ECE300 – Pset 2

Jonathan Lam

September 15, 2020

Figures are located at the end of the document.

Load a short audio clip (either one you record, one from online or one built in to MATLAB) into MATLAB and call this signal $m(t)$. You will explore noise-free AM strategies using $m(t)$ as your message signal. Scale the message so that the maximum, in absolute value, is 1.

```
clear; close all; clc;
set(0, 'defaultTextInterpreter', 'latex');

% Most variable names should be fairly clear, but the following
% abbreviations are made for commonly-used meanings:
% - sig:    signal (in time domain)
% - ft:     fourier transform (in freq. domain)
% - smp:    relating to the raw audio sample data
% - am:     relating to the AM-modulated wave
% - us/ds:  upsampled/downsampled
% - cv:     conventional AM
% - rcv:    rectified conventional AM
% - lrcv:   low-pass filtered rectified conventional AM

pset_name = 'ece300_pset2';         % for exporting figures as PDFs

% audio file parameters
file_smp = 'preamble.wav';          % filename of audio input file
                                    % audio source: [2]
file_out = 'preambleAM.wav';        % filename of audio output file
dur_smp = 5;                        % truncate audio file to this duration
                                    % in seconds to speed up program
W = 5000;                           % approximate bandwidth of audio;
                                    % 5kHz is typical in AM radio [1]

% AM-modulation parameters
fc_am = 600000;                     % AM carrier frequency; tend to range
                                    % from 550 to 1720kHz [1]
fs_am = 2000000;                    % sampling frequency (has to be >=2*Fc)

% loads audio signal into smp, and sets the variable Fs_smp to be the
% audio file's sampling rate
[sig_smp, fs_smp] = audioread(file_smp);
sig_smp = sig_smp(1:round(fs_smp * dur_smp));   % truncate audio file
N = length(sig_smp);                            % number of samples

% message signal should have |m(t)| <= 1
max_amplitude_smp = max(sig_smp);
```

```
sig_m = sig_smp / max_amplitude_smp;

% samplesT are the time values at which y were sampled
t_smp = linspace(0, dur_smp, N);
```

1. *Plot $m(t)$ in time and its Fourier transform $M(\omega)$ (both amplitude and phase). What is the bandwidth of your message?*

```
figure('visible', 'off');
subplot(3, 1, 1);
plot(t_smp, sig_m);
ylabel('$m(t)$');
xlabel('$t$ (s)');
title('Baseband $m(t)$ (not upsampled)');

wd = linspace(-pi, pi, N);           % these three lines for generating
f_smp = wd * fs_smp / (2 * pi);      % the FFT plots
ft_m = fftshift(fft(sig_m)) / fs_smp;

subplot(3, 1, 2);
plot(f_smp, abs(ft_m));
ylabel('$|\mathcal{F}\{m\}(f)|$');
xlabel('$f$ (Hz)');
title('Magnitude of $\mathcal{F}\{m\}(f)$');
subplot(3, 1, 3);
plot(f_smp, unwrap(angle(ft_m)));
ylabel('$\angle\{\mathcal{F}\{m\}(f)\}$');
xlabel('$f$ (Hz)');
title('(Unwrapped) Phase of $\mathcal{F}\{m\}(f)$');
```

Most of the information seems to have a frequency less than $\approx$ 7kHz, so this is a reasonable value for the bandwidth (the valuable is also reasonable given that the input is a calm speaking voice). It seems that there's some information in the 7kHz-10kHz range, but its amplitude doesn't seem very significant. Past 10kHz the signal seems to have trivial frequency components.

(Despite this, I've approximated limited the bandwidth W to be 5kHz, since according to [1] this is a common value for the upper modulating frequency in AM. This is also the cutoff frequency used in the LPF for the conventional AM.)

2. *Generate a carrier signal over the same timespan to be used for DSB-SC AM and DSB AM. There is no one signal that will work here, but try to make it realistic!*

```
% generate carrier frequency at Fc, and sample it at Fs_am (which should
% be >= 2*Fc)
t_am = linspace(0, dur_smp, dur_smp * fs_am);
N_am = length(t_am);
sig_carrier = cos(2 * pi * fc_am * t_am);

% upsample sample to Fs_am (same sampling frequency as for the carrier)
sig_m_us = interp1(t_smp, sig_m, t_am);
```

