SIO227A Homework 6: 7.1 (Eric Gallimore)

Table of Contents

Setup 1
Find parameters for LFM chirp from 1-4Hz, 20s long 1
Generate LFM chirp 1
Autocorrelation of LFM Probe 4
Do it again with a less extreme taper 4
Applying autocorrelation a second time 8

Setup

```
close all;
```

Find parameters for LFM chirp from 1-4Hz, 20s long

Note that the chirp rate is the instantaneous frequency. Instantaneous frequency is the derivative of the phase, so we integrate the instantaneous frequency function to get the corresponding phase function. b is a phase parameter, and it conveiently works out to be k/2.

```
fs = 100; %Hz
t = 0:(1/fs):20; %s
% starting frequency
f0 = 1; % Hz

% chirp rate, b
chirp_rate = 3 / 20; % Hz/s
b = chirp_rate/2;

% Print the values
fprintf('f0 = %d Hz\n', f0);
fprintf('b = %d\n', b);
```

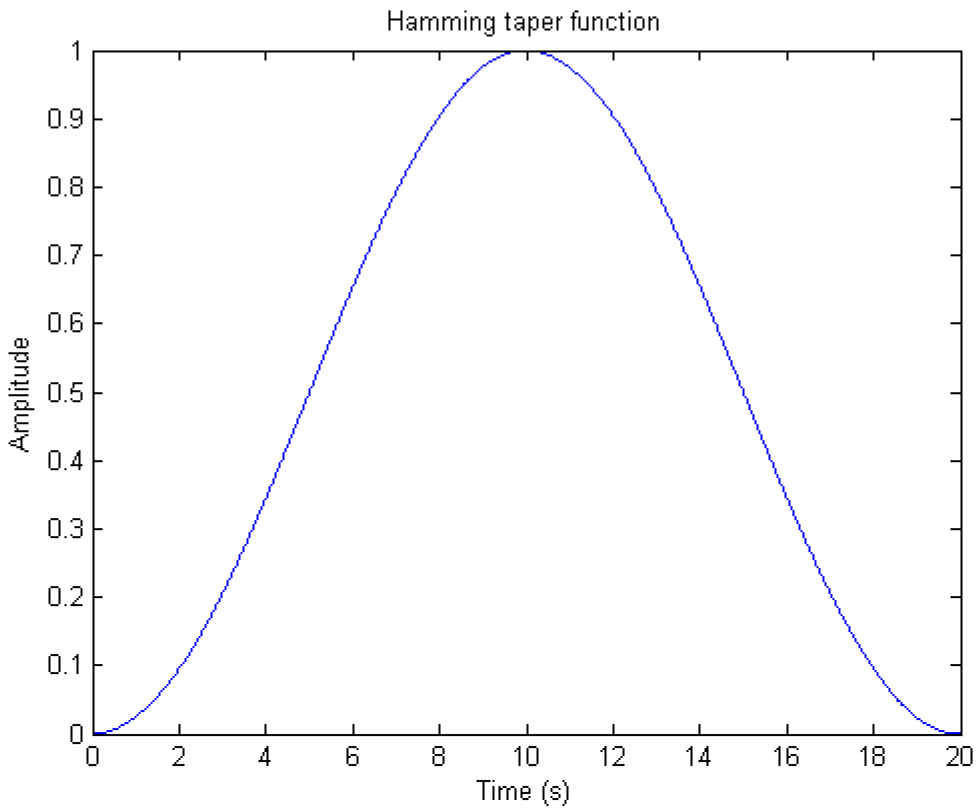
