J-DSP Ion Channel Exercises

These exercises are aimed at creating a basic understanding of using J-DSP for Ion channel Conductance Fluctuation analysis. Please direct your browser to this link http://jdsp.asu.edu/JDSP_Ion/jdsp.html to load the ion-channel version of the J-DSP software.

Getting Started with J-DSP:

For using J-DSP, latest Java runtime engine is required on your computer, which can be downloaded and installed from http://java.com/en/download/manual.jsp. Once the ***Editor Frame*** opens, all the functions of J-DSP/ESE can be seen as buttons in the top and left ***function panels*** of the applet. To use a particular function, just click on the button, move the cursor to required area in the white ***canvas*** and click once again. To modify the parameters of the block, double-click to open a ***dialog window*** for the block, set the required parameters and click *Update*. *Help* button can be used to open a separate help window that displays help on that particular block. A screenshot of the editor frame, along with ***Long Signal Generator*** block and its dialog ***window*** are displayed in Figure 1. The other functional blocks of JDSP will be described as they are introduced in the exercises.

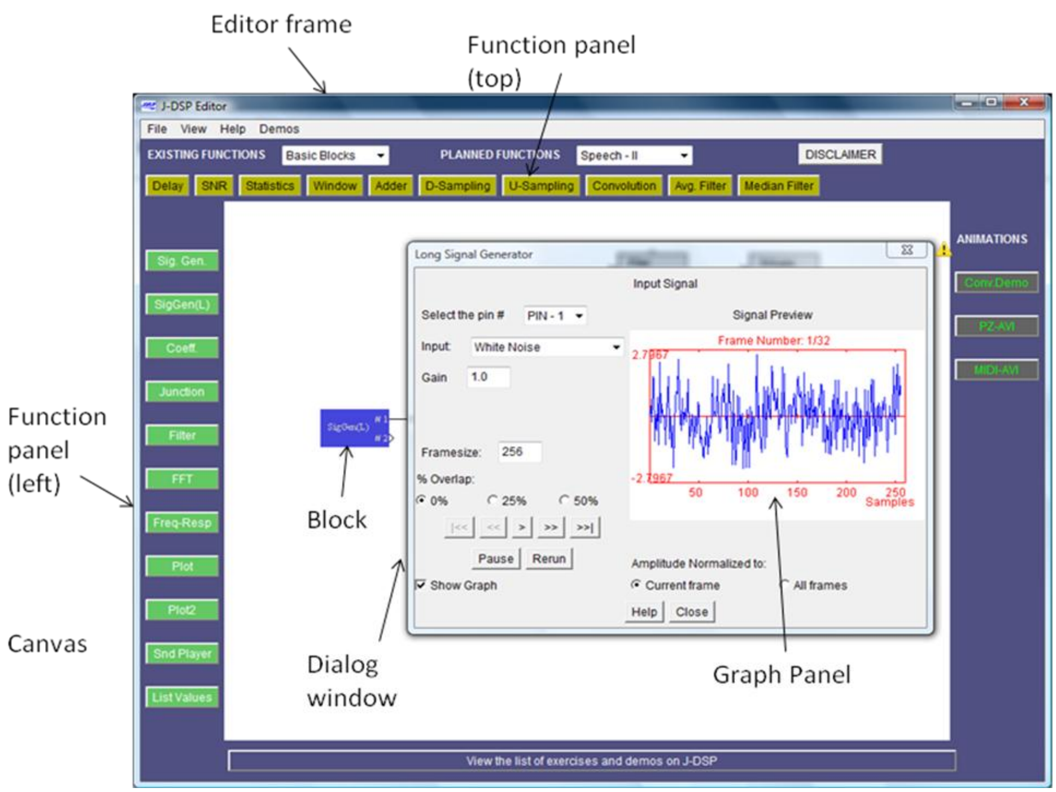


Figure 1: J-DSP editor frame with the dialog window for *Long Signal Generator* block.