3. *Generate the DSB-SC and DSB modulated signals with message $m(t)$ and plot them in the time and frequency domains. In one sentence, describe the difference.*

```
sig_dsbsc = sig_carrier .* sig_m_us;

% plot DSB-SC signal in time domain
figure('visible', 'off');
subplot(3, 2, 1);
plot(t_am, sig_dsbsc);
ylabel('$dsbsc\{m\}(t)$');
xlabel('$t$ (s)');
title('$dsbsc\{m\}(t)$');

% plot DSB-SC signal in frequency domain
wd = linspace(-pi, pi, N_am);
f_am = wd * fs_am / (2 * pi);
ft_dsbsc = fftshift(fft(sig_dsbsc)) / fs_am;
subplot(3, 2, 3);
plot(f_am, abs(ft_dsbsc));
ylabel('$|\mathcal{F}\{dsbsc\{\{m\}\}\}(f)|$');
xlabel('$f$ (Hz)');
title('Magnitude of $\mathcal{F}\{dsbsc\{m\}\}(f)$');
xlim([-fc_am * 1.25, fc_am * 1.25]);
subplot(3, 2, 5);
plot(f_am, unwrap(angle(ft_dsbsc)));
ylabel('$\angle\{\mathcal{F}\{dsbsc\{m\}\}(f)\}$');
xlabel('$f$ (Hz)');
title('(Unwrapped) Phase of $\mathcal{F}\{dsbsc\{m\}\}(f)$');
xlim([-fc_am * 1.25, fc_am * 1.25]);

% generate DSB signal, plot in time domain
% a small pilot amplitude Ap was used here so that the original frequency
% spectrum would still be visible in the fourier transform; if Ap = 1, for
% example, then it would look like the conventional AM case (just two
% deltas, which is less informative)
Ap_dsb = 0.05;
sig_dsb = sig_dsbsc + Ap_dsb * sig_carrier;
ft_dsb = fftshift(fft(sig_dsb)) / fs_am;
subplot(3, 2, 2);
plot(t_am, sig_dsb);
ylabel('$dsb\{m\}(t)$');
xlabel('$t$ (s)');
title('$dsb\{m\}(t)$');

% plot DSB signal in frequency domain
subplot(3, 2, 4);
plot(f_am, abs(ft_dsb));
ylabel('$|\mathcal{F}\{dsb\{m\}\}(f)|$');
xlabel('$f$ (Hz)');
title('Magnitude of $\mathcal{F}\{dsb\{m\}\}(f)$');
xlim([-fc_am * 1.25, fc_am * 1.25]);
subplot(3, 2, 6);
plot(f_am, unwrap(angle(ft_dsb)));
ylabel('$\angle\{\mathcal{F}\{dsb\{m\}\}(f)\}$');
xlabel('$f$ (Hz)');
title('(Unwrapped) Phase of $\mathcal{F}\{dsb\{m\}\}(f)$');
xlim([-fc_am * 1.25, fc_am * 1.25]);
```

There is a much higher peak in the Fourier transform of the DSB signal than in that of the DSB-SC signal (which makes the former less power-efficient but phase-coherent.)

4. *Using the same carrier frequency, generate lower SSB and upper SSB AM signals, and plot them in time and frequency.*

```
% from previous problem set: take hilbert transform of a signal
function res = hilbertTransform(sig_x)
    ft_x = fft(sig_x);
    len_x = length(sig_x);
    % get signum of frequency; +1 for 0 to pi, -1 for -pi to 0
    sgn = [ones(1, floor(len_x/2)), -1*ones(1, ceil(len_x/2))];
    res = ifft(-1j * sgn .* ft_x);
end

% generate lssb, ussb AM-modulated signals
sig_mhat = hilbertTransform(sig_m_us);
sig_quad = sig_mhat .* sin(2 * pi * fc_am * t_am);  % quadrature component
sig_lssb = (sig_dsbsc + sig_quad) / 2;
sig_ussb = (sig_dsbsc - sig_quad) / 2;

% plot lssb in time domain
figure('visible', 'off');
subplot(3, 2, 1);
plot(t_am, real(sig_lssb));
ylabel('$lssb\{m\}(t)$');
xlabel('$t$ (s)');
title('$lssb\{m\}(t)$');

% plot lssb in frequency domain
ft_lssb = fftshift(fft(sig_lssb)) / fs_am;
subplot(3, 2, 3);
plot(f_am, abs(ft_lssb));
ylabel('$|\mathcal{F}\{lssb\{m\}\}(f)|$');
xlabel('$f$ (Hz)');
title('Magnitude of $\mathcal{F}\{lssb\{m\}\}(f)$');
xlim([-fc_am * 1.25, fc_am * 1.25]);
subplot(3, 2, 5);
plot(f_am, unwrap(angle(ft_lssb)));
ylabel('$\angle\{\mathcal{F}\{lssb\{m\}\}(f)\}$');
xlabel('$f$ (Hz)');
title('(Unwrapped) Phase of $\mathcal{F}\{lssb\{m\}\}(f)$');
xlim([-fc_am * 1.25, fc_am * 1.25]);

% plot ussb in time domain
subplot(3, 2, 2);
plot(t_am, real(sig_ussb));
ylabel('$ussb\{m\}(t)$');
xlabel('$t$ (s)');
title('$ussb\{m\}(t)$');

% plot ussb in frequency domain
ft_ussb = fftshift(fft(sig_ussb)) / fs_am;
subplot(3, 2, 4);
plot(f_am, abs(ft_ussb));
ylabel('$|\mathcal{F}\{ussb\{m(t)\}\}(f)|$');
xlabel('$f$ (Hz)');
title('Magnitude of $\mathcal{F}\{ussb\{m\}\}(f)$');
```

```
xlim([-fc_am * 1.25, fc_am * 1.25]);
subplot(3, 2, 6);
plot(f_am, unwrap(angle(ft_ussb)));
ylabel('$\angle\{\mathcal{F}\{ussb\{m\}\}(f)\}$');
xlabel('$f$ (Hz)');
title('(Unwrapped) Phase of $\mathcal{F}\{ussb\{m\}\}(f)$');
xlim([-fc_am * 1.25, fc_am * 1.25]);
```