```
f0 = 1 Hz
b = 7.500000e-02
```

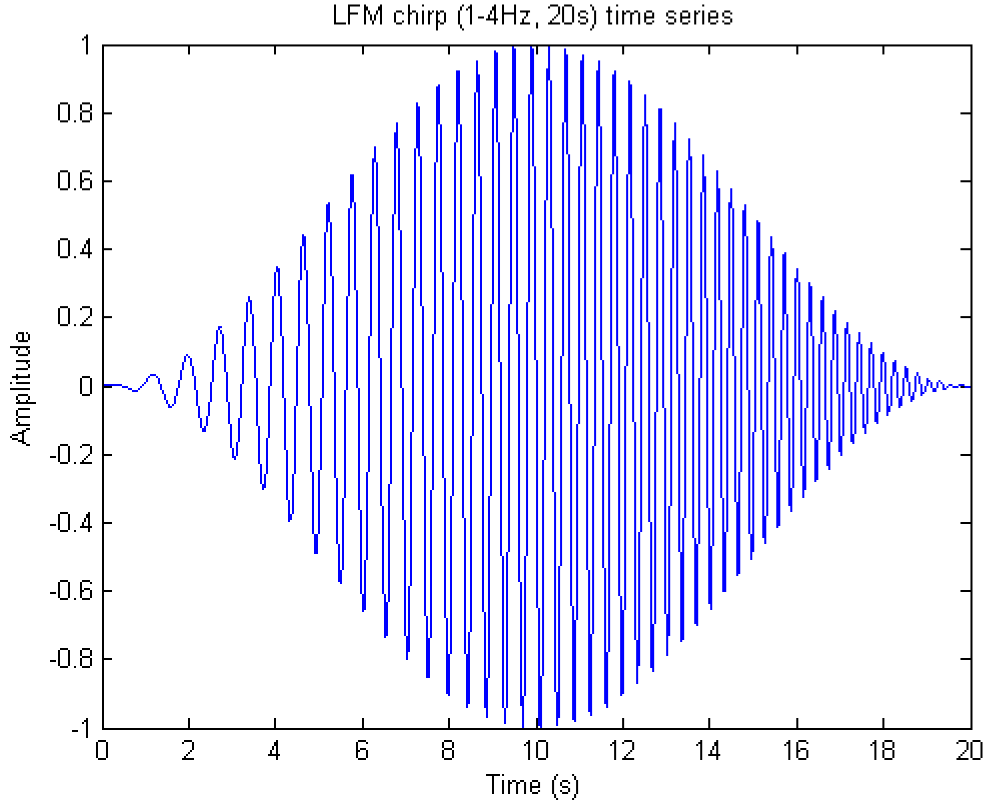
Generate LFM chirp

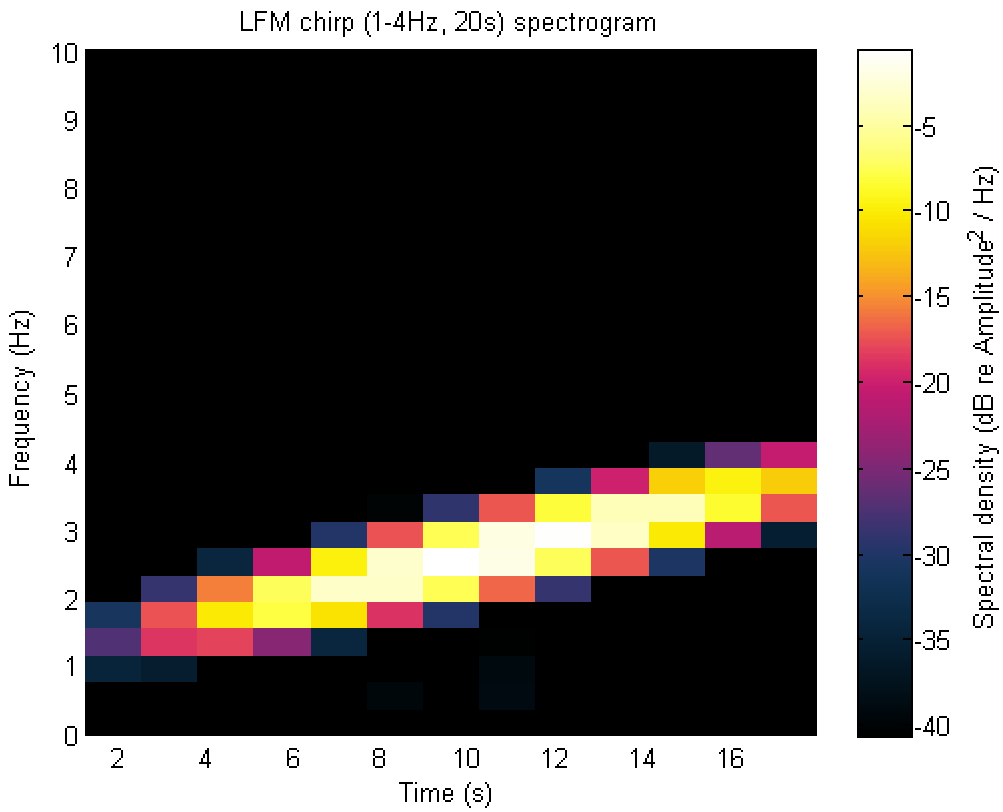
Hamming taper with 20s interval

```
A = sin(pi*t/20).^2;
```

```
% Verify that the Hamming function looks like a Hamming function...
figure('name', 'Hamming taper time series');
plot(t,A);
ylabel('Amplitude');
xlabel('Time (s)');
title('Hamming taper function');

% Make the sweep
v = A.*sin(2*pi*(f0 + b*t).*t);

% Plot the time series
figure('name', 'LFM chirp time series');
plot(t, v);
ylabel('Amplitude');
xlabel('Time (s)');
title('LFM chirp (1-4Hz, 20s) time series');

% Plot a spectrogram to verify that we got the frequency right.
figure('name', 'LFM chirp spectrogram');
plot_nice_spectrogram(v, 256, fs, 40, 'Amplitude^2');
ylim([0 10]);
title('LFM chirp (1-4Hz, 20s) spectrogram');
```



