
```
% BME671L Lab #5: arrays, logical indexing, linear algebra
```

```
% Your name: Ravitashaw Bathla (rb369)
```

```
% LABEL ALL AXES WHERE APPROPRIATE
```

```
close all, clear all
warning('off')
```

Q1: Use the 'dlmread' command to read the file 'lab5_spectrum.txt'.

```
lab5_spectrum = dlmread('lab5_spectrum.txt', '\t');
```

Q2: Extract the first column into a frequency vector (f_pos) and the second into a vector of Fourier coefficients (X_pos)

```
f_pos = lab5_spectrum(:, 1);
X_pos = lab5_spectrum(:, 2);
```

Q3: This file contains only the positive half of a spectrum. Use the 'wrev' and concatenation to reconstruct the full spectrum. Make sure that element corresponding to $f = 0$ appears only ONCE in the spectrum.

```
f_neg = -1*wrev(f_pos);
X_neg = wrev(conj(X_pos));
```

```
% exclude zero from one
f_neg(end)=[ ];
X_neg(end)=[ ];
```

```
% merge two vectors
f_full = [f_neg; f_pos];
X_full = [X_neg; X_pos];
```

Q4: To simplify this expansion write a function expand_spec that accepts two vectors as arguments: a vector of frequencies, a, and a vector of Fourier coefficients, b. Return two vectors p and q as outputs. Have your function check that the vectors passed as arguments are the same length. Use 'error' to stop execution if they are not. Make sure your function works even if $f = 0$ is not given.

Q5: Add comments to the beginning of the function so that the command: 'help expand_spec' (or 'help lab5>expand_spec') prints information about the function, what it does, and how to call it.

```
help expand_spec
```

```
expand_spec(f,X) reconstruct the full spectrum by appending the
conjugate
```

```
frequencies and amplitude of the provided spectrum
```

```
function call: expand_spec(f,X)
```

```
Input:
```

```
f = frequency of the spectrum
```

```
X = Amplitude of the spectrum corresponding to frequencies in f
```

```
throws error if the length of f and X are not same
```

```
Output:
```

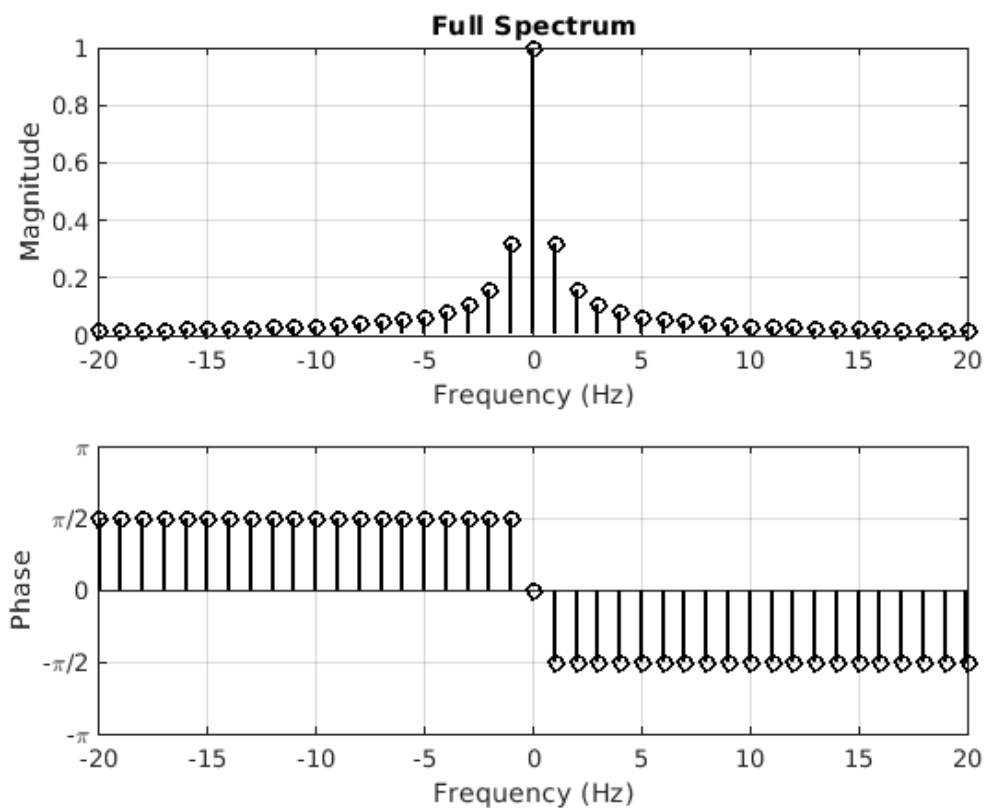
```
p = all the frequencies in the full spectrum
```

```
q = all the amplitudes in the full spectrum for the frequencies in p
```