5. *Using the same carrier frequency, create a conventional AM signal (ensure you choose the amplitude of the carrier correctly). Plot this signal in the time domain and frequency domain as well.*

```
% this is pretty much the same as DSB AM, but with a different pilot
% amplitude
sig_cv = (1 + sig_m_us) .* cos(2 * pi * fc_am * t_am);

% plot conventional AM signal in time domain
figure('visible', 'off');
subplot(3, 3, 1);
plot(t_am, sig_cv);
ylabel('$cv\{m\}(t)$');
xlabel('$t$ (s)');
title('Conventional AM Modulation');

% plot conventional AM signal in frequency domain
ft_cv = fftshift(fft(sig_cv)) / fs_am;
subplot(3, 3, 4);
plot(f_am, abs(ft_cv));
ylabel('$|\mathcal{F}\{cv\{m\}\}(f)|$');
xlabel('$f$ (Hz)');
title('Magnitude of $\mathcal{F}\{cv\{m\}\}(f)$');
xlim([-fc_am * 1.25, fc_am * 1.25]);
subplot(3, 3, 7);
plot(f_am, unwrap(angle(ft_cv)));
ylabel('$\angle\{\mathcal{F}\{cv\{m\}\}(f)\}$');
xlabel('$f$ (Hz)');
title('(Unwrapped) Phase of $\mathcal{F}\{cv\{m\}\}(f)$');
xlim([-fc_am * 1.25, fc_am * 1.25]);
```

