Spectrum estimation using a recorded signal

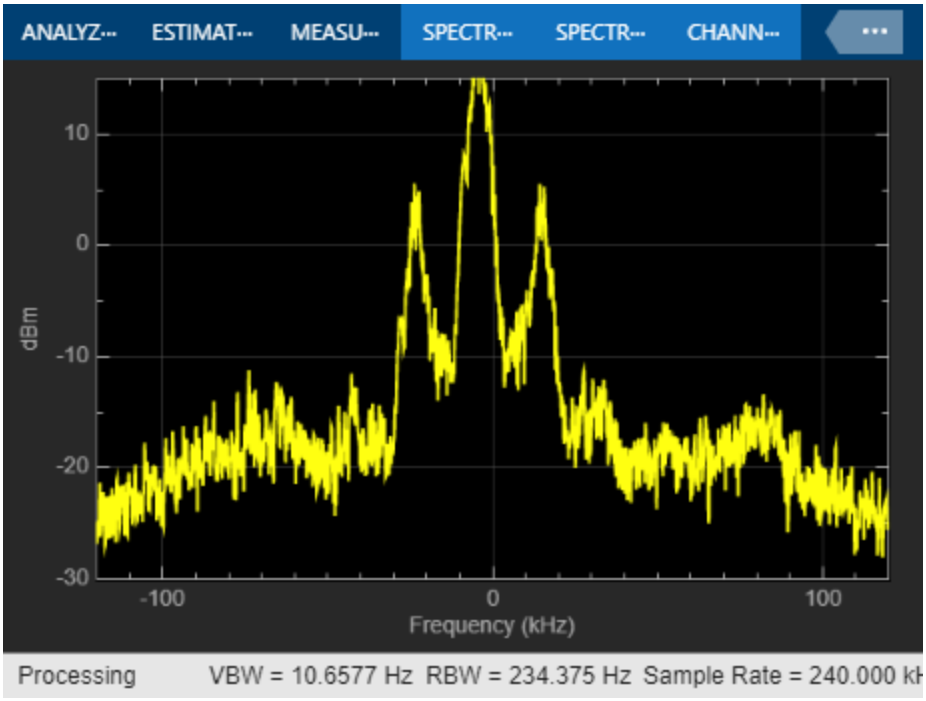
```
% The functions whose names start with 'rw_' are the ones you need to
% design yourself. Without these function the script does not work.
% Matlab's spectrum analyzer works, though.
%
% By R.W.

% Ensure that some previous versions of the script don't mess up the
% performance.
clear all, close all

% The script simply defines the input arguments for rw_welch, own spectrum
% analyzer
[nSection, nOverlap, wintype, nFFT] = rw_ini_spectrum();

% Read the .mat file. Contains the parameters rxSigStore, nFrame, nSample,
% FESR, expFreq. The last one is the center frequency of the received signal,
 not
% needed here
load spectrum_signal
% Averaging of consecutive SpectrumAnalyzer plots
SpecAve = 1;
% Depends on the gain
Ylims = [-30,15];
% Parameters are left bottom, width height
% Figposition gnerates the position irrespective of the screen resolution
% If your Matlab version does not have the function, just comment it out
% here and in spectrum analyzer objects
Scopepos = figposition([50 10 30 40]);
Figpos = figposition([50, 50, 30, 40]);

% Spectrum analysis object. dsp.SpectrumAnalyzer will be removed so using
% the substitute. The parameter ForgettingFactor controls the averaging.
% 0 = no memory, no overaging. 1 = full-memory. You can tune the parameter
% to improve the match with your own function
hSpectrumAnalyzer = spectrumAnalyzer(...
    'FrequencySpan',     'Full', ...
    'SampleRate',        FESR, ...
    'YLimits',           Ylims,...
    'SpectralAverages', SpecAve, ...
    'AveragingMethod', 'exponential',...
    'ForgettingFactor', 0.7,...
    'FrequencySpan',     'Start and stop frequencies', ...
    'StartFrequency',    -FESR/2, ...
    'StopFrequency',     FESR/2,...
    'SpectrumType', 'Power',...
    'Method', 'welch',...
    'Position',    Scopepos);


% Own spectrum analyzes object
```