Q6: Use your function to create the full spectrum [f, X] that corresponds to the data in 'lab5_spectrum.txt'. Plot the spectrum using the plot_spectra function you designed last week. Assume the given frequencies are in Hz.

```
[f, X] = expand_spec(f_pos, X_pos);
figure(1), clf;
plot_spectra(f, X, 'f', 'Full Spectrum');

% *****
%   SHOW YOUR SPECTRUM TO THE TA TO RECEIVE CREDIT FOR THE LAB
% *****
% *****
```



Q7: Construct a vector X_neg that contains all of the Fourier Coefficients for the negative frequencies ($f \leq 0$) from the vector X. Do NOT use conjugate symmetry, loops or if statements. Do it in ONE command.

```
X_neg = X(f <= 0);
```

Q8: what are the sizes of vectors f, X, and X_neg?

```
length(f)
length(X)
length(X_neg)
% YOUR ANSWER:
% f: 41
```

```
% X: 41
% X_neg: 21
```

```
ans =

    41
```

```
ans =

    41
```

```
ans =

    21
```

Q8: Which of the following operations are allowed? For each operation write down the size of the result and explain what mathematical operation is performed. For operations that are not allowed, explain the reason.

```
% f .* X: 41x1
% Reason: It a row by row mutiplication of two 41x1 matrices. The
number of
% rows are same in both the matrix (41)

% X * f: Not applicable
% Reason: Not allowed as the dimension of matrix is not suitable for
% multiplication. The number of columns of X should be equal to number
of
% rows of f.

% X' * f: 1x1
% Reason: It's a matrix multiplication of 1x41 with 41x1, so the
result
% is 1x1

% X * f': 41x41
% Reason: It's a matrix multiplication of 41x1 with 1x41, so the
result
% is 41x41.
```

Q9: In the previous lab we used Fourier synthesis to reconstruct a signal by hand (or method of choice). This method involved analyzing the spectral terms and determining the cosine representation. i.e.:

$$y(n) = A_k(1) \cdot \cos(w_k(1) \cdot t(n) + \phi_k(1)) + A_k(2) \cdot \cos(w_k(2) \cdot t(n) + \phi_k(2)) + \dots \text{ -- or -- } y(n) = X_k(1) \cdot \exp(j \cdot w_k(1) \cdot t(n)) + X_k(2) \cdot \exp(j \cdot w_k(2) \cdot t(n)) + \dots$$

This time, we will synthesize the signal, $y(t)$, from f and X using matrix multiplication. HINT: You do not need a for loop. HINT: there are two transpose function ($A.'$ and A') use the correct one!

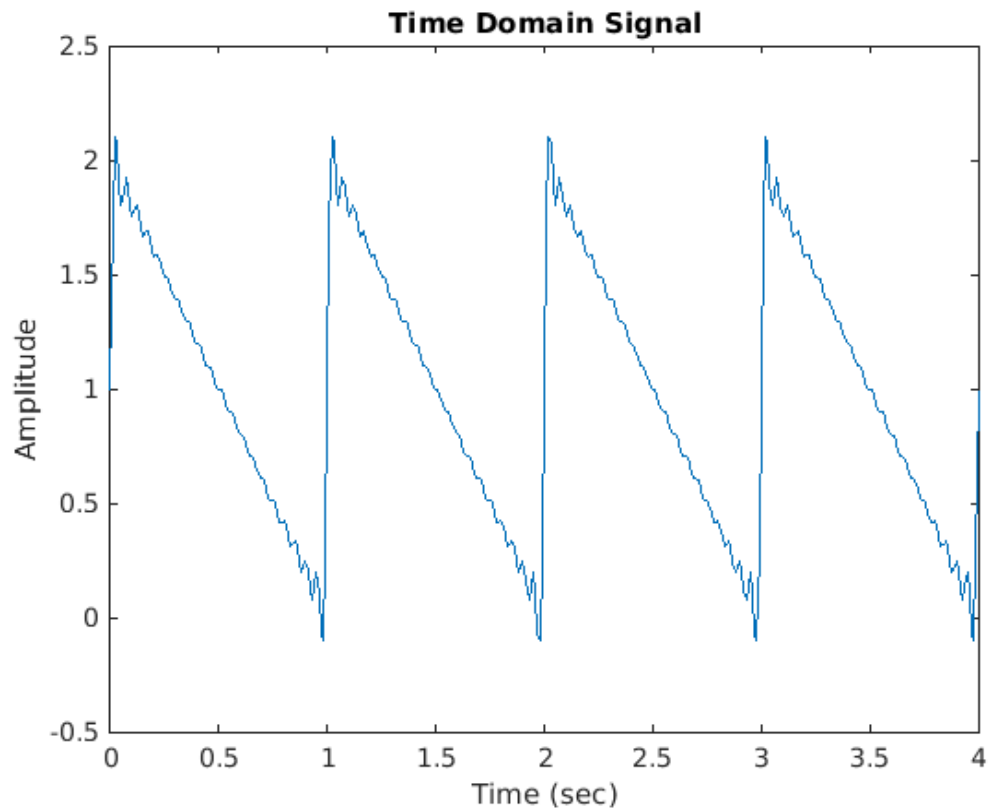
```
% Create a figure with a plot of the time-domain signal.

t = 0:0.01:4; % time vector
```

```

y = @(t) X.*exp(1j*2*pi*f*t);
figure(2), clf;
plot(t, y(t));
xlabel('Time (sec)');
ylabel('Amplitude');
title('Time Domain Signal');