6. *For the conventional AM signal, do demodulate we rectify and then apply a low-pass filter. In MATLAB, you should be able to rectify the signal in one line. Plot the rectified signal in time and frequency. Then, design a first-order low-pass filter (you decide the cutoff frequency) and apply it to the signal. You can do this using MATLAB filter functions, or through convolution in time/multiplication in frequency directly. Plot the resultant output signal in time and frequency and play the signal as audio.*

```
% rectify signal, subtract 1; estimate rectification with abs()
sig_rcv = abs(sig_cv);

% low pass in frequency domain:
% first-order LPF has transfer function H(f) = (1 + j*(f/fc))^{-1}
% choose cutoff frequency fc = W (original bandwidth)
% also, there seems to be some low-frequency periodic error when rectifying
% using abs() (i.e., the sample x-intercepts are not exactly correct), so
% I added an additional HPF here with fc = 20Hz
fc_lpf = W;                                % LPF cutoff freq. at W
fc_hpf = 20;                               % HPF cutoff freq. at 20Hz
lpf = (1 + f_am/fc_lpf*1j).^-1;            % LPF
hpf = (1 - fc_hpf*f_am.^-1*1j).^-1;        % HPF
ft_rcv = fftshift(fft(sig_rcv)) / fs_am;   % rect. sig. in freq. domain
ft_lrcv = hpf .* lpf .* ft_rcv;            % apply filters in freq. domain
sig_lrcv = ifft(ifftshift(ft_lrcv) * fs_am);

% plot rectified/shifted conventional signal in time domain
subplot(3, 3, 2);
plot(t_am, sig_rcv);
ylabel('$|cv\{m\}|(t)-1$');
xlabel('$t$ (s)');
title('Rectified, shifted conventional AM');

% plot rectified/shifted conventional signal in frequency domain
subplot(3, 3, 5);
plot(f_am, abs(ft_rcv));
ylabel('$|\mathcal{F}\{|cv\{m\}|-1\}(f)|$');
xlabel('$f$ (Hz)');
title('Magnitude of $\mathcal{F}\{|cv\{m\}|-1\}(f)$');
xlim([-2*fc_am * 1.25, 2*fc_am * 1.25]);
subplot(3, 3, 8);
plot(f_am, unwrap(angle(ft_rcv)));
ylabel('$\angle\{\mathcal{F}\{|cv\{m\}\}-1|(f)\}$');
xlabel('$f$ (Hz)');
title('(Unwrapped) Phase of $\mathcal{F}\{|cv\{m\}\}-1|(f)$');
xlim([-fc_am * 1.25, fc_am * 1.25]);

% plot rectified, shifted, filtered conventional AM signal in time domain
subplot(3, 3, 3);
plot(t_am, real(sig_lrcv));
ylabel('$lpf\{hpf\{|cv\{m\}|-1\}\}(t)$');
xlabel('$t$ (s)');
title('Rectified, shifted, filtered conventional AM');

% plot rectified, shifted, filtered conventional AM signal in freq. domain
subplot(3, 3, 6);
plot(f_am, abs(ft_lrcv));
ylabel('$|\mathcal{F}\{lpf\{hpf\{|cv\{m\}|-1\}\}\}(f)|$');
xlabel('$f$ (Hz)');
title('Magnitude of $\mathcal{F}\{lpf\{hpf\{|cv\{m\}|-1\}\}\}(f)$');
xlim([-W * 1.25, W * 1.25]);
subplot(3, 3, 9);
plot(f_am, unwrap(angle(ft_lrcv)));
ylabel('$\angle\{\mathcal{F}\{lpf\{hpf\{|cv\{m\}|-1\}\}\}(f)\}$');
xlabel('$f$ (Hz)');
title('(Unwrapped) Phase of $\mathcal{F}\{lpf\{hpf\{|cv\{m\}|-1\}\}\}(f)$');
xlim([-W * 1.25, W * 1.25]);

% reconstruct original audio signal, should approximately be sig_m:
% 1. downsample to original sampling frequency fs_smp (at times t_smp)
%    (MATLAB cannot use such a high sample rate for audio anyways [3])
% 2. take the real part (filter produces complex wave)
% 3. re-scale to initial volume
sig_lrcv_ds = real(interp1(t_am, sig_lrcv, t_smp)) .* max_amplitude_smp;
```

```
% play new audio signal and save to file_out
sound(sig_lrcv_ds, fs_smp);
audiowrite(file_out, sig_lrcv_ds, fs_smp);
```

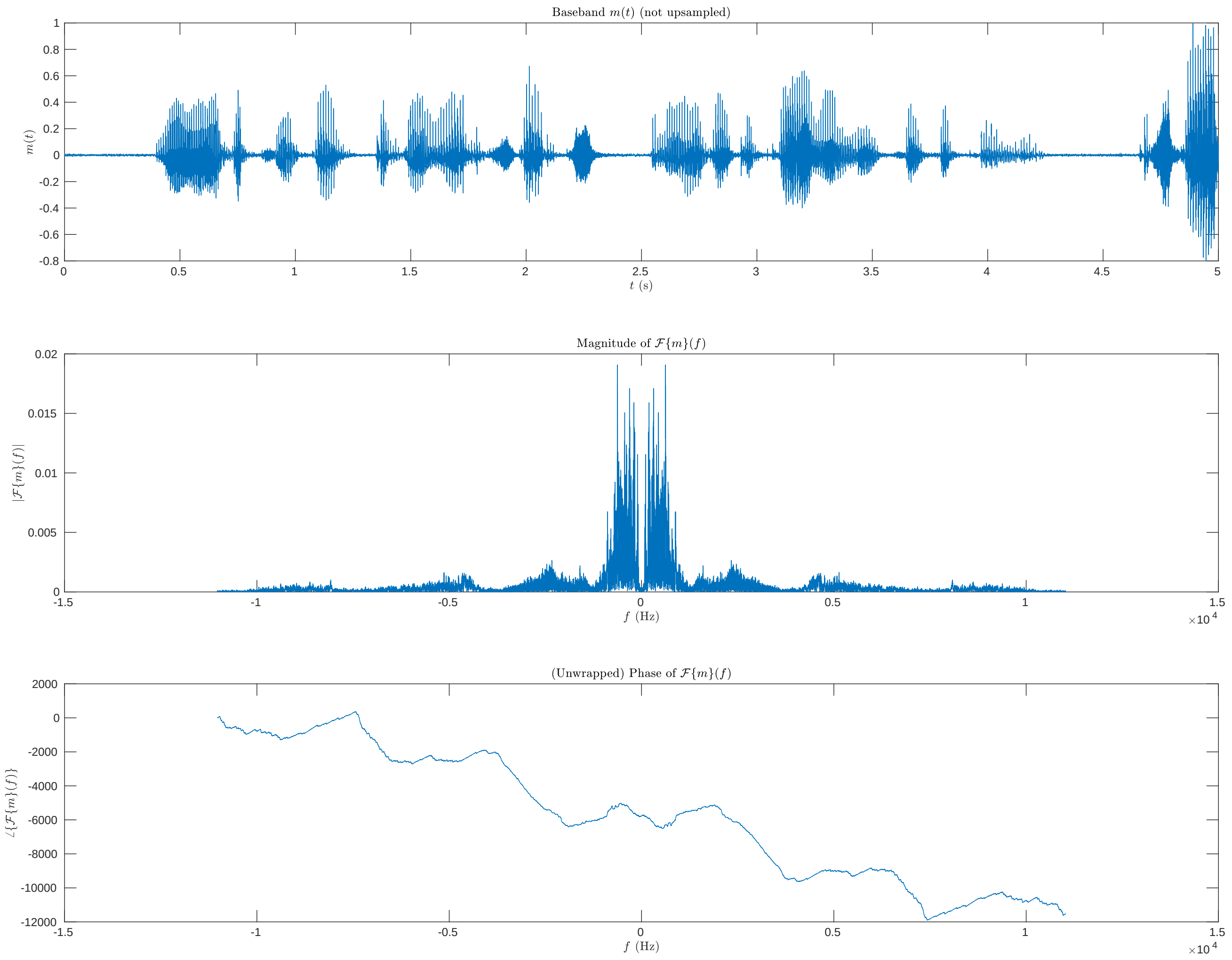
It is clear from the figure ("Rectified, shifted conventional AM") that there is some distortion caused by rectification. If the rectification were perfect, the bottom line would be flat. However, due to limitations in machine accuracy and the periodic nature of the signal and sampling, there is a resulting low-frequency distortion that I had to filter out with a HPF. I chose 20Hz as the cutoff frequency for this filter, since that is approximately the lowest frequency we can hear, and it removes most but not all of the distortion; there is still a spike in the Fourier transform of the signal near $f = 0$ that was not present in the original signal. The resultant audio quality is not terrible.

