry\_2; carry\_3];

end

function Array\_of\_carries = CarriersDifferentFc(Carrier\_frequencyuency\_1, Carrier\_frequencyuency\_2, frequency\_dc,t)

carry\_1 = cos(2\*pi \* (Carrier\_frequencyuency\_1 + frequency\_dc) \* t);

carry\_2 = cos(2\*pi \* (Carrier\_frequencyuency\_2 + frequency\_dc) \* t);

carry\_3 = sin(2\*pi \* (Carrier\_frequencyuency\_2 + frequency\_dc) \* t);

Array\_of\_carries=[carry\_1; carry\_2; carry\_3];

end