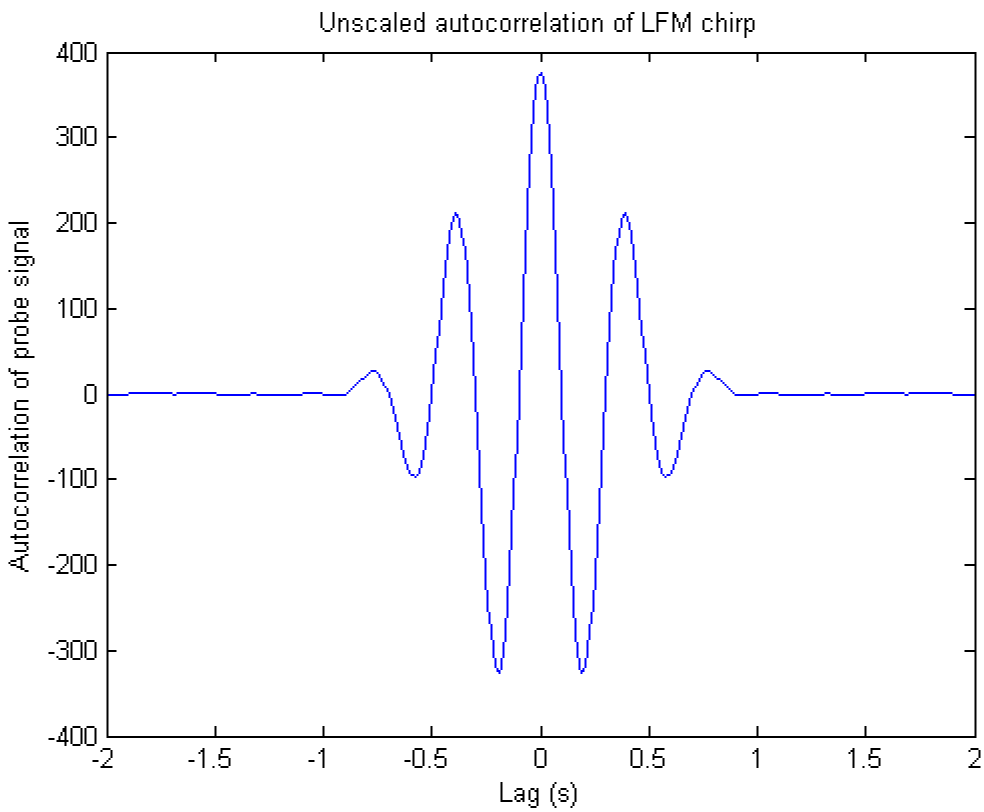

LFM chirp (1-4Hz, 20s) time series
Amplitude
Time (s)
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1
0
2
4
6
8
10
12
14
16
18
20


LFM chirp (1-4Hz, 20s) spectrogram
Frequency (Hz)
Time (s)
Spectral density (dB re Amplitude^2 / Hz)
10
9
8
7
6
5
4
3
2
1
0
2
4
6
8
10
12
14
16
-5
-10
-15
-20
-25
-30
-35
-40