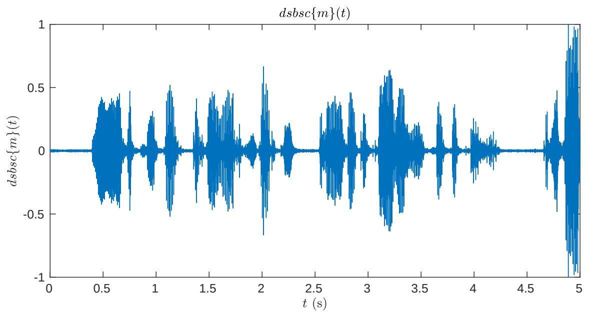
Cited sources

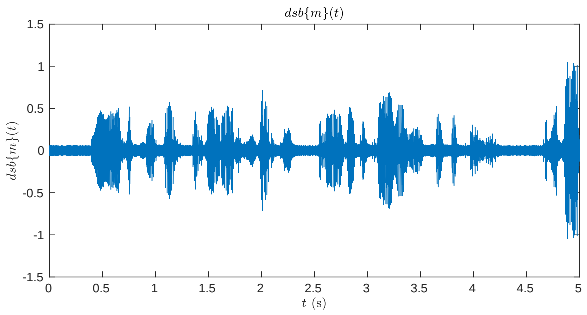
1. https://fas.org/man/dod-101/navy/docs/es310/AM.htm

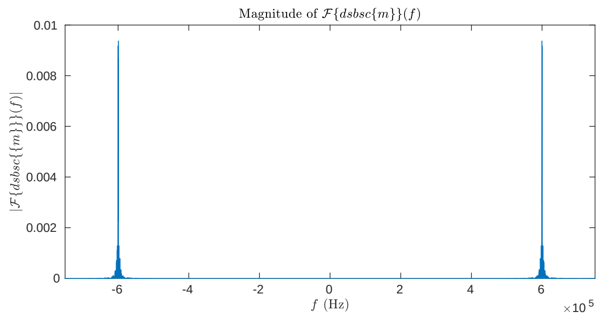
2. https://www2.cs.uic.edu/~i101/SoundFiles/

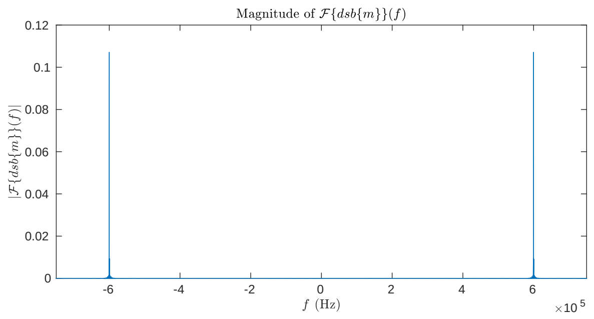
3. https://www.mathworks.com/matlabcentral/answers/195697-what-is-the-maximum-sample-rate-supported-by-sound)