Signal Characterization by Transform Domain Analysis

The objective of this exercise is to perform transform domain analysis of synthetic data of fluctuations in conductance of two different channels with J-DSP.

Ion-channel signals arise from fluctuations in the conductance of cell membranes. The Long signal generator block in J-DSP has time series data obtained two different channels. The statistical characteristics of these fluctuations of two channels reflect the corresponding physical processes.

Exercise 1: Statistics of Fluctuations

Place the Long Signal Generator and the Plot blocks and establish connections as shown in figure below.

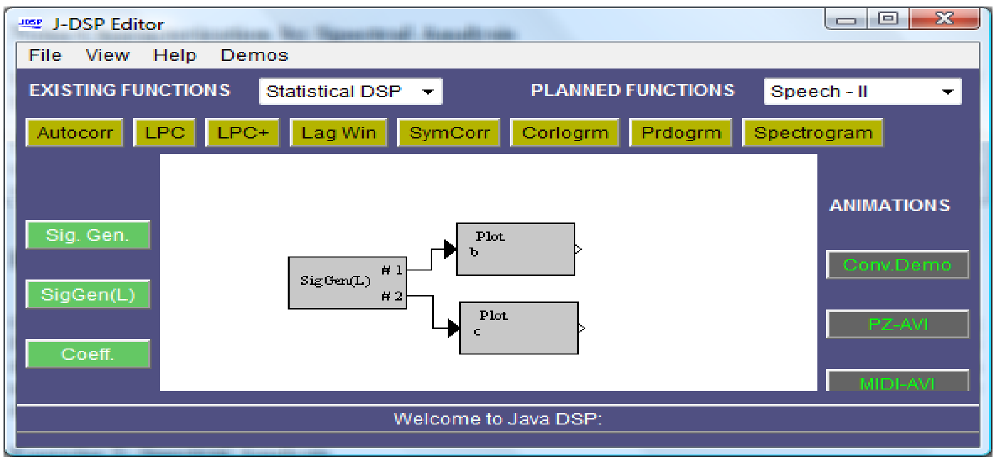


Figure 2: Long Signal Generator Block with Plot Blocks.

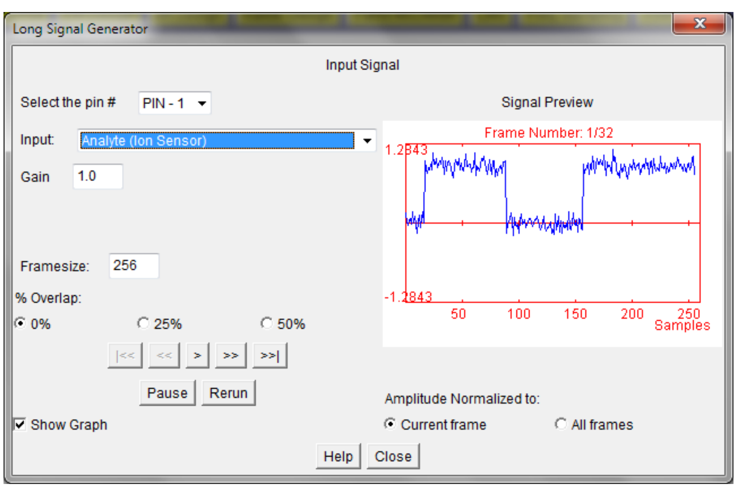


Figure 3: Long Signal Generator Dialog Box

Choose ***Analyte (Ion Sensor)*** as the input option of the ***Long Signal Generator*** and pick a frame that does not contain jumps in signal amplitude. Plot the time series with the ***Plot***. Using *Graph/Value/Stats* display the signal statistics.

Pay attention to the variance of the time series and make a note of it. This is the noise variance.

Repeat the same for ***No Analyte (Ion Sensor).*** (Use the Pin 2 for this).

