EECS 3451 Lab 5

Table of Contents

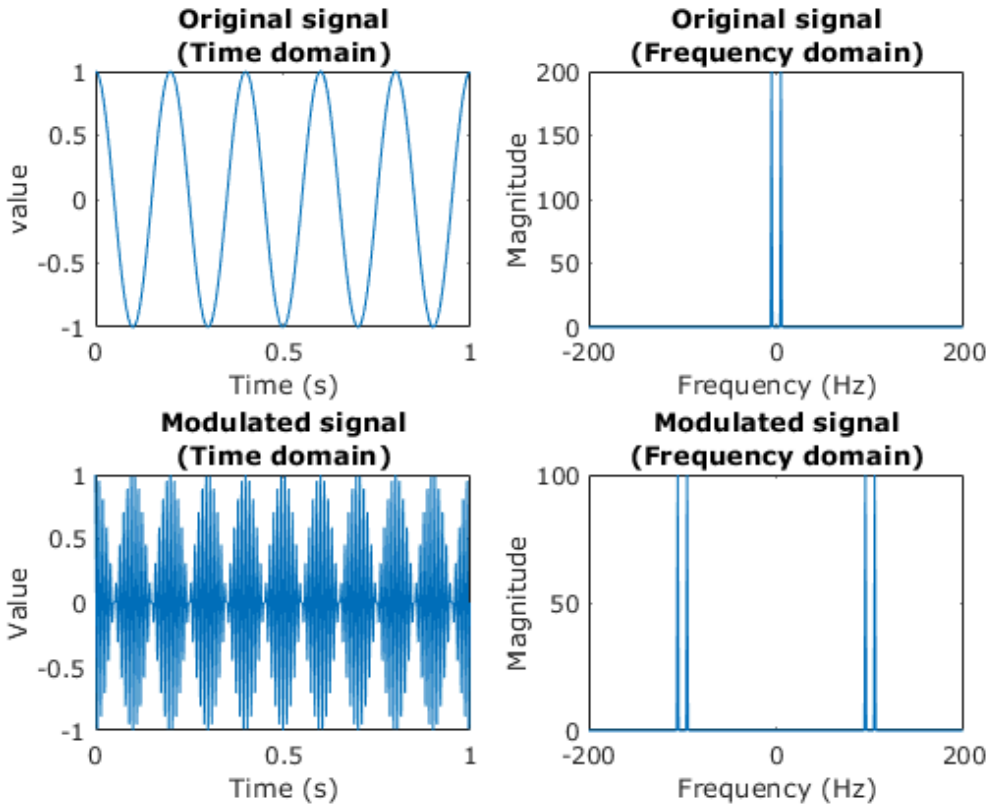
AUTHORS

- Jonathan Baldwin (212095691)
- Mark Savin (212921128)
- Sarwat Shaheen (214677322)

Q1

```
Fs = 400;
Ts = 1/Fs;
Fc = 100;
t = 0:Ts:1-Ts;
x = cos(10*pi.*t);

[xm,tm] = modulate(x,Fc,Fs); % modulate x with carrier frequency Fc
[F, X] = do_fft(x, Fs); % Take the FFT of the original signal
[Fm, Xm] = do_fft(xm, Fc); % Take the FFT of the modulated signal

playsound(xm,Fs); % Play the modulated signal

subplot(2,2,1);
plot(t, x);
title(["Original signal", "(Time domain)"]);
xlabel("Time (s)");
ylabel("value");

subplot(2,2,2);
plot(F, abs(X));
title(["Original signal", "(Frequency domain)"]);
xlabel("Frequency (Hz)");
ylabel("Magnitude");

subplot(2,2,3);
plot(tm,xm);
title(["Modulated signal", "(Time domain)"]);
xlabel("Time (s)");
ylabel("Value");
```

```
subplot(2,2,4);
plot(F, abs(Xm));
title(["Modulated signal", "(Frequency domain)"]);
xlabel("Frequency (Hz)");
ylabel("Magnitude");

save q1.mat t tm x xm Fs Fc
```

