```
% problem 1a
A = [1 1];
B = [1 -1];
C = xcorr(A) + xcorr(B);
disp(C);
% C is a scaled impulse


% problem 1b

D = xcorr([A B]) + xcorr([A -B]);
disp(D)
% D is a scaled impulse
```

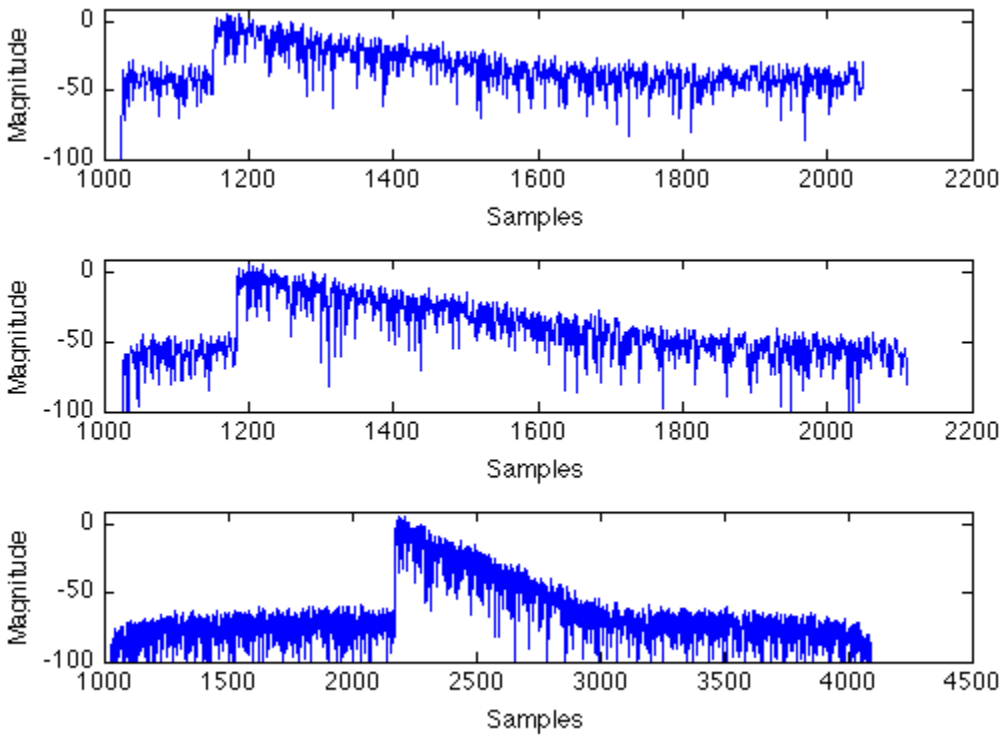
```
     0     4     0

     0     0     0     8     0     0     0
```

Published with MATLAB® 7.10

```
function [ a, b ] = golay( n )
%GOLAY Summary of this function goes here
%   Detailed explanation goes here


A = 1;
B = 1;

for i=1:n
    A_new = [A B];
    B_new = [A -B];
    A = A_new;
    B = B_new;
end

a=A;
b=B;

end
```

Published with MATLAB® 7.10

```
% problem 2b

n=[0, 5, 10];
titletext = ['a','b','c'];
for i=1:length(n)
    [a,b] = golay(n(i));
    a = [a zeros(1,1024)];
    b = [b zeros(1,1024)];
    response_a = xcorr(hmeasure(a),a);
    response_b = xcorr(hmeasure(b),b);
    resp = (response_a+response_b)/2^n(i);
    subplot(3,1,i), plot(20*log10(abs(resp)));
    ylim([-100 10]);
    ylabel('Magnitude');
    xlabel('Samples');

end


% problem 2c

%  SNR for golay code length 1 = 40 db
%  SNR for golay code length 32 = 55 db
%  SNR for golay code length 1024 = 70 db
```