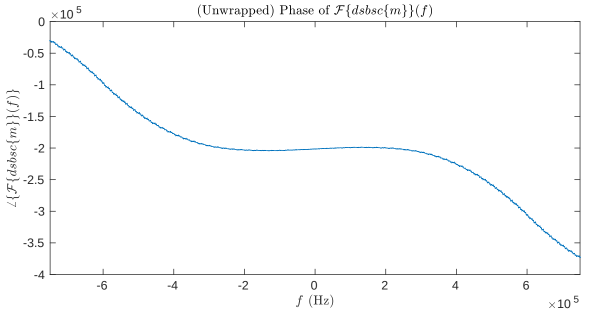

Baseband $m(t)$ (not upsampled)
$m(t)$
t (s)
Magnitude of $\mathcal{F}\{m\}(f)$
$|\mathcal{F}\{m\}(f)|$
f (Hz)
$\times 10^4$
(Unwrapped) Phase of $\mathcal{F}\{m\}(f)$
$\angle\{\mathcal{F}\{m\}(f)\}$
f (Hz)
$\times 10^4$

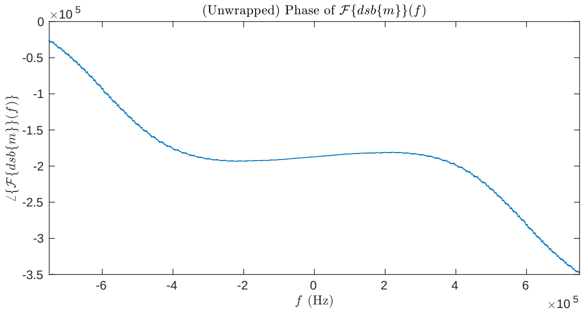

$dsbsc\{m\}(t)$
$dsbsc\{m\}(t)$
t (s)
1
0.5
0
-0.5
-1
0 0.5 1 1.5 2 2.5 3 3.5 4 4.5 5