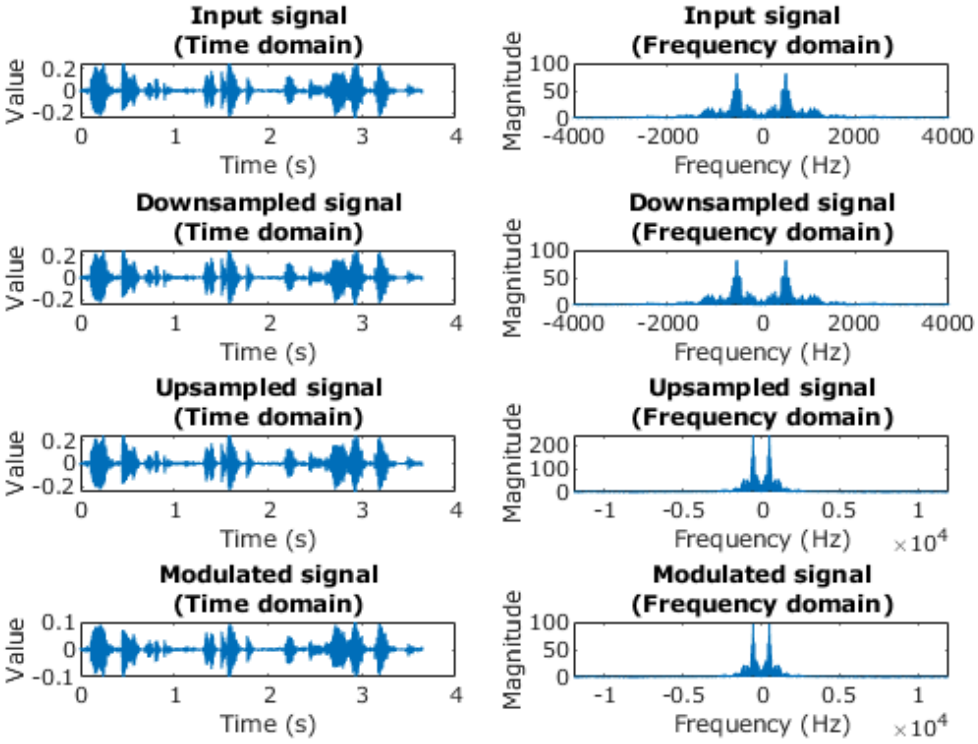


Q2

```
% Read original signal
[x, Fs] = audioread('P_12_1.wav');
x = x';
t = (0:length(x)-1)/Fs; % time domain of original signal

% Downsample to 8kHz
Fc = 8000;
Fs2 = Fc; % Effective bandwidth = 4000 Hz

td = 0:1/Fs2:max(t);
xd = interp1(t,x,td,'linear');

% Upsample to 24kHz
Fs3 = 2*Fc+Fs2;
```

```
Fsu = Fs3;
tu = 0:1/Fsu:max(td);
xu = interp1(td,xd,tu,'linear');

% Modulate
xm = modulate(xu, tu, Fc);

% Plot
subplot(4,2,1);
plot(t,x);
title(["Input signal","(Time domain)"]);
xlabel("Time (s)");
ylabel("Value");

fp1 = subplot(4,2,2);
[F, X] = do_fft(x, Fs);
plot(F, abs(X));
title(["Input signal","(Frequency domain)"]);
xlabel("Frequency (Hz)");
ylabel("Magnitude");

subplot(4,2,3);
plot(td, xd);
title(["Downsampled signal","(Time domain)"]);
xlabel("Time (s)");
ylabel("Value");

fp2 = subplot(4,2,4);
[Fd, Xd] = do_fft(xd, Fs2);
plot(Fd, abs(Xd));
title(["Downsampled signal","(Frequency domain)"]);
xlabel("Frequency (Hz)");
ylabel("Magnitude");

% generate FFT and plot
subplot(4,2,5);
plot(tu, xu);
title(["Upsampled signal","(Time domain)"]);
xlabel("Time (s)");
ylabel("Value");

fp3 = subplot(4,2,6);
[Fu, Xu] = do_fft(xu, Fsu);
plot(Fu, abs(Xu));
title(["Upsampled signal","(Frequency domain)"]);
xlabel("Frequency (Hz)");
ylabel("Magnitude");

% generate FFT and plot
subplot(4,2,7);
plot(tu, xm);
title(["Modulated signal","(Time domain)"]);
xlabel("Time (s)");
ylabel("Value");
```

