3.4

Table of Contents

Template of sigBeat Struct

sigBeat.Amp = 10; %-- B in Equation (3) sigBeat.fc = 480; %-- center frequency in Eq. (3) sigBeat.phic = 0; %-- phase of 2nd sinusoid in Eq. (3) sigBeat.fDelt = 20; %-- modulating frequency in Eq. (3) sigBeat.phiDelt = -2*pi/3; %-- phase of 1st sinusoid Eq. (3) sigBeat.t1 = 1.1; %-- starting time sigBeat.t2 = 5.2; %-- ending time %

```
%----- extra fields for the parameters in Equation (4)
%
% sigBeat.f1        %-- frequencies in Equation (4)
% sigBeat.f2        %--
% sigBeat.X1        %-- complex amps for sinusoids in Equation (4)
% sigBeat.X2        %-- derived from A's and phi's
%
% sigBeat.values    %-- vector of signal values sigBeat.times
% sigBeat.times     %-- vector of corresponding times
```

3.4(a)

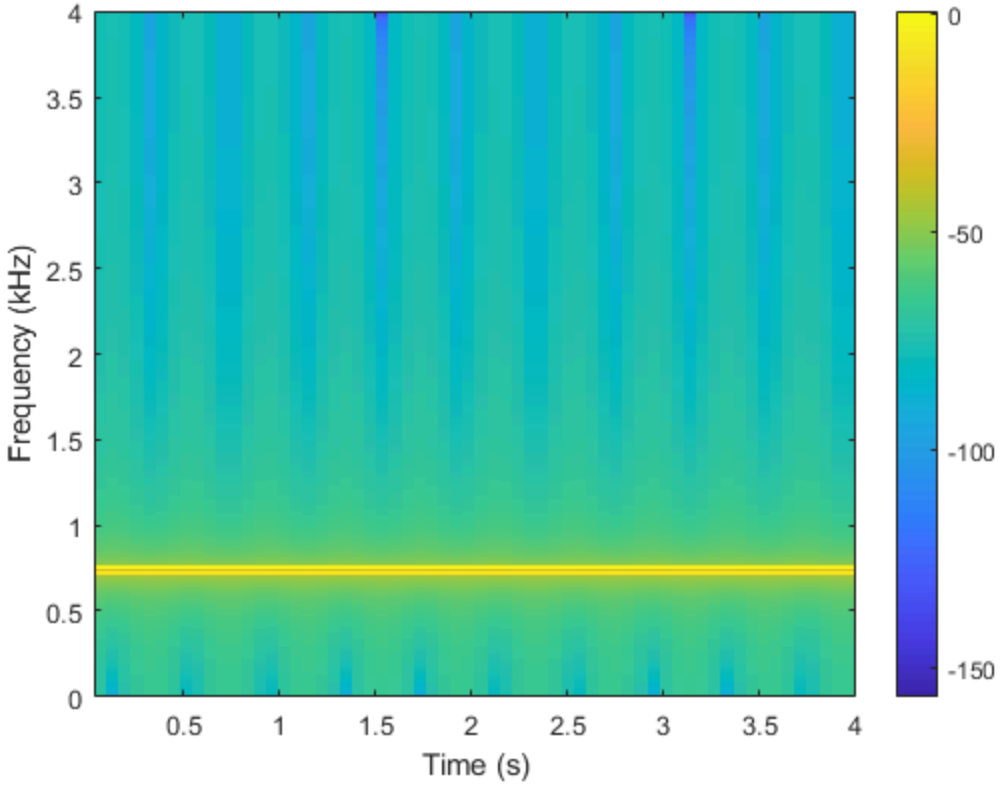
Complete the sum2BeatStruct function at the end

3.4(b)

Create a beat signal with two frequency components: one at 720 Hz and one at 750 Hz