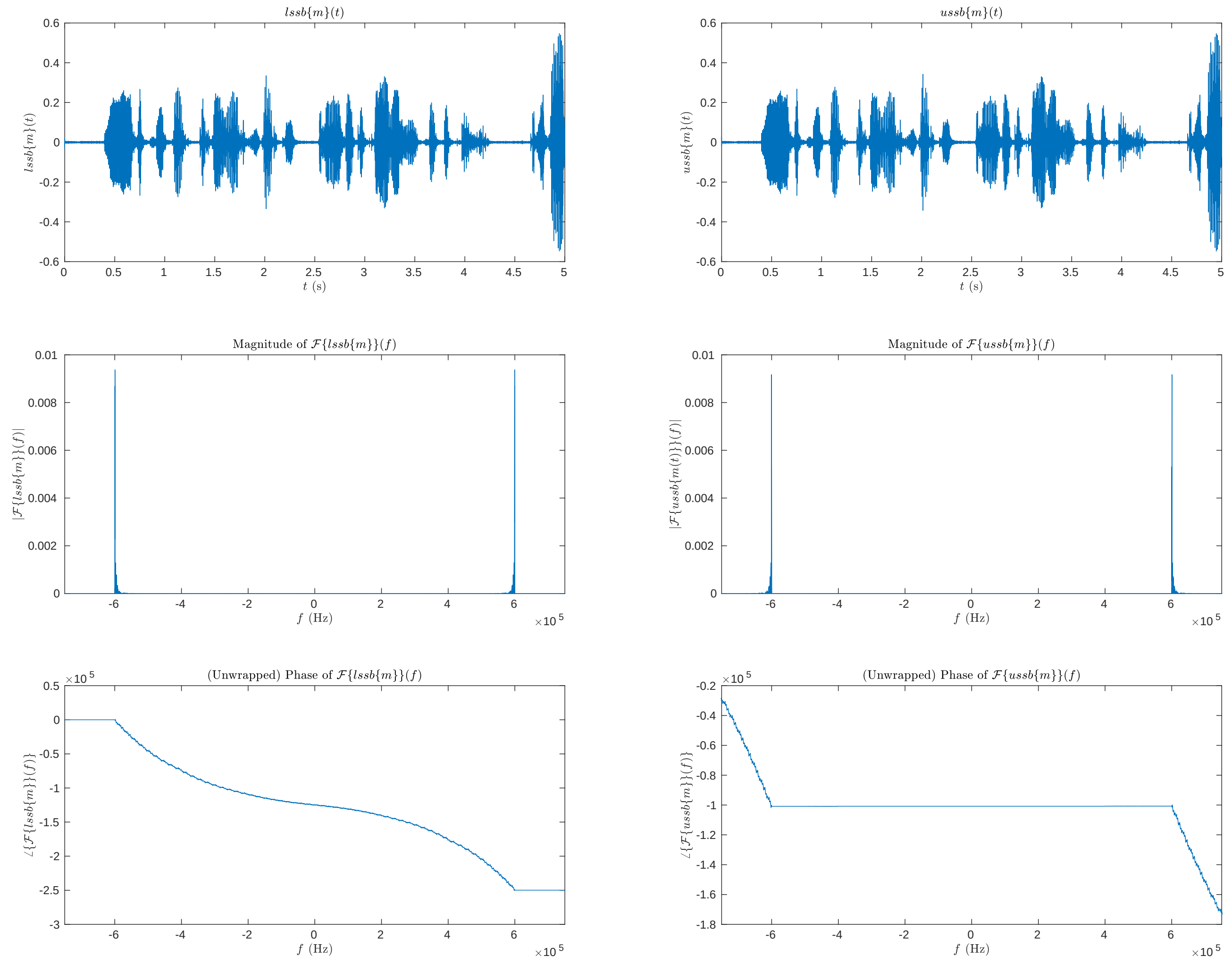

$dsb\{m\}(t)$
$dsb\{m\}(t)$
t (s)
1.5
1
0.5
0
-0.5
-1
-1.5
0 0.5 1 1.5 2 2.5 3 3.5 4 4.5 5

