Analogue Communication Theory

Final Lab Project • Ahmed Osama 6078

- Yehia Salah 6218 • Youssef Hassan 6259
- Noureldein Hazem 6261 Amr Ayman 6273 • Omar Mohamed 6290

- Youssef Sabry 6239
- **Analogue Communication Theory**
- **Final Lab Project DSB** Modulation Conclusions for DSB Modulation: SSB Modulation **NBFM Modulation**
- Conclusions for NBFM:

- **DSB Modulation**

- - Modulation classdef dsbmod
 - properties fs_original

 - resampled_msg carrier end methods function obj = dsbmod(msgsig,fc,fs_original,Xnyquist)
 - obj.fs_original = fs_original; new_fs = fc*Xnyquist; [p,q] = rat(new_fs/fs_original);
- - end
 - end
 - obj.resampled_msg = resample(msgsig,p,q); msg_len = length(obj.resampled_msg); obj.carrier = commonspectrum.gen_carrier(fc,0,msg_len/new_fs,msg_len); function sc = suppressed_carrier(obj,Ac) sc = (Ac .* obj.carrier) .* obj.resampled_msg; end
 - function tc = transmitted_carrier(obj,Mu,Ac) tc = Ac*(1 + Mu * obj.resampled_msg/max(obj.resampled_msg)) .* obj.carrier;
 - end Demodulation classdef dsbdemod methods (Static)
 - function demodulated=envelope(modulated,fcut,fs) temp= abs(hilbert(modulated)); filtered=lowpass(temp,fcut,fs,'ImpulseResponse','iir'); %HardCoded [p,q]=rat(48/500);demodulated=resample(filtered,p,q); end function demodulated=coherent(modulated,fc,fs,phase,fcut)
 - len=length(modulated); demod= modulated.*commonspectrum.gen_carrier(fc,phase,len/fs,len); filtered=lowpass(demod,fcut,fs,'ImpulseResponse','iir'); %HardCoded [p,q]=rat(48/500);demodulated=resample(filtered,p,q);
 - end
 - end
 - **Figures**
 - 2500 2000
 - 1500 1000 500 Frequency **DSB-SC**
 - 0.15 0.1 0.05
 - **Envelope of DSB SC**

 - **SNR0 Time** 0.05 0 -0.05 -0.1 **SNR10 Time**

0.1

-0.1

-0.15

0.1 0.08

0.06

0.04

0.02

-0.02

-0.04

-0.06

-0.08

0.08

0.06

0.04

0.02

-0.02

-0.04

-0.06

SNR30 Time

Frequency Drift in Time

-0.08 -0.1 <u></u>0 **Phase Shift in Time Conclusions for DSB Modulation:** Envelope Detector should only be used for transmitted carrier only Frequency Error is called Frequency Drift

SSB Modulation

methods (Static)

fs = fc * 5;

fs = fc * 5;

fs = fc * 5;

function ideal = idealsC(dsbsc,fc)

flow = (fc-4e3)/(fs/2); fhigh = fc/(fs/2); fcut = [flow, fhigh]; [b, a] = butter(4,fcut);

ideal = bandpass(dsbsc,[1e5-4e3

function practical = practicalSC(dsbsc,fc)

practical = filter(b, a, dsbsc);

tc = bandpass(dsbtc,[1e5-4e3

function tc = transmitted_carrier(dsbtc,fc)

1e5],fs,'ImpulseResponse','iir','Steepness',0.999,'StopbandAttenuation',90);

Modulation:

classdef ssbmod

end

end Demodulation: classdef ssbdemod methods (Static) function demodulated = envelope(modulated,fcut,fs) temp= abs(hilbert(modulated)); demod = temp - mean(temp); filtered=lowpass(demod,fcut,fs,'ImpulseResponse','iir','Steepness',0.999,'StopbandAttenuation',90); %HardCoded

demodulated=resample(filtered,p,q);

demodulated = resample(filtered,p,q);

len = length(modulated);

function demodulated = coherent(modulated,fc,fs,phase,fcut)

demod = modulated.*commonspectrum.gen_carrier(fc,phase,len/fs,len);

