```
clc; clear all; close all;

load ('handel');

%% Plot the original signal
S = y';
t = (1:length(S))/Fs;

% figure(1)
% plot(t, S);
% xlabel('Time [sec]');
% ylabel('Amplitude');
% title('Signal of Interest (Original)');

% p8 = audioplayer(v, Fs);
% playblocking(p8);

%% Setup the time and frequeuncy domain
n = length(t) - 1;
t = t(1 : n);
S = S(1 : n);
L = t(n);
k = 2*pi/L*[0:(n/2 - 1), (-n/2):-1];
ks = fftshift(k);

%% FFT the signal
St = fft(S);

%% Plot the signal in frequency and time domain
figure(2)
plot(t, S);
xlabel('Time [s]');
ylabel('S(t)');
title('Signal of Interest (Time Domain)');

figure(3)
plot(ks, fftshift(abs(St)));
xlabel('Frequency [\omega]');
ylabel('FFT(S(t))');
title('Signal of Interest (Frequency Domain)');

%% Compute spectrograms using Gaussian window with
different widths
a = [10, 1, 0.1];
```

```
tslide = linspace(0, t(n), 25); % t0 = 0.3719
Sgt_spec = cell(1, length(a));


for i = 1 : length(a)
    for j = 1 : length(tslide)
        g = exp(-a(i)*(t - tslide(j)).^2);
        Sg = g.*S;
        Sgt = fft(Sg);
        Sgt_spec{i} = [Sgt_spec{i}; fftshift(abs(Sgt))];
        figure(4)
        if j == 12
            subplot(3, 1, i);
            plot(t, S);
            hold on;
            plot(t, g, 'r');
            xlabel('Time [s]');
            ylabel('S(t), g(t)');
            title(['Window width a = ' num2str(a(i))]);
        end
    end
    figure(i + 4)
    pcolor(tslide, ks, Sgt_spec{i}.');
    xlabel('Time [s]');
    ylabel('Frequency [\omega]');
    title(['Spectrogram of the signal for window width a =
' num2str(a(i))]);
    shading interp;
    colormap(hot);
end



%% Compute spectrograms using Gaussian window with
undersampling size
tslideUS = linspace(0, t(n), 5);    % t0 = 2.2312
SgUSt_spec = cell(1, length(a));

for i = 1 : length(a)
    for j = 1 : length(tslideUS)
        gUS = exp(-a(i)*(t - tslideUS(j)).^2);
        SgUS = gUS.*S;
        SgUSt = fft(SgUS);
        SgUSt_spec{i} = [SgUSt_spec{i};
fftshift(abs(SgUSt))];
    end
    figure(i + 7)
    pcolor(tslideUS, ks, SgUSt_spec{i}.');
    xlabel('Time [s]');
    ylabel('Frequency [\omega]');
    title(['Spectrogram of the signal for window width a =
' num2str(a(i)) ' (undersampled)']);
    shading interp;
    colormap(hot);
end

%% Compute spectrogram using square, Haar & ramp wavelet
Sst_spec = [];
Sht_spec = [];
Srt_spec = [];

for j = 1 : length(tslide)
    s = zeros(1, length(t));
    h = zeros(1, length(t));

    for i = 1 : length(t)
        tau = tslide(j);
        low = tau - a(2)/2;
        high = tau + a(2)/2;
        if t(i) < low || t(i) >= high
            s(i) = 0;
        else
            s(i) = 1;
        end

        if t(i) >= low && t(i) < tau
            h(i) = 1;
        elseif t(i) >= tau && t(i) < high
            h(i) = -1;
        else
            h(i) = 0;
        end
    end

    r = conv(s, h, 'same');
    r = r/max(r);
    if j == 12
        figure(11)
        plot(t, S);
        hold on;
        plot(t, s, 'r');
        xlabel('Time [s]');
        ylabel('S(t), h(t)');

        figure(12)
        plot(t, S);
        hold on;
        plot(t, h, 'r');
        xlabel('Time [s]');
        ylabel('S(t), h(t)');

        figure(13)
        plot(t, S);
        hold on;
        plot(t, r, 'r');
        xlabel('Time [s]');
        ylabel('S(t), r(t)*h(t)');

    end

    Ss = s.*S;
    Sst = fft(Ss);
    Sst_spec = [Sst_spec; fftshift(abs(Sst))];

    Sh = h.*S;
    Sht = fft(Sh);
    Sht_spec = [Sht_spec; fftshift(abs(Sht))];

    Sr = r.*S;
    Srt = fft(Sr);
    Srt_spec = [Srt_spec; fftshift(abs(Srt))];
end

figure(14)
pcolor(tslide, ks, Sst_spec.');
xlabel('Time [s]');
ylabel('Frequency [\omega]');
title('Spectrogram of the signal using square wavelet');
shading interp;
colormap(hot);
```

```
figure(15)
pcolor(tslide, ks, Sht_spec.');
xlabel('Time [s]');
ylabel('Frequency [\omega]');
title('Spectrogram of the signal using Haar wavelet');
shading interp;
colormap(hot);

figure(16)
pcolor(tslide, ks, Srt_spec.');
xlabel('Time [s]');
ylabel('Frequency [\omega]');
title('Spectrogram of the signal using triangle wavelet');
shading interp;
colormap(hot);

%% Compute spectrogram using Mexican hat wavelet
Smt_spec = [];

for j = 1 : length(tslide)
    tau = tslide(j);
    m = (1 - (t - tau).^2).*exp(-(t - tau).^2./2);

    figure(17)
    if j == 12
        plot(t, S);
        hold on;
        plot(t, m, 'r');
        xlabel('Time [s]');
        ylabel('S(t), h(t)');
    end

    Sm = m.*S;
    Smt = fft(Sm);
    Smt_spec = [Smt_spec; fftshift(abs(Smt))];
end

figure(18)
pcolor(tslide, ks, Smt_spec.');
xlabel('Time [s]');
ylabel('Frequency [\omega]');
title('Spectrogram of the signal using Mexican hat
wavelet');
```

```
shading interp;
colormap(hot);
```