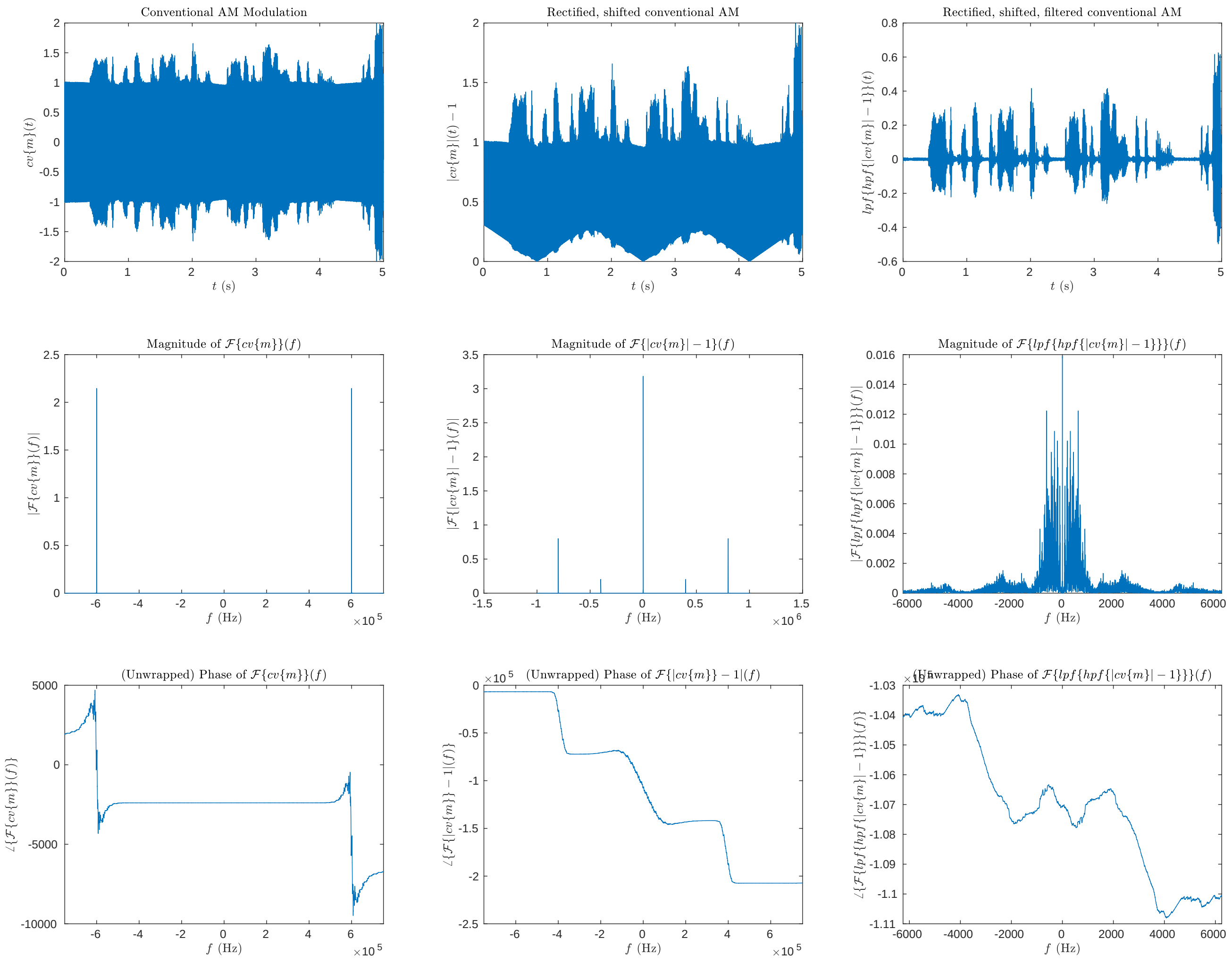

Magnitude of $\mathcal{F}\{dsbsc\{m\}\}(f)$
$|\mathcal{F}\{dsbsc\{m\}\}(f)|$
f (Hz)
$\times 10^5$
0.01
0.008
0.006
0.004
0.002
0
-6 -4 -2 0 2 4 6


Magnitude of $\mathcal{F}\{dsb\{m\}\}(f)$
$|\mathcal{F}\{dsb\{m\}\}(f)|$
f (Hz)
$\times 10^5$
0.12
0.1
0.08
0.06
0.04
0.02
0
-6 -4 -2 0 2 4 6


(Unwrapped) Phase of $\mathcal{F}\{dsbsc\{m\}\}(f)$
$\angle\{\mathcal{F}\{dsbsc\{m\}\}(f)\}$
$\times 10^5$
0
-0.5
-1
-1.5
-2
-2.5
-3
-3.5
-4
f (Hz)
$\times 10^5$
-6 -4 -2 0 2 4 6


(Unwrapped) Phase of $\mathcal{F}\{dsb\{m\}\}(f)$
$\angle\{\mathcal{F}\{dsb\{m\}\}(f)\}$
$\times 10^5$
0
-0.5
-1
-1.5
-2
-2.5
-3
-3.5
f (Hz)
$\times 10^5$
-6 -4 -2 0 2 4 6


$lssb\{m\}(t)$
$lssb\{m\}(t)$
t (s)
$ussb\{m\}(t)$
$ussb\{m\}(t)$
t (s)
Magnitude of $\mathcal{F}\{lssb\{m\}\}(f)$
$|\mathcal{F}\{lssb\{m\}\}(f)|$
f (Hz)
$\times 10^5$
Magnitude of $\mathcal{F}\{ussb\{m\}\}(f)$
$|\mathcal{F}\{ussb\{m(t)\}\}(f)|$
f (Hz)
$\times 10^5$
(Unwrapped) Phase of $\mathcal{F}\{lssb\{m\}\}(f)$
$\angle\{\mathcal{F}\{lssb\{m\}\}(f)\}$
f (Hz)
$\times 10^5$
(Unwrapped) Phase of $\mathcal{F}\{ussb\{m\}\}(f)$
$\angle\{\mathcal{F}\{ussb\{m\}\}(f)\}$
f (Hz)
$\times 10^5$


Conventional AM Modulation
$cv\{m\}(t)$
t (s)
Rectified, shifted conventional AM
$|cv\{m\}|(t) - 1$
t (s)
Rectified, shifted, filtered conventional AM
$lpf\{hpf\{|cv\{m\}| - 1\}\}(t)$
t (s)
Magnitude of $\mathcal{F}\{cv\{m\}\}(f)$
$|\mathcal{F}\{cv\{m\}\}(f)|$
f (Hz)
$\times 10^5$
Magnitude of $\mathcal{F}\{|cv\{m\}| - 1\}(f)$
$|\mathcal{F}\{|cv\{m\}| - 1\}(f)|$
f (Hz)
$\times 10^6$
Magnitude of $\mathcal{F}\{lpf\{hpf\{|cv\{m\}| - 1\}\}\}(f)$
$|\mathcal{F}\{lpf\{hpf\{|cv\{m\}| - 1\}\}\}(f)|$
f (Hz)
(Unwrapped) Phase of $\mathcal{F}\{cv\{m\}\}(f)$
$\angle\{\mathcal{F}\{cv\{m\}\}(f)\}$
f (Hz)
$\times 10^5$
(Unwrapped) Phase of $\mathcal{F}\{|cv\{m\}\} - 1|(f)$
$\angle\{\mathcal{F}\{|cv\{m\}\} - 1|(f)\}$
f (Hz)
$\times 10^5$
$\times 10^5$
(Unwrapped) Phase of $\mathcal{F}\{lpf\{hpf\{|cv\{m\}| - 1\}\}\}(f)$
$\angle\{\mathcal{F}\{lpf\{hpf\{|cv\{m\}| - 1\}\}\}(f)\}$
f (Hz)
$\times 10^5$