[1]The data is obtained using the QuB simulation Software for two mode single channel are sampled at 20 kHz (0.05 ms).

<u>Exercise 2</u>: De-nosing using low pass filtering

We will use low pass filtering in this exercise to de-noise the ion-channels signals to obtain better separation between the states of the signals.

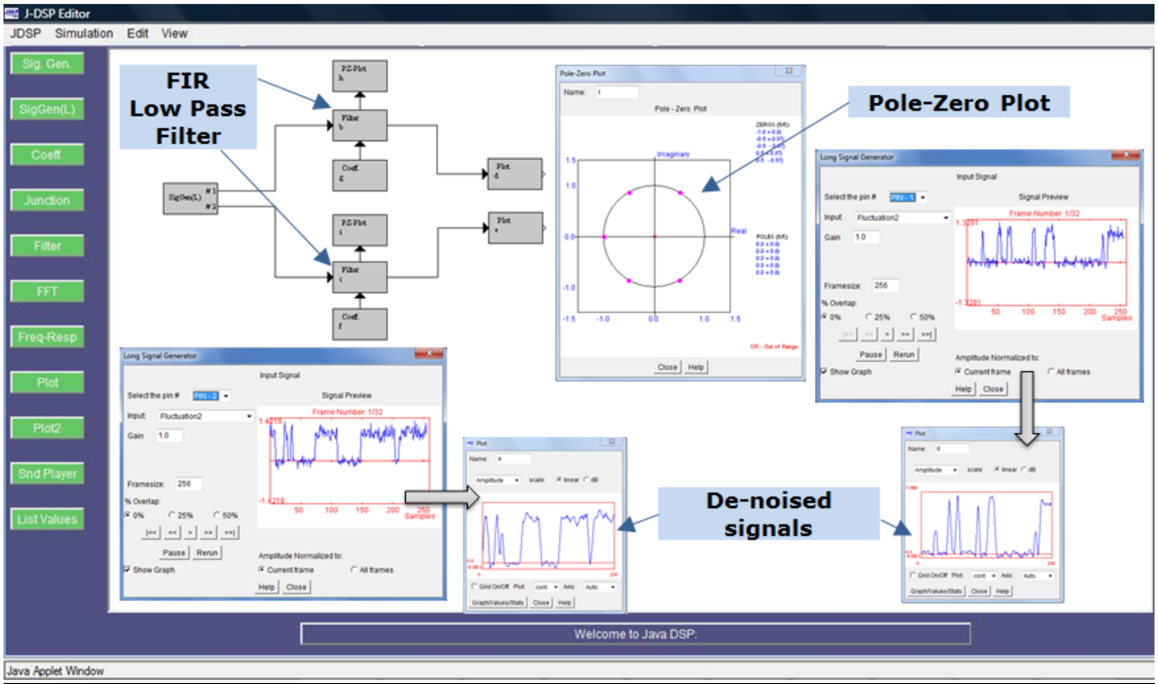


Figure 4: -DSP block diagram to perform de-noising of two the ion-channel signals in J-DSP. A low pass de-noising filter is created using Filter block. The coefficients are adjusted using a Coeff block or a graphical Pole-Zero placement block.

The following blocks are used in this exercise:

1. ***Filter:*** This block performs the filtering operation based on the coefficients.
2. ***Coeff:*** The coefficients for the filter are entered in the dialog window of this block.
3. ***PZPlacement:*** This block provides a graphical method to design the filter.
4. ***PZPlot:*** A block to visualize the poles and zeros of a filter.
5. ***FreqResp:*** This block shows the magnitude and phase response of the filter.

The low pass filter to de-noise the ion-channel signals can be created using the ***Filter*** block. The poles and zeros can be placed using ***PZPlacement*** block. Alternatively, set the coefficients of the filter using ***Coeff*** block. The frequency response can be observed using ***FreqResp*** block.