```



Q10: What happens if we reconstruct the signal without using all of the frequency terms? Make a figure with 6 plots (3x2). In each plot synthesize y using a different number of frequencies $n = [1, 2, 3, 4, 8, \text{ and } 16]$ (each positive and negative pair counts as 1 frequency term, always include the DC value). * Always use the frequencies with the largest X , they contain the majority of the "power" of the signal. * The title of each plot should indicate how many frequency pairs were used * Comment on the differences between the graphs. HINT: You should be able to use a for loop instead of copy/pasting the same commands 6 times.

```

% YOUR ANSWER:
n = [1, 2, 3, 4, 8, 16];
figure(3), clf;
for i=1:length(n)
    idx = (length(X)+1)/2-n(i):(length(X)+1)/2+n(i);

    y = @(t) X(idx).'*exp(1j*2*pi*f(idx)*t);

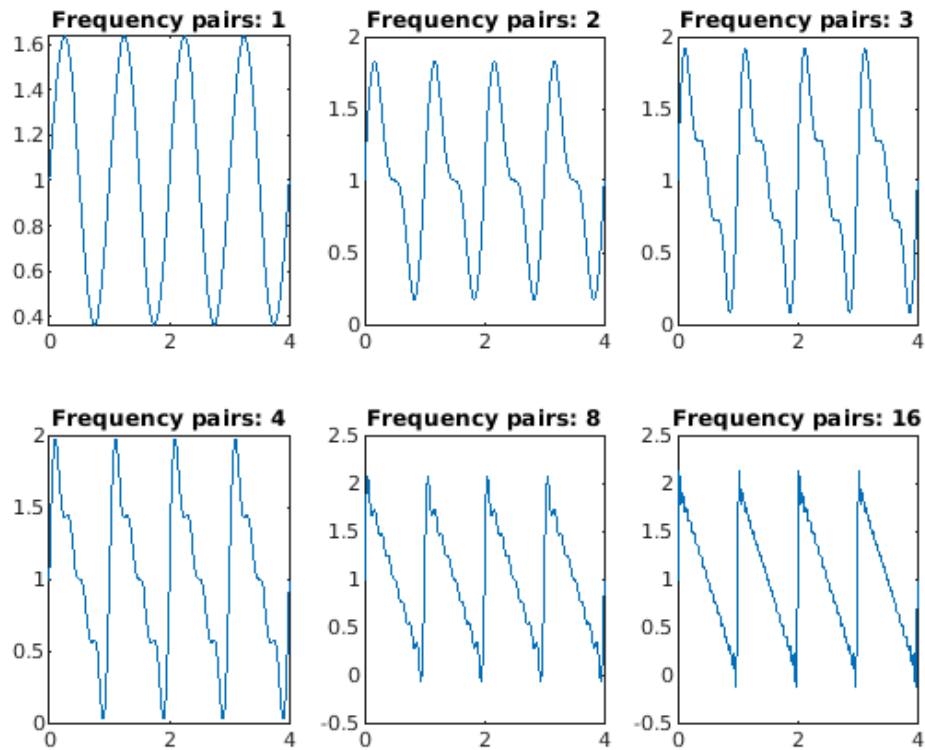
    subplot(2, 3, i, 'align');
    plot(t, y(t));
    hold on;
    title(['Frequency pairs: ', num2str(n(i))]);

```

```

    hold off;
end

```



When you are done:

```

% * Show the indicated result/figure to the TA during the lab period
%   to receive credit
% * For publishing/grading purposes include your function below
% * upload your script to Sakai
% * upload a pdf containing your script and outputs.

```

YOUR FUNCTION:

```

function [p,q] = expand_spec(f,X)
    % expand_spec(f,X) reconstruct the full spectrum by appending the
    % conjugate
    % frequencies and amplitude of the provided spectrum
    % function call: expand_spec(f,X)
    % Input:
    % f = frequency of the spectrum
    % X = Amplitude of the spectrum corresponding to frequencies in f
    % throws error if the length of f and X are not same
    % Output:
    % p = all the frequencies in the full spectrum
    % q = all the amplitudes in the full spectrum for the frequencies
    % in p

```

```
if length(f) ~= length(X)
    error("Error: Input Vectors f and X are not of same length");
end

f_conj = -1*wrev(f);
X_conj = wrev(conj(X));

% exclude common/repeated from one side
f_conj(end)=[];
X_conj(end)=[];

% merge two vectors
p = [f_conj; f];
q = [X_conj; X];
end
```

Published with MATLAB® R2020b