Autocorrelation of LFM Probe

plot the autocorrelation of the chirp

```
figure('name', 'Autocorrelation of chirp')
[ac_probe, ac_probe_lags] = xcorr(v);
% Turn lags into seconds
lags_s = ac_probe_lags / fs;
plot(lags_s, ac_probe);
xlim([-2 2]);
ylabel('Autocorrelation of probe signal');
xlabel('Lag (s)')
title('Unscaled autocorrelation of LFM chirp');
```

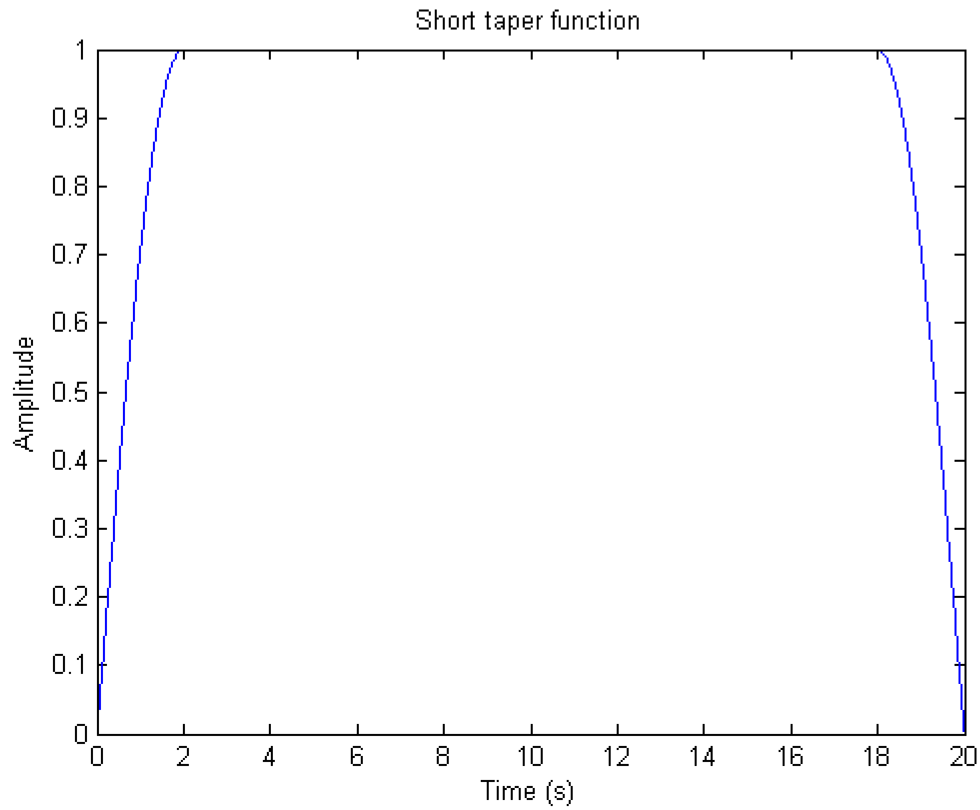


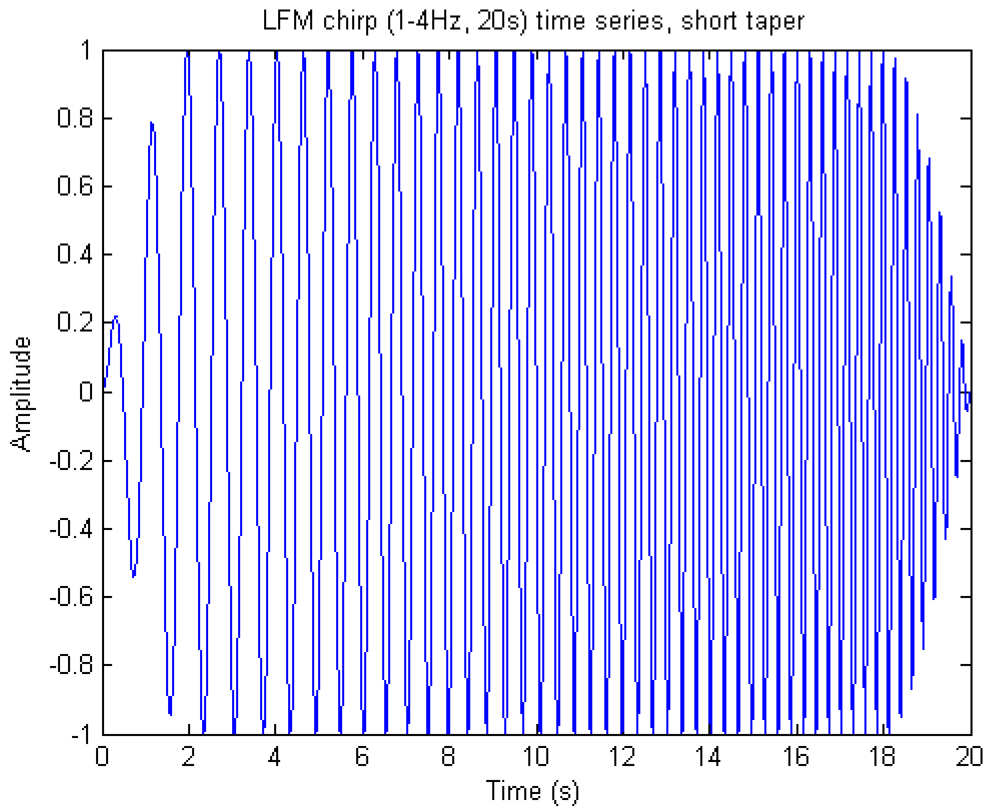
Do it again with a less extreme taper

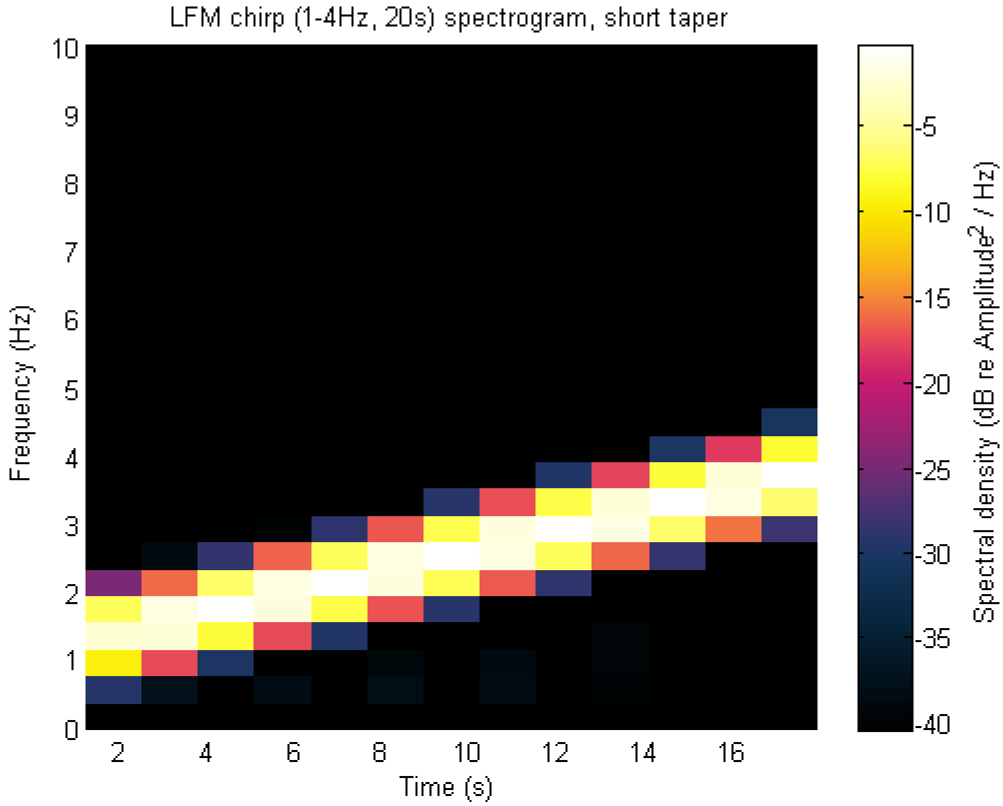
Unity from 2-18 seconds, taper at ends

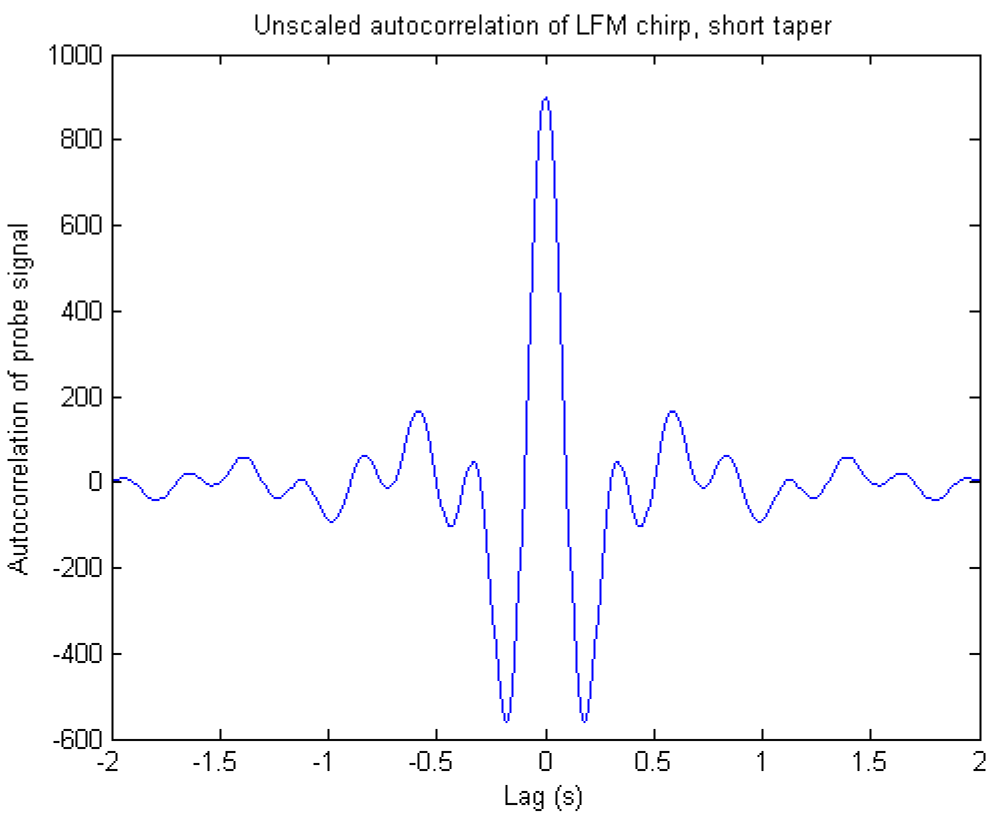
```
A_short = [sin(pi*t(t<2)/4) ones(1,length(t(2<=t&t<=18))) sin(pi*t(t>18)/4)];