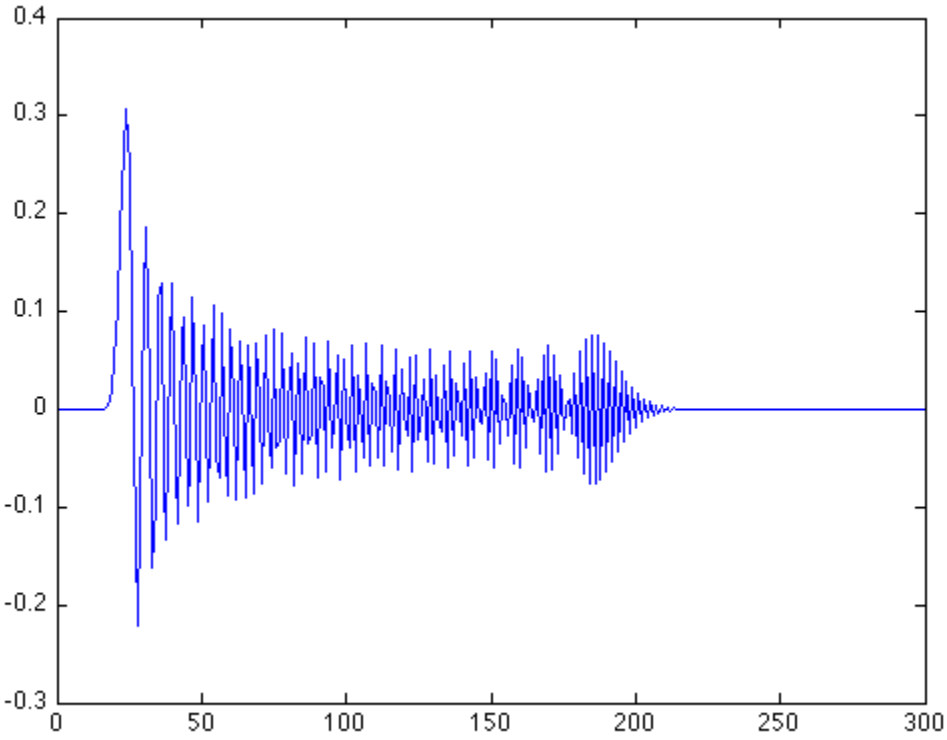


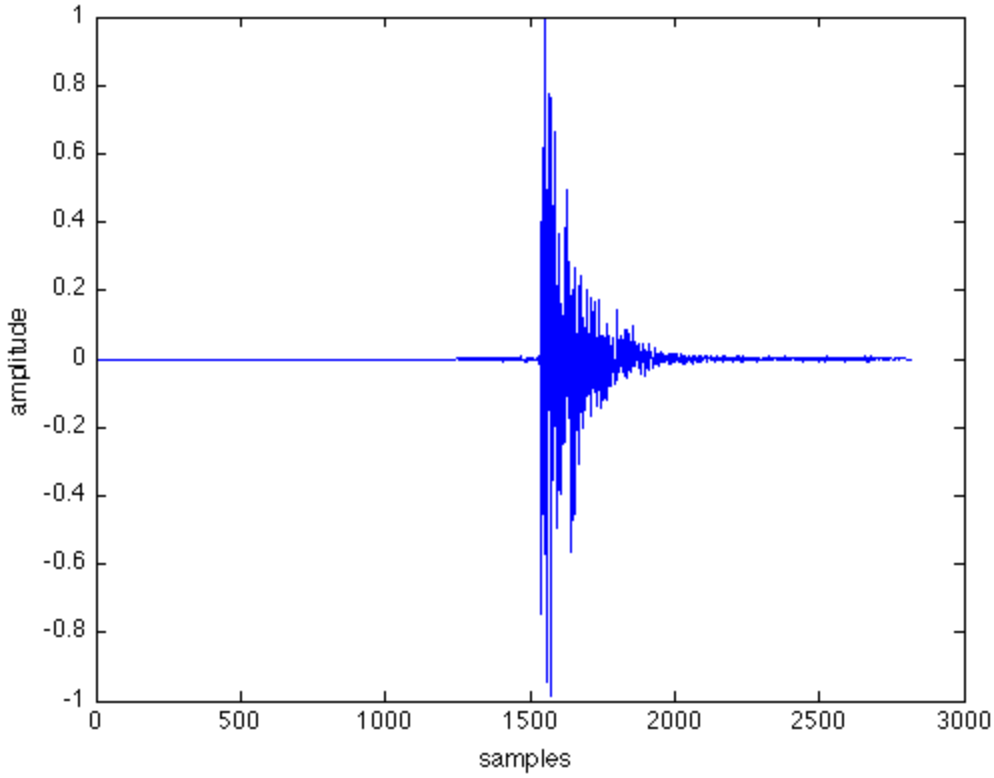
Published with MATLAB® 7.10

```
function [ h_prime ] = gCascade( rho, n )
%GCASCADE Summary of this function goes here
%   Detailed explanation goes here


G1_B = [ rho 1 ];
G1_A = [ 1 rho ];

[h,t] = impz(G1_B,G1_A);

h_prime = 1;

for i=1:n
  h_prime  = conv(h_prime,h);
end

end
```