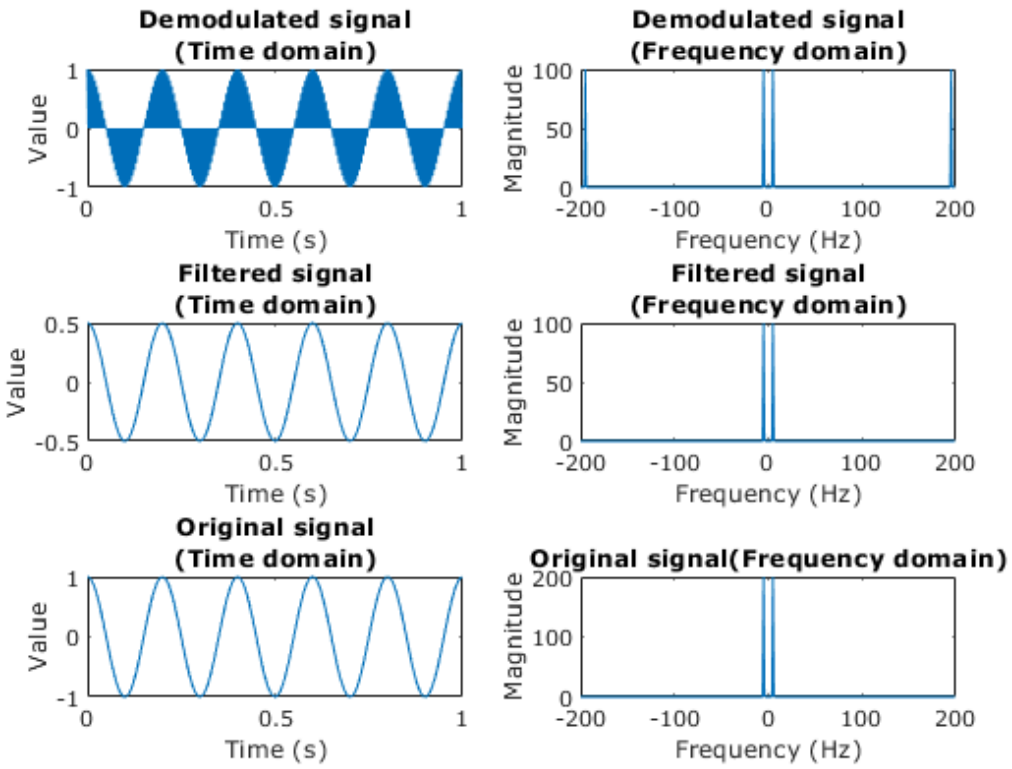
```
fp4 = subplot(4,2,8);
[Fm, Xm] = do_fft(xm, Fsu);
plot(Fm, abs(Xm));
title(["Modulated signal","(Frequency domain)"]);
xlabel("Frequency (Hz)");
ylabel("Magnitude");
```



Q3

```
load q1.mat t x xm Fs Fc

x2 = xm.*cos(2*pi*Fc*t);              % demodulate the signal
x2f = low_pass_filter(x2,Fs,Fc/2);  % low pass with cutoff Fc/2

[F2, X2] = do_fft(x2, Fs);            % Take FFT of demodulated
[F2f, X2f] = do_fft(x2f, Fs);         % Take FFT of filtered
[F, X] = do_fft(x, Fs);               % Take FFT of original

subplot(3,2,1);
plot(t,x2);
title(["Demodulated signal", "(Time domain)"]);
xlabel('Time (s)');
ylabel('Value');
```