% Plot to to make sure it makes sense.
figure('name', 'Short taper time series');
plot(t,A_short);
ylabel('Amplitude');
xlabel('Time (s)');
title('Short taper function');
```

```
% Make the sweep with the short taper
v_short = A_short.*sin(2*pi*(f0 + b*t).*t);

% Plot the time series
figure('name', 'LFM chirp time series, short taper');
plot(t, v_short);
ylabel('Amplitude');
xlabel('Time (s)');
title('LFM chirp (1-4Hz, 20s) time series, short taper');

% Plot a spectrogram to verify that we got the frequency right.
figure('name', 'LFM chirp spectrogram, short taper');
plot_nice_spectrogram(v_short, 256, fs, 40, 'Amplitude^2');
ylim([0 10]);
title('LFM chirp (1-4Hz, 20s) spectrogram, short taper');

% Autocorrelation of LFM Probe
% plot the autocorrelation of the chirp
figure('name', 'Autocorrelation of chirp, short taper')
[ac_probe_short, ac_probe_lags_short] = xcorr(v_short);
% Turn lags into seconds
lags_s_short = ac_probe_lags_short / fs;
plot(lags_s_short, ac_probe_short);
xlim([-2 2]);
ylabel('Autocorrelation of probe signal');
xlabel('Lag (s)')
title('Unscaled autocorrelation of LFM chirp, short taper');
```


Short taper function
Amplitude
Time (s)
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
0
2
4
6
8
10
12
14
16
18
20


LFM chirp (1-4Hz, 20s) time series, short taper
Amplitude
Time (s)
-1
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8
1
0
2
4
6
8
10
12
14
16
18
20

