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

- **DSB Modulation**
- Conclusions for 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);
- 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); end
- function sc = suppressed_carrier(obj,Ac) sc = (Ac .* obj.carrier) .* obj.resampled_msg; function tc = transmitted_carrier(obj,Mu,Ac) tc = Ac*(1 + Mu * obj.resampled_msg/max(obj.resampled_msg)) .* obj.carrier;
- end end end Demodulation
- classdef dsbdemod function demodulated=envelope(modulated,fcut,fs) temp= abs(hilbert(modulated)); dcShift=max(temp)-min(temp);
- demod=temp-dcShift/2; filtered=lowpass(demod,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);
- [p,q]=rat(48/500);demodulated=resample(filtered,p,q);

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

Magnitude

0.5

3.0

700

600

400

300

200

100

600

500

300

200

100

600

400

100

300

250

200

150

100

50

350

300

250

200

150

100

50

-2

Magnitude 300 -2

Frequency

DSB-TC

Envelope of DSB TC

Frequency

Frequency

Frequency

SNR30 Frequency

Frequency Drift in Spectrum

Frequency

Phase Shift in Spectrum

 $\times 10^4$

 3×10^4

 $\times 10^4$

SNR10 Frequency

SNR0 Frequency

filtered=lowpass(demod,fcut,fs,'ImpulseResponse','iir');

end **Figures**

end

5000

4000

3000

2000

1000

0.1 0.05 0 -0.05

-0.15

0

-0.1 -0.2

-0.4

0.3

0

-0.1

-0.05 -0.1

-0.2 ^{__}0

0.1

0.05

-0.1

-0.15

0.15

-0.05

-0.1

-0.2 L

classdef ssbmod

methods (Static)

end

end

classdef ssbdemod

methods (Static)

%HardCoded

%HardCoded

end end

6000

5000

4000

3000

2000

0.5

5000

4500

3500

3000

2500 2000

1000

500

400

300

100

800

700

600

500 400

300

200

100

500

400

300

200

100

600

400

300

100

-2

-2

-2

methods(Static)

end

end

end

0.5

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

end

end

Frequency

DSB-SC

Envelope of DSB SC

SNR0 Time

SNR10 Time

SNR30 Time

Frequency Drift in Time

Phase Shift in Time

function ideal = idealsC(dsbsc,fc)

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

function practical = practicalSC(dsbsc,fc)

practical = filter(b, a, dsbsc);

function tc = transmitted_carrier(dsbtc,fc)

function demodulated = envelope(modulated,fcut,fs)

filtered=lowpass(demod,fcut,fs,'ImpulseResponse','iir');

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

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

0.4 0.3

0.2

2.5

1.5

0.5 0 -0.5

0.25

0.2 0.15

0.1

0.05

-0.05 -0.1 -0.15

-0.2

-0.25

0.1

0.05

-0.05 -0.1

-0.2 L

0.15

0.05 0 -0.05 -0.1

0.4

0.2

0

-0.2

-0.4

-0.6

0.25 0.2

0.15 0.1

0.05

-0.1 -0.15

-0.2

0.1

-0.1

-0.2

0 -0.05

2

2

 $\times 10^4$

 $\times 10^4$

SSB SC Time

SSB TC Time

SSB Butterworth Time

SSB-SC Coherent Rx Time

SSB-TC Envelope Rx Time

SNR0 Time

SNR10 Time

SNR30 Time

filtered = lowpass(demod, fcut, fs, 'ImpulseResponse', 'iir');

temp= abs(hilbert(modulated)); dcShift=max(temp)-min(temp);

demodulated=resample(filtered,p,q);

demodulated = resample(filtered,p,q);

Frequency

SSB SC Spectrum

SSB TC Spectrum

SSB Butterworth Spectrum

Frequency

Frequency

Frequency

SNR0 Frequency

SNR10 Frequency

Frequency

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

int_M = cumsum(dsbobj.resampled_msg);

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

tunction demodulated = demodFM(moddedFM,fs)

envelop=abs(hilbert(differentiated));

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

demod = envelop - mean(envelop);

demodulated=resample(demod,p,q);

Frequency

t=linspace(0,length(int_M)/fs,length(int_M));

% Integration of Msg

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

% Envelope Detection

% Remove High Freqs

% Downsample Again

% Remove DC

% Generating FM Modded Msg

dsbobj = dsbmod(audio.filtered_data,1e5,audio.fs,5);

demod = lowpass(demod,4000,fs,'ImpulseResponse','iir');

3000

2500

1500

1000

500

 $eta = rac{arDelta f}{f_m} << 1$

Demodulated FM Spectrum

SNR30 Frequency

SSB-TC Envelope Rx Spectrum

SSB-SC Coherent Rx Spectrum

len = length(modulated);

demod=temp-dcShift/2;

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

ideal = lowpass(dsbsc,fc,fs,'ImpulseResponse','iir');

tc = lowpass(dsbtc,fc,fs,'ImpulseResponse','iir');

• Envelope Detector should only be used for transmitted carrier only

Conclusions for DSB Modulation:

• Frequency Error is called Frequency Drift

fs = fc * 5;

fs = fc * 5;

fs = fc * 5;

- **SSB Modulation** Modulation:
- Demodulation:
- end

NBFM Modulation Modulation and Demodulation: classdef NBFM

- **Modulated FM Spectrum Conclusions for NBFM:** • Spectrum of NBFM takes the shape of DSB-TC • For Narrow Band Modulation , the Modulation Index β should be less than 1