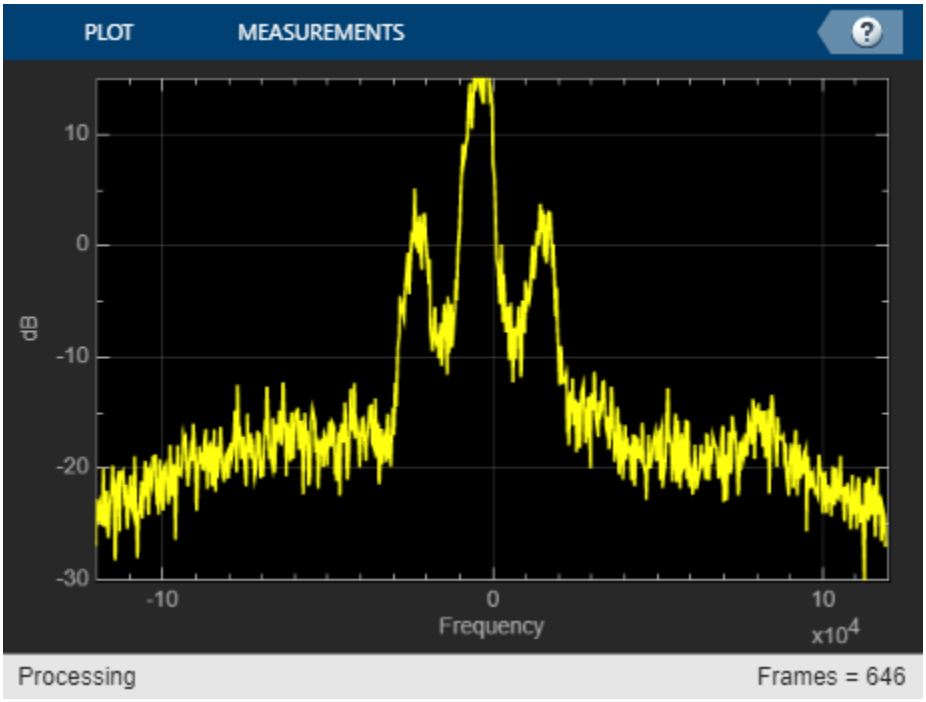
```
hOwnSpec = dsp.ArrayPlot(...
    'Name', 'Own Spectrum',...
    'XDataMode',  'Custom',...
    'CustomXData', linspace(-FESR/2,FESR/2,nFFT),...
    'YLimits', Ylims,...
    'XLabel', 'Frequency',...
    'YLabel', 'dB',...
   'PlotType', 'Line',...
   'Position', Figpos);
```

Stream Processing

```
% Clock time duration of the signal
fprintf('Receive time %f [s]   \n', nSample/FESR*nFrame)

for iFrame = 1 : nFrame
    % Read nSample samples from the signal store
    rxSig = rxSigStore((iFrame-1)*nSample+1:iFrame*nSample);
    rxSig = rxSig - mean(rxSig);  % Remove DC component

    % Display received frequency spectrum
    hSpectrumAnalyzer(rxSig);

    % Own spectrum analyzer. Parameters:
    % Input signal
    % Length of the section (=M in the slides)
    % Overlap between sections, (0...1)
    % Type of the window, see wvtool()
    % Length of the FFT
    % Output is the pediodogram in linear scale
    perio = rw_welch(rxSig,nSection,nOverlap,wintype,nFFT);
    hOwnSpec(10*log10(perio));

    % Slow down the loop to match the real signal
    % Assuming the execution time of the loop is negligible
    pause(nSample/FESR)
end
```

```
Receive time 17.066667 [s]
```





Published with MATLAB® R2023a