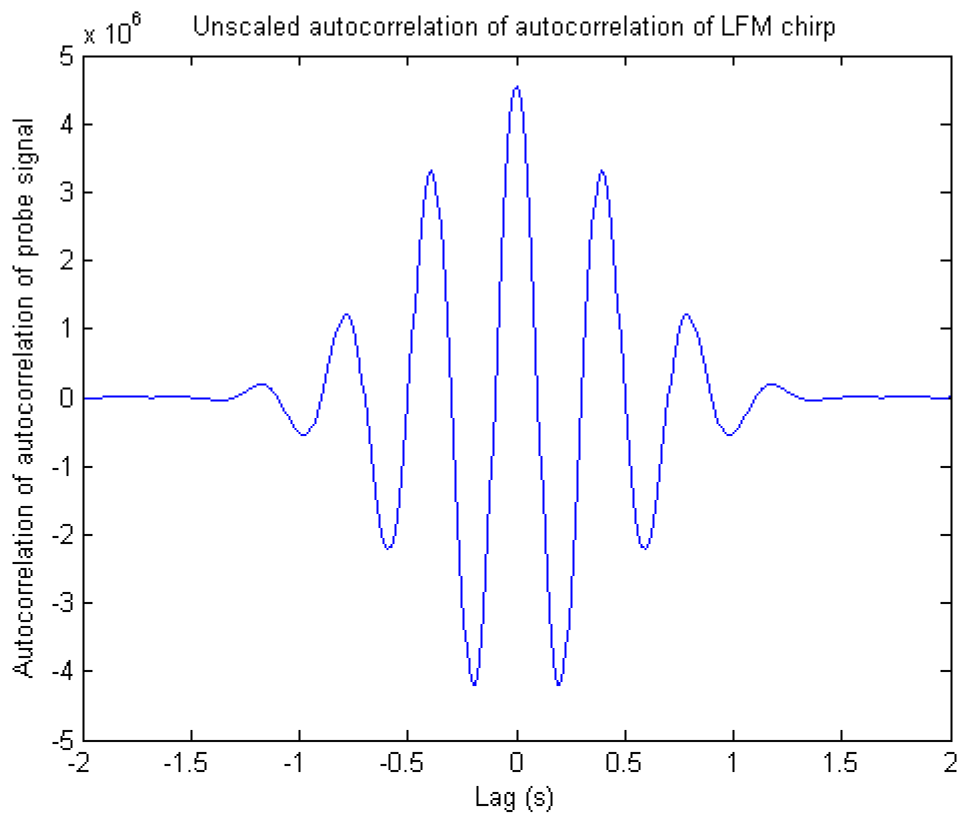

LFM chirp (1-4Hz, 20s) spectrogram, short taper
Frequency (Hz)
Time (s)
Spectral density (dB re Amplitude^2 / Hz)
0
1
2
3
4
5
6
7
8
9
10
2
4
6
8
10
12
14
16
-5
-10
-15
-20
-25
-30
-35
-40


Unscaled autocorrelation of LFM chirp, short taper
Autocorrelation of probe signal
Lag (s)
1000
800
600
400
200
0
-200
-400
-600
-2
-1.5
-1
-0.5
0
0.5
1
1.5
2

Applying autocorrelation a second time

We autocorrelate the autocorrelation of the probe. In doing so, we illustrate that this is not a good idea if the goal is to approximate a delta function, since the autocorrelation of the autocorrelation is less impulsive. The LFM chirp is a handy function to use for matched filtering because its autocorrelation approximates a delta function (particularly when using a Hilbert transform and using it to estimate the impulse response of a channel). Many signals do now have autocorrelations that approximate delta functions, and the autocorrelation of the LFM chrip is one of those signals.

```
% plot the autocorrelation of the autocorrelation of the chirp
figure('name', 'Autocorrelation of autocorrelation of chirp')
[ac_probe_twice, ac_probe_lags_twice] = xcorr(ac_probe);
% Turn lags into seconds
lags_s_twice = ac_probe_lags_twice / fs;
plot(lags_s_twice, ac_probe_twice);
xlim([-2 2]);
ylabel('Autocorrelation of autocorrelation of probe signal');
xlabel('Lag (s)')
title('Unscaled autocorrelation of autocorrelation of LFM chirp');
```



Published with MATLAB® R2013b