EE125 Matlab Project 4 Part 1: Circular Convolution

Exploring circular convolution by looking at how acousticians design concert halls

```
% Reading the wav file for the data
[x,Fs]=audioread('onscreen.wav'); % Time series, sample frequency

% Load and listen to the data
% soundsc(x,Fs) % Guy says on screen

% Plot the data
tx = 0:1/Fs:(length(x)-1)/Fs;
figure(1)
subplot(2,1,1)
plot(tx,x)
ylabel('Signal')
xlabel('Time (sec)')
title('Original Signal')

% Load the small church impulse response, sample rate matches
load('smallChurch_fs22k.mat')
h = hChurch;

% Plot church's impulse response
th = 0:1/Fs:(length(h)-1)/Fs;
subplot(2,1,2)
plot(th,h)
ylabel('Signal')
xlabel('Time (sec)')
title('Church Impulse Response')

suptitle('Plots for steps 1 and 2')

% Use linear convolution with the impulse response to see what the
 signal
% would sound like in a small church
s = conv(h,x);
% soundsc(s,Fs)
```

This signal sounds much "deeper" or "fuller" - like there is some sort of echo, which makes sense when looking at what the impulse response of the church is, the convolution will have some sort of decreasing repetition of copies overlaid of the original signal giving it an echo effect

In order to have FFT based circular convolution equal to linear convolution, you just have to pad the signals being convolved to be the same length and linear conv. So for this case, the math is shown below:

```
L = length(x);
M = length(h);
N = M + L - 1;
xpad = [x; zeros(M-1,1)];
hpad = [h; zeros(L-1,1)];
```

```
% With linear convolution the length of the output is going to be the
 sum
% of the length of the inputs minus the overlap, which in this case is
 1
xp = [x; zeros(M-L,1)];
cc = ifft(fft(xp).*fft(h));
ccp = ifft(fft(xpad).*fft(hpad));

% Checking that the padded circular conv is the same as the linear
 conv
 MSE_Padded_Circ_Conv_Lin_Conv = immse(ccp,s) % ~0

% Listening to the two outputs
% soundsc(cc,Fs)
% soundsc(ccp,Fs)
% Maybe I did this wrong, but I can't really hear much of a difference
% between the two signals. I know that there is a difference because
 there
% will be aliasing when using, I guess after listening to it several
 times
% there is some sort of cutoff at the end of the circular convolution
% without padding, but it could just be a sort of placebo that I'm
 hearing

% Plot the required stuff for step 5
figure(2)

% The original signal
subplot(5,1,1)
plot(tx,x)
ylabel('Signal')
xlabel('Time (sec)')
title('Original Signal')

% The linearly convolved signal
ts = 0:1/Fs:(length(s)-1)/Fs;
subplot(5,1,2)
plot(ts,s)
ylabel('Signal')
xlabel('Time (sec)')
title('Linearly Convolved Signal')
axis([0 2.5 -10 10])

% The difference between linear conv and fft based padded conv
diff = s - ccp;
subplot(5,1,3)
plot(ts,diff)
ylabel('Difference')
xlabel('Time (sec)')
title('Difference Between Linear Conv and FFT Based Padded Conv')
axis([0 2.5 -10 10])

% FFT based convolution without padding
```

```
tcc = 0:1/Fs:(length(cc)-1)/Fs;
subplot(5,1,4)
plot(tcc,cc)
ylabel('Signal')
xlabel('Time (sec)')
title('Circular Conv Without Padding')

% The difference between linear conv and unpadded circ conv
diff2 = s(1:length(cc)) - cc;
subplot(5,1,5)
plot(tcc,diff2)
ylabel('Signal')
xlabel('Time (sec)')
title('Difference Between Linear Conv and Unpadded Circular Conv')
```

Based off of the graphs, the time domain aliasing is going to occur mostly in the beginning of the signal, since thats where the difference between the circular unpadded and linear convolutions are the greatest. This makes sense because the aliasing occurs because of the periodic nature of the tranformations so the signal should "leak" over into the left side of the time domain, but I was expecting the aliasing to be more pronounced than it was.

Published with MATLAB® R2016b