To see the separation in the state amplitudes use the ***Hist*** block available in ***Basic blocks*** section. Compare the separation distance of the states in the original and de-niosed signals.

<u>Exercise 2</u>: Autocorrelation Function

The autocorrelation function specifies over what time periods the noise is correlated. If the fluctuations are slowly varying at a particular time the chance of predicting right at a later time is high and vice versa.

Connect the ***Autocorrelation*** Block to two outputs of ***Long Signal Generator*** Block. Set the two outputs of ***Long Signal Generator*** Block to ***Analyte*** and ***No Analyte*** using Pin 1 and Pin 2 as before. ***Plot*** can be connected to the ***Autocorrelation*** to plot the values. Using *Graph/Value/Stats* display the signal statistics.

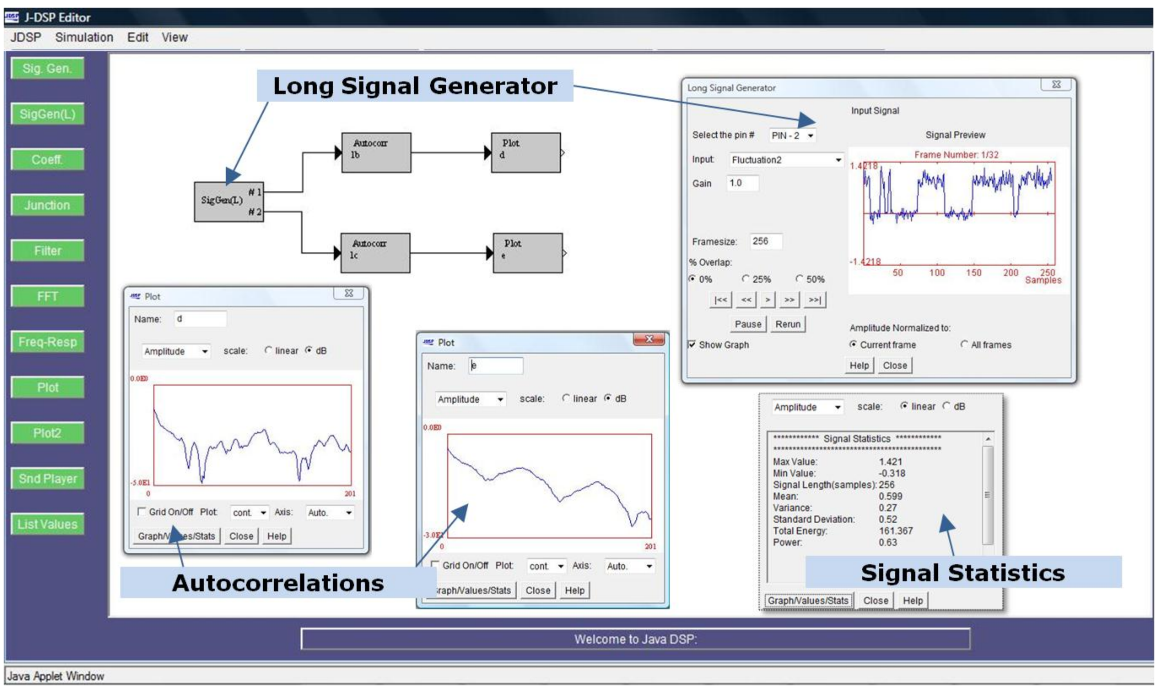


Figure 5: Block Connections for finding plotting Autocorrelations.

Simple Molecular Mechanisms give rise to exponential covariance functions:

$$C(T) = var e^{-\left(\frac{T}{\tau}\right)}$$

Notice that T=0, the Autocorrelation function is simply the time average squared amplitude of the fluctuations.

<u>Exercise 3</u>: Transform Domain Feature Extraction

A slowly varying signal will contain lower frequency component than a rapidly varying signal.