 $\times 10^5$

 $\times 10^5$

 $\times 10^4$

lowpass(demod,fcut,fs,'ImpulseResponse','iir','Steepness',0.999,'StopbandAttenuation',90);

end 6000 5000

%HardCoded

end end

4000

3000

0.5

250

100

0 └ -3

3000

2500

2000

1500

1000

500

450

400

350 300

250

200

150

100 50

0 ^L -3

300

200

150

100

50

-2

NBFM Modulation

Modulation and Demodulation:

methods(Static)

classdef NBFM

SNR30 Frequency

function modulated = modFM(audio,fc,fs,kf)

int_M = cumsum(dsbobj.resampled_msg);

modulated = cos(2*pi*fc*t' + kf.*int_M);

function demodulated = demodFM(moddedFM,fs)

% Discriminator to Convert to AM differentiated = diff(moddedFM);

% Integration of Msg

[p,q] = rat(48/500);

% Generating FM Modded Msg

-2

[p,q] = rat(48/500);

[p,q]=rat(48/500);

- 2000 1000 Frequency **SSB SC Spectrum**
- Frequency $\times 10^5$ **SSB TC Spectrum** 5000 4500 4000 3500 3000 2500 2000 1500 1000 500

SSB Butterworth Spectrum

SSB-SC Coherent Rx Spectrum

0

Frequency

Frequency

SNR0 Frequency

SSB-TC Envelope Rx Spectrum

- 200 150 100 50 -2 Frequency **SNR10 Frequency** 250
- % Envelope Detection envelop=abs(hilbert(differentiated)); % Remove DC demod = envelop - mean(envelop); % Remove High Freqs % Downsample Again demodulated=resample(demod,p,q); end

end

Figures

16

10 Frequency **Modulated FM Spectrum**

Conclusions for NBFM:

• For Narrow Band Modulation , the Modulation Index β should be less than 1

• Spectrum of NBFM takes the shape of DSB-TC

0.5

-0.6

250

200

150

100

50

0

300

200

150

100

160

140

120

100

60

40

20

-2

Frequency

Phase Shift in Spectrum

Magnitude 80 -2

- - Frequency

DSB-TC

Envelope of DSB TC SNR0 Frequency

×10⁴

 $\times 10^4$

50 -2 Frequency $\times 10^4$ **SNR30 Frequency** 140 120 100 Magnitude 80 60 40 20 -2 Frequency $\times 10^4$ **Frequency Drift in Spectrum**

Frequency

SNR10 Frequency

- 1e5],fs,'ImpulseResponse','iir','Steepness',0.999,'StopbandAttenuation',90);

0.1

-0.1

60

-0.05

0.1 0.08

0.06 0.04 0.02

-0.02

-0.04-0.06 -0.08

-0.1

0.1

0

-0.1

-0.2

-0.4 0

0

-0.05

-0.1

-0.2 L

 $\times 10^4$

40 20 -20 -40 -60 **SSB TC Time** 0.25 0.2 0.1

SSB Butterworth Time

SSB SC Time

SSB-SC Coherent Rx Time 0.05 **SSB-TC Envelope Rx Time** 0.5 0.4 0.3

SNR0 Time

SNR10 Time

- 0.08 0.06 0.04 -0.04 -0.06 -0.08 -0.1 **SNR30 Time** dsbobj = dsbmod(audio.filtered_data,1e5,audio.fs,5); t=linspace(0,length(int_M)/fs,length(int_M));
- demod = lowpass(demod,4000,fs,'ImpulseResponse','iir'); 4500 4000
 - $eta = rac{arDelta f}{f_m} << 1$

3500 3000

Demodulated FM Spectrum