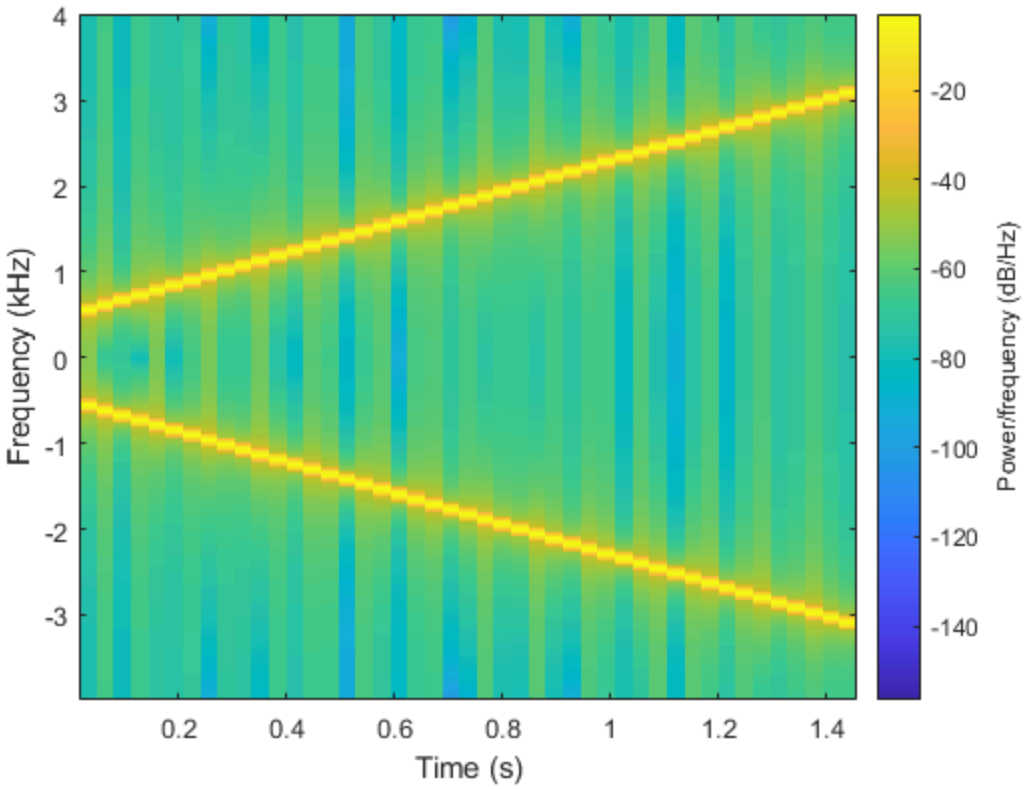
3.5

```
myLFMsig.f1 = 500;
myLFMsig.t1 = 0;
myLFMsig.t2 = 1.5;
myLFMsig.slope = 1800;
myLFMsig.complexAmp = 10*exp(j*0.3*pi);
dt = 1/8000; % 8000 samples per sec is the sample rate
outLFMsig = makeLFMvals(myLFMsig,dt);
%- Plot the values in outLFMsig
plot(outLFMsig.times(1:500), outLFMsig.times(1:500))
%- Make a spectrogram for outLFMsig to see the linear frequency change
spectrogram(outLFMsig.values, 512,[ ],[ ],1/dt,'centered','yaxis')


function sigOut = makeLFMvals( sigLFM, dt )
% MAKELFMVALS       generate a linear-FM chirp signal
%
% usage: sigOut = makeLFMvals( sigLFM, dt )
% sigLFM.f1 = starting frequency (in Hz) at t = sigLFM.t1
% sigLFM.t1 = starting time (in secs)
% sigLFM.t2 = ending time
% sigLFM.slope = slope of the linear-FM (in Hz per sec)
% sigLFM.complexAmp = defines the amplitude and phase of the FM signal
% dt = time increment for the time vector, typically 1/fs (sampling frequency)
%
% sigOut.values = (vector of) samples of the chirp signal
% sigOut.times = vector of time instants from t=t1 to t=t2
%
if( nargin < 2 ) %-- Allow optional input argument for dt
    dt = 1/8000; %-- 8000 samples/sec
end

%--------NOTE: use the slope to determine mu needed in psi(t)
%-------- use f1, t1 and the slope to determine f0 needed in psi(t)
tt = sigLFM.t1:dt:sigLFM.t2;
mu = sigLFM.slope/2;
f0 = sigLFM.f1;
psi = 2*pi*( f0*tt + mu*tt.*tt);
xx = real( sigLFM.complexAmp * exp(1j*psi) );
sigOut.times = tt;
sigOut.values = xx;

end
```



Published with MATLAB® R2022a