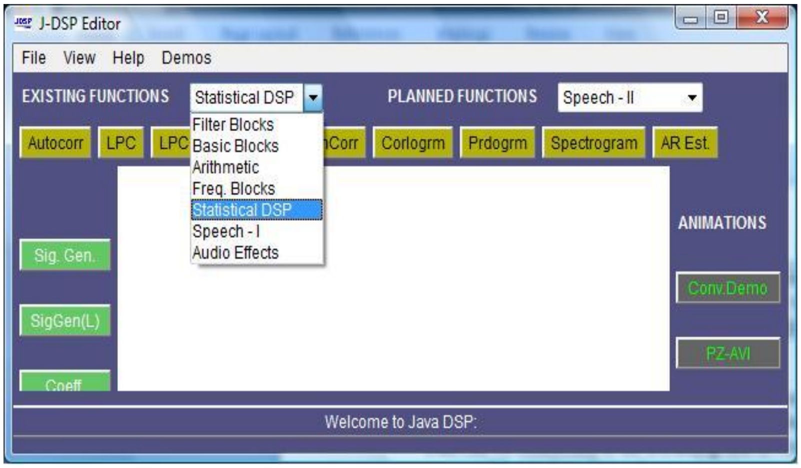


Figure 4: The Periodogram block can be found in class of functions called Statistical DSP

The next step is to find the power spectrum of each signal using ***Periodogram***. The ***Periodogram*** block can be found in class of functions called ***Statistical DSP***. Connect the ***Long Signal Generator*** and the ***Peridogram*** blocks. Set the periodogram type as *Welch*, and the window should be set as *Rectangular*. Use *Value* to display the signal statistics.

Just as exponential functions arise in physical relaxation process, so does a particular spectral form called the simple spectrum, whose equation is given below:

$$S(f) = \frac{2var/\pi fo}{1 + (f/fo)^2}$$

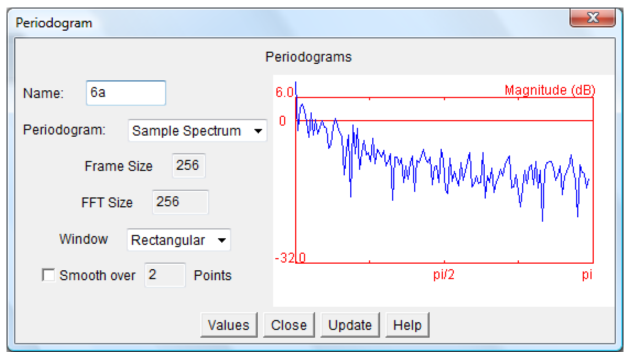
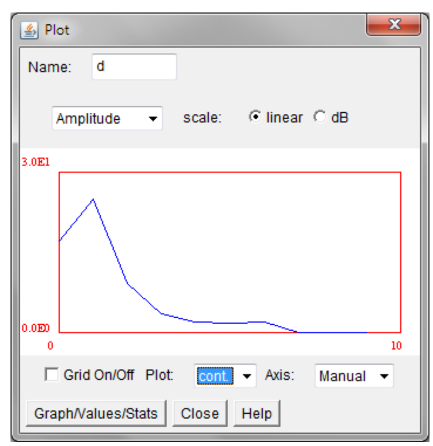


Figure 5: Peridogram block Dialog Box

Observe that as you browse through frames the Periodogram changes drastically.

Power Spectral Density (PSD) Features

We can build a unique feature for a signal by summing the power in the dyadic bins across frames. Use the ***Dyadic Bin*** block to perform this. This block can be found in Basic blocks section. Adjust axes to see the plot clearly.



The **corner frequency** f_0 is the frequency bin at which the power starts dropping. The opening and closing time can be approximated by the corner frequency.

Observe that the feature values now are stable across frames.

The same can be repeated using Walsh-Hadamard transform by replacing ***Periodogram*** block with ***WHT*** block.