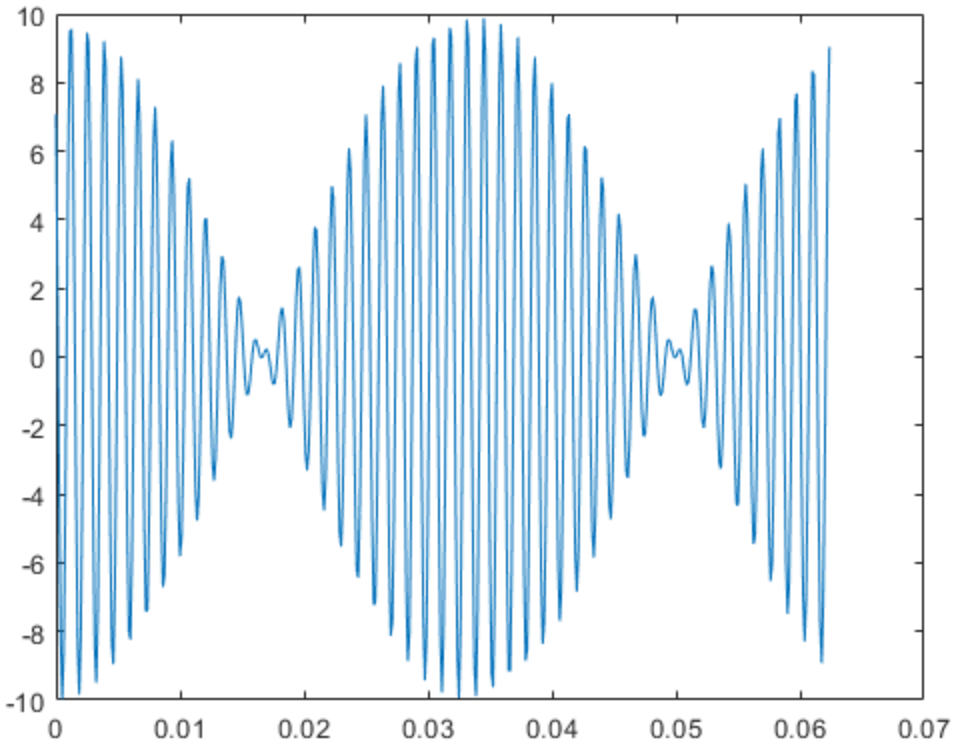
```
fs = 8000;
sigBeat.Amp = 10; %-- B in Equation (3)
sigBeat.fc = 735; %-- center frequency in Eq. (3)
sigBeat.phic = pi/4; %-- phase of 2nd sinusoid in Eq. (3)
sigBeat.fDelt = 15; %-- modulating frequency in Eq. (3)
sigBeat.phiDelt = 0; %-- phase of 1st sinusoid Eq.~(3)
sigBeat.t1 = 0; %-- starting time
sigBeat.t2 = 4.04; %-- ending time %

testingBeat = sum2BeatStruct( sigBeat );
testingBeat.times = sigBeat.t1:1/fs:sigBeat.t2;
testingBeat.values =
 real( testingBeat.X1*exp(1j*2*pi*testingBeat.f1*testingBeat.times) ...
```

```
    + testingBeat.X2*exp(1j*2*pi*testingBeat.f2*testingBeat.times) );

figure
spectrogram(testingBeat.values,1024,[ ],[ ],fs,'yaxis'); colorbar
soundsc(testingBeat.values, fs)
```