Published with MATLAB® 7.10

```
% problem 3a

h = gCascade(0.5,64);

% Matlab-compatible plot:
plot(h);
xlim([0 300]);

% problem 3b
min_rho=-1;
min = 10000;
for rho = 0.25:0.0001:0.3
   h = gCascade(rho,64);
   m = max(h);
   if (min>m)
       min=m;
       min_rho = rho;
   end
end

% problem 3c

h64 = gCascade(min_rho,64);
rhoGain = 1/max(abs(h64));
h_chirp = hmeasure(rhoGain*[h64' zeros(1,1024)]);
h_impulse = xcorr(h_chirp,h64)/rhoGain;
figure()
plot(h_impulse);

ylabel('amplitude');
xlabel('samples');

% problem 3d

% SNR for chirp measurement = 50 dB as compared to 70 dB for 1024 length
% Go
```


0.4
0.3
0.2
0.1
0
-0.1
-0.2
-0.3
0
50
100
150
200
250
300


1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1
amplitude
0
500
1000
1500
2000
2500
3000
samples

```
% problem 4a

% read golay signals
[g_a,fs] = wavread('gcC4A02105a.wav');
g_b = wavread('gcC4A02105b.wav');

% read room responses
g_a_response = wavread('gcAresponse.wav');
g_b_response = wavread('gcBresponse.wav');

% only need first channel
g_a_response = g_a_response(:,1)';

g_a_response = [g_a_response zeros(1,length(g_b_response) - length(g_a_response))]

g_a_response = [g_a_response zeros(1,2048)];
g_b_response = [g_b_response' zeros(1,2048)];


% MAKE BOTH THE SAME SIZE
g_a_corr = fftfilt(g_a(end:-1:1),g_a_response);
g_b_corr = fftfilt(g_b(end:-1:1),g_b_response);

irhatg = (g_a_corr + g_b_corr ) ./ (2*(length(g_a)));

wavwrite(irhatg,fs,16,'out.wav');

ftgram(irhatg, fs, 'music', 'waveform', true, 'dbrange', 80);

% problem 4b

[chirp,fs] = wavread('ssl_48_18.wav');
[chirp_response,fs] = wavread('ssl_response.wav');
figure_8 = [chirp_response(:,1)' zeros(1,1024)];
omni = [chirp_response(:,2)' zeros(1,1024)];

figure()
spectrogram(figure_8,1024,'yaxis');
title('figure8 chirp');

figure()
spectrogram(omni,1024,'yaxis');
title('omni chirp');


% the few lines that we see are unexpected

impulse_response_omni = fftfilt(chirp,omni(end:-1:1));
impulse_response_figure_8 = fftfilt(chirp,figure_8(end:-1:1));

impulse_response_omni = [zeros(1,2048) impulse_response_omni(end:-1:1)];
impulse_response_figure_8 = [zeros(1,2048) impulse_response_figure_8(end:-1:1)];

figure()
irhat1 = [impulse_response_omni' impulse_response_figure_8'];


ftgram(irhat1, fs, 'music', 'waveform', true, 'dbrange', 80);
```