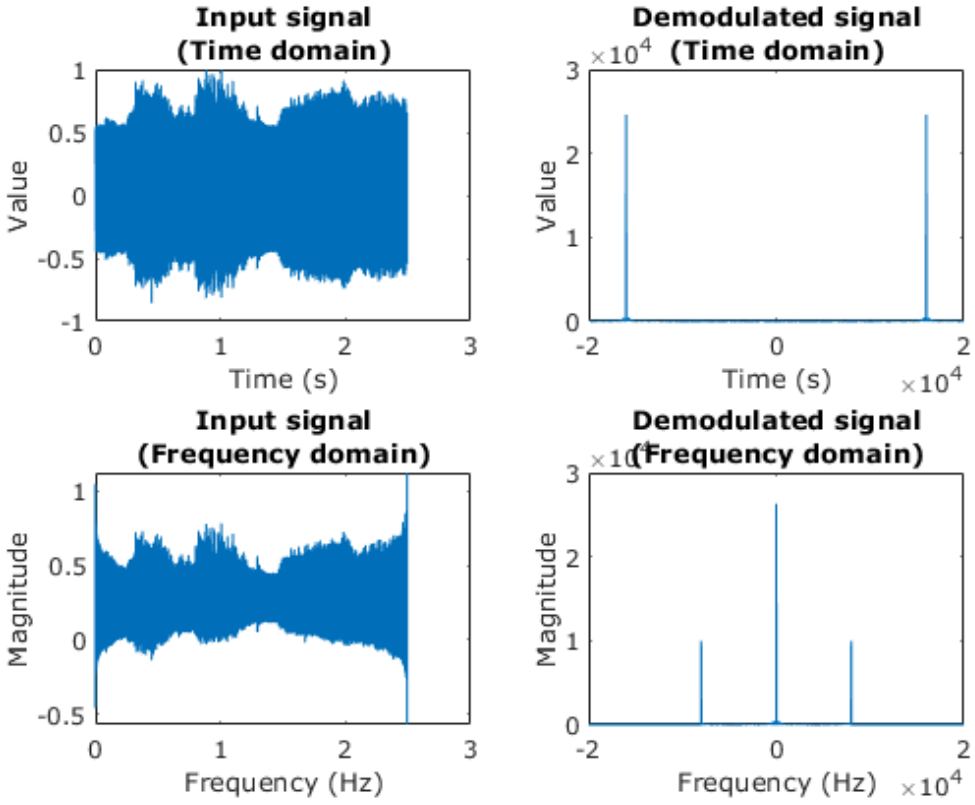
```
subplot(3,2,2);
plot(F2, abs(X2));
title(["Demodulated signal", "(Frequency domain)"]);
xlabel('Frequency (Hz)');
ylabel('Magnitude');

subplot(3,2,3);
plot(t,x2f);
title(["Filtered signal", "(Time domain)"]);
xlabel('Time (s)');
ylabel('Value');

subplot(3,2,4);
plot(F2f, abs(X2f));
title(["Filtered signal", "(Frequency domain)"]);
xlabel('Frequency (Hz)');
ylabel('Magnitude');

subplot(3,2,5);
plot(t,x);
title(["Original signal", "(Time domain)"]);
xlabel('Time (s)');
ylabel('Value');


subplot(3,2,6);
plot(F, abs(X));
title(['Original signal', '(Frequency domain)']);
xlabel('Frequency (Hz)');
ylabel('Magnitude');
```



Q4

```
[xm, Fs] = audioread("P_12_2.WAV");     %  Reading the input audio
 file provided in Moodle
t = (0:length(xm)-1)/Fs;
xm = xm';

Fc = 16000;                             % Carrier frequency of 16000,
 determined by plotting the FFT of xm
x2 = xm .* cos(2*pi*Fc*t);              % Demodulate the signal
x2f = low_pass_filter(x2, Fs, Fc/2);    % Low pass filter with cutoff
 Fc/2

[Fm,Xm] = do_fft(xm, Fs);
[F2f,X2f] = do_fft(x2f, Fs);

subplot(2,2,1);
plot(t,xm);
title(["Input signal", "(Time domain)"]);
xlabel('Time (s)');
ylabel('Value');
subplot(2,2,3);
plot(t,x2f);
title(["Input signal", "(Frequency domain)"]);
xlabel('Frequency (Hz)');
```

```
ylabel('Magnitude');
subplot(2,2,2);
plot(Fm,abs(Xm));
title(["Demodulated signal", "(Time domain)"]);
xlabel('Time (s)');
ylabel('Value');
subplot(2,2,4);
plot(F2f,abs(X2f));
title(["Demodulated signal", "(Frequency domain)"]);
xlabel('Frequency (Hz)');
ylabel('Magnitude');

playsound(x2f, Fs);
```



What we learned

We learned how to amplitude modulate and demodulate a signal, how to identify the carrier frequency of an amplitude modulated signal from it's FFT plot.

Functions

```
function playsound(y, Fs)
%PLAYSOUND blocking version of sound()
    % On Linux, refuses to play at anything other than 44100 Hz
    if isunix()
```

```
        y = resample(y,44100,Fs);
        Fs = 44100;
    end
    playblocking(audioplayer(y,Fs));
end

function [F, X] = do_fft(x,Fs)
%DO_FFT wrapper around fft, fftshift
%Returns the frequency domain and a zero-centered FFT
    NFFT = length(x);               % NFFT = length of vector
    F = Fs*(-NFFT/2:NFFT/2-1)/NFFT; % The frequency domain values
    X = fftshift(fft(x,NFFT));      % Return frequency domain values,
 zero centered
end

function y = low_pass_filter(x,Fs,cutoff)
%LOW_PASS_FILTER
    NFFT = length(x);
    X = fftshift(fft(x, NFFT));     % Put in frequency domain
    F = Fs*(-NFFT/2:NFFT/2-1)/NFFT; % The freqency domain values
    R = rectpuls(F,2*cutoff);       % Multiply by rect with pulse
 width 2*cutoff
    X = X.*R;
    y = real(ifft(ifftshift(X)));   % Return real-domain of FFT
end
```

Published with MATLAB® R2019a