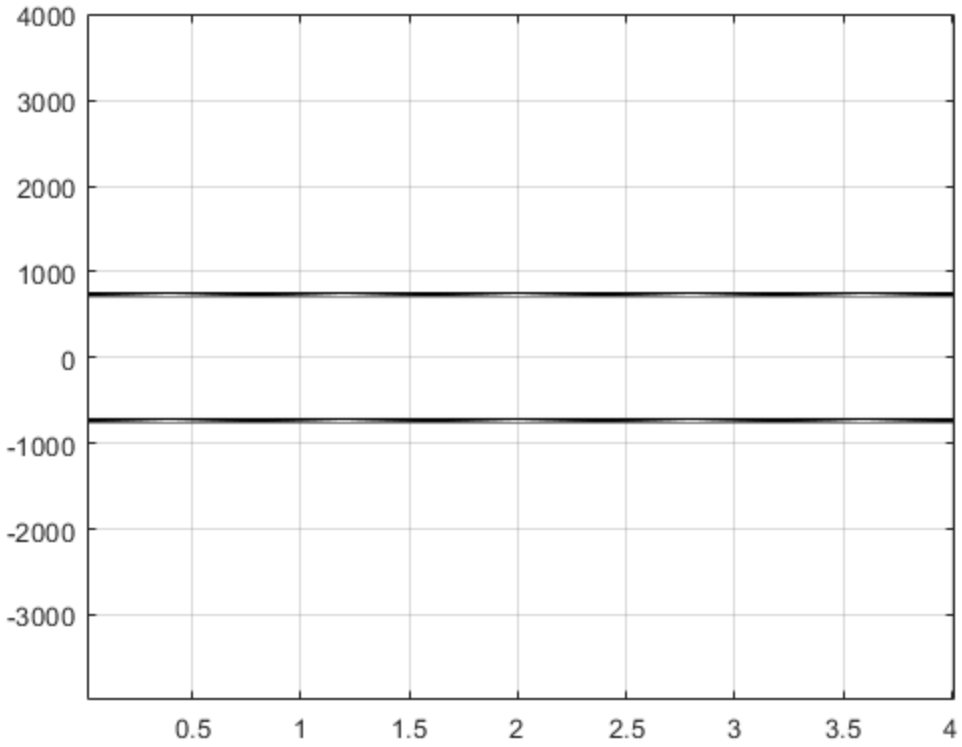
3.4.1(a)

```
fs = 8000;
sigBeat.Amp = 10; %-- B in Equation (3)
sigBeat.fc = 735; %-- center frequency in Eq. (3)
sigBeat.phic = pi/4; %-- phase of 2nd sinusoid in Eq. (3)
sigBeat.fDelt = 15; %-- modulating frequency in Eq. (3)
sigBeat.phiDelt = 0; %-- phase of 1st sinusoid Eq.~(3)
sigBeat.t1 = 0; %-- starting time
sigBeat.t2 = 4.04; %-- ending time %

testingBeat = sum2BeatStruct( sigBeat );
testingBeat.times = sigBeat.t1:1/fs:sigBeat.t2;
testingBeat.values =
 real(testingBeat.X1*exp(1j*2*pi*testingBeat.f1*testingBeat.times) ...
    + testingBeat.X2*exp(1j*2*pi*testingBeat.f2*testingBeat.times) );

figure
plot( testingBeat.times(1:500), testingBeat.values(1:500) )
```



3.4.1(c)

```
plotspec(testingBeat.values+j*1e-12,fs,512); grid on, shg
```




```
function sigBeatSum = sum2BeatStruct( sigBeatIn ) %
%--- Assume the five basic fields are present, plus the starting and ending
 times
%--- Add the four fields for the parameters in Equation (4)
%
% sigBeatSum.f1, sigBeatSum.f2, sigBeatSum.X1, sigBeatSum.X2

sigBeatSum.f1 = sigBeatIn.fc + sigBeatIn.fDelt;
sigBeatSum.f2 = sigBeatIn.fc - sigBeatIn.fDelt;
% Amplitude --> See Eq. (4)
A1 = sigBeatIn.Amp/2;
A2 = sigBeatIn.Amp/2;
% Phase --> See Eq. (4)
phi1 = sigBeatIn.phic + sigBeatIn.phiDelt;
phi2 = sigBeatIn.phic - sigBeatIn.phiDelt;
% Compute complex amplitude
sigBeatSum.X1 = A1 * exp(1j * phi1);
sigBeatSum.X2 = A2 * exp(1j * phi2);

end
```

Published with MATLAB® R2022a