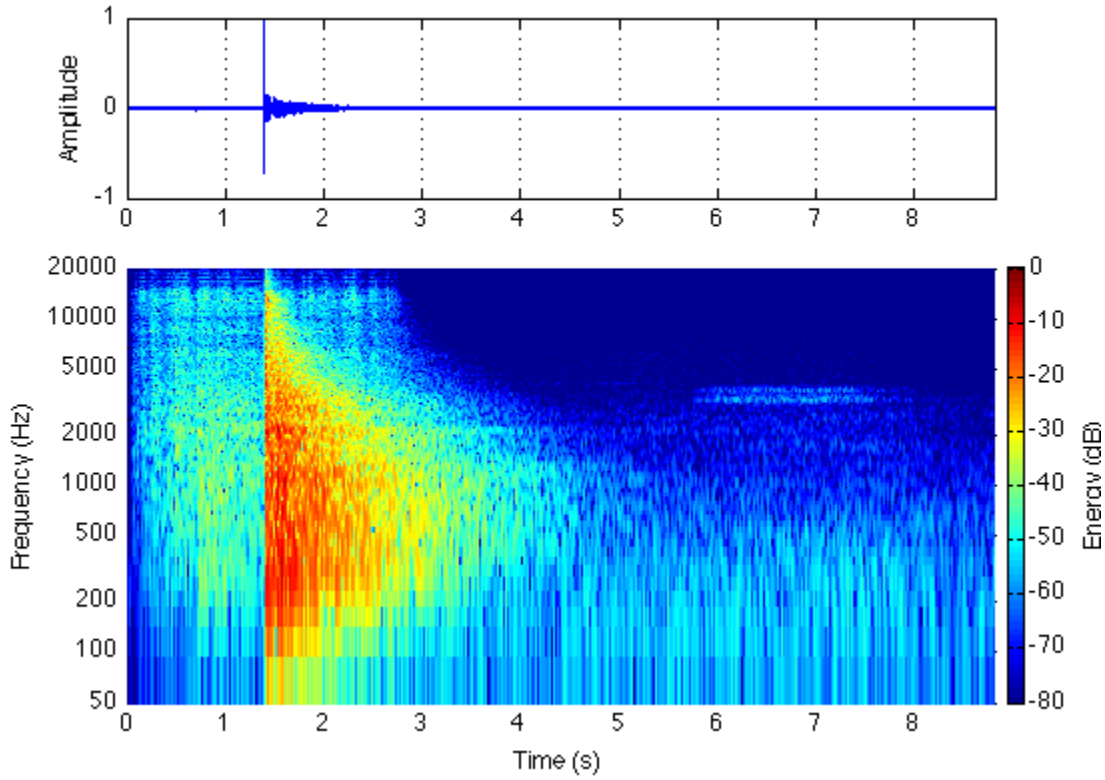
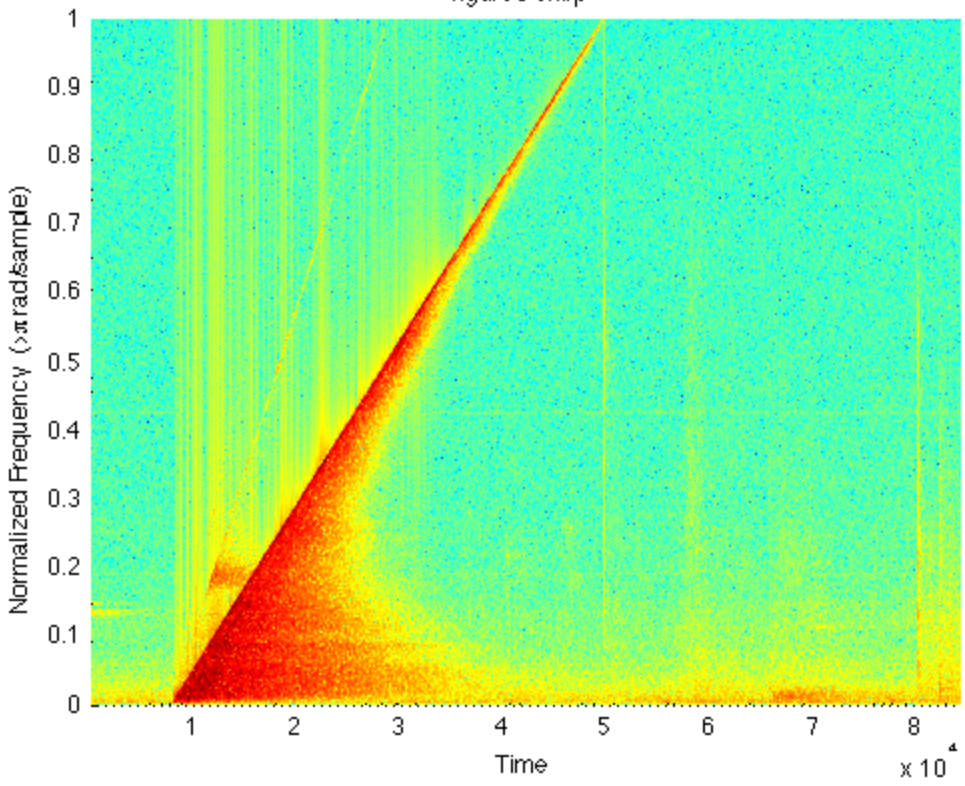
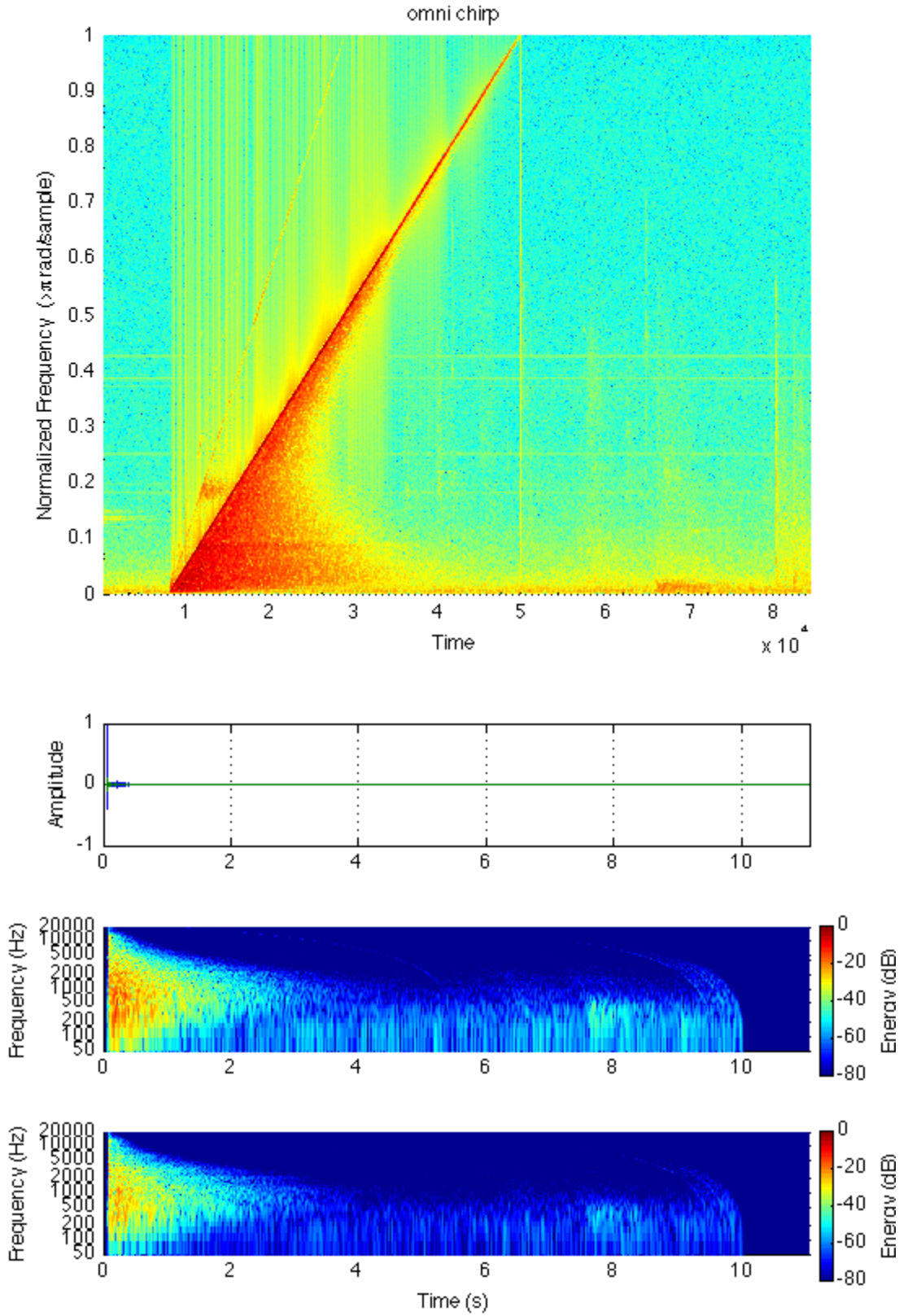


figure8 chirp




omni chirp
Normalized Frequency (×π rad/sample)
Time
x 10^4
Amplitude
Frequency (Hz)
Energy (dB)
Time (s)