iJDSP User Manual

Table of Contents

Introduction

The iJDSP environment is a standalone application capable of running without an internet connection on mobile devices running iOS. It is compatible with devices running iOS 3.2 or a later version. The application consists of multiple layers of views, whose architecture is illustrated in the figure below:

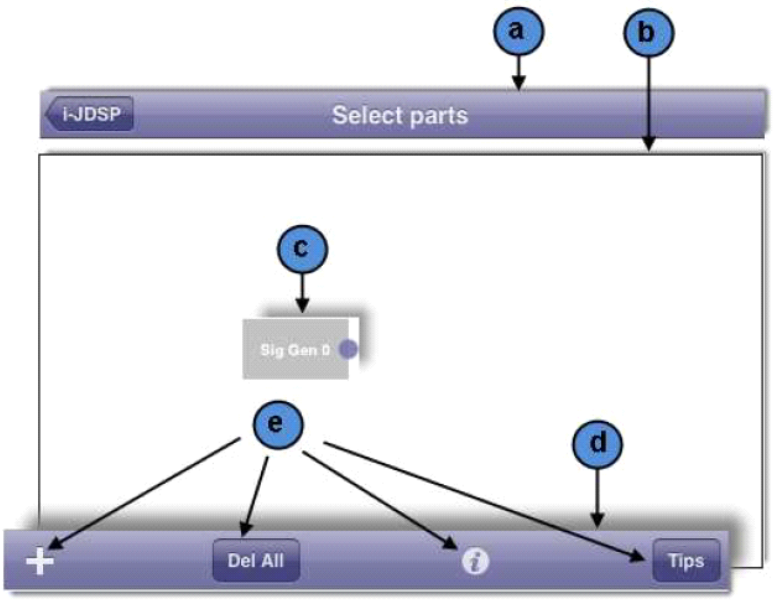


Fig. Architecture of views in iJDSP : (a) Navigation bar (hidden in main view) for accessing multiple view controllers; (b) Main canvas; (c) DSP block diagrams; (d) Toolbar for displaying controls; (e) Utility panel buttons.

The navigation bar is the view of the object UINavigationController, which is the control managing the hierarchy of views and navigating across the views. The pin configuration of a general block in the iJDSP environment is shown below. A block may contain one to six pins. The pins 0, 1 and 2 serve as input pins from the parent blocks and the rest of the pins serve as output pins to be sent to child blocks.

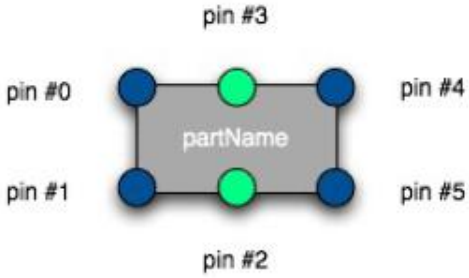


Fig. The view of a block in the main canvas.

When the block needs only one input, pin 0 alone is used. If two inputs are needed, then pins 0 and 1 are used. Similarly, pin 4 alone is used if only one output is given by a block, and pins 4 and 5 are used in case of the presence of two outputs. The pin configuration of a block is fixed and cannot be changed by the user. The type of signal to be passed as input to a block (viz. time domain or frequency domain) is also fixed and cannot be changed by the user. Examples of some blocks with different number of pins are shown below.

Fig. : Views of a few commonly used blocks, illustrating the variability in the usage of pins by blocks.

The main canvas supports gestures such as double taps and drags for making connections and selecting or deleting functional blocks. A list of such gestures supported by the canvas is tabulated in Table below:

Gesture	**Operation**
Double tap on a block	Open a block to see its property dialogue
Long Hold on a block	Delete the block
Single tap on a pin	Make a connection between blocks
Single tap on a connection	Delete a connection between blocks
Hold and drag on a block	Move blocks with your fingers
Swipe down/up on the main canvas	Hide/show the bottom bar

Signal Generator

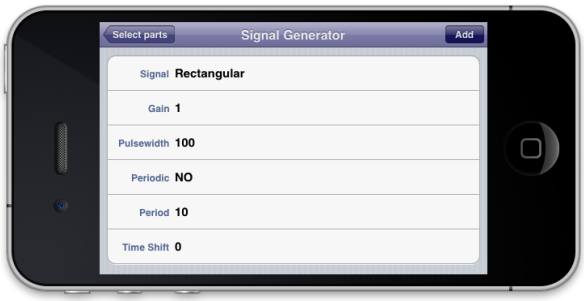
Function Description:

Generates a variety of discrete time-domain signals. It supports: pulses, triangular, delta, exponential, sinusoid, sinc, random, and user-defined signals. The pulse width of each signal and the amplitude of the signal ("gain") can be set. A signal can be made periodic if the "periodic" option is selected. The base of the exponential can also be varied. Random signals can have uniform, normal, and Rayleigh distributions with variable mean and variance.

Pin	Signal	Domain
P4	*x(n)*	Time
Pin Description		

Block Parameters:

- Signal - Possible types: Rectangular, triangular, delta, exponential, sinusoid, sinc, random (uniform, Rayleigh, Gaussian) and user defined.
- Gain – Any valid double precision number.
- Pulse width – from 0 to 256 samples.
- Periodic – YES or NO
- Period – 0 to 256.
- Time shift – 0 to 256.



Parameter Selection Menu

Signal Selection Menu

Junction

Function Description:

This block propagates its input signal at its two outputs. The input signal can be either time-domain, frequency-domain, or filter coefficients. The ***Junction*** block essentially allows other blocks to share the same signal or parameters.

Pin	Signal	Domain
P0	*Input*	Time
P4	*Output = Input*	Time
P5	*Output = Input*	Time
Pin Description		

Block Parameters:

- None.

Filter

Function Description:

This block filters the input signal based on the provided numerator and denominator coefficients and the standard difference equation. The filter coefficients must be provided using the ***Coeff.*** block. An option is provided to start with zero initial conditions or non-zero initial conditions.

Pin	Signal	Domain
P0	*Input*	Time
P4	*Output*	Time
P2	*Filter Coeffs*	N/A
P3	*Output*	Frequency
Pin Description		

Block Parameters:

- None.

Equations implemented:

$$y(n) = \sum_{i=0}^{L} b_i x(n-i) - \sum_{j=1}^{M} a_j y(n-j)$$

where

$x(n) = input\ signal$

$y(k) = output\ signal$

Filter Coefficients (Filter Coeff)

Function Description:

This block allows the user to enter filter coefficients. A maximum of 11 coefficients can be used. Coefficients can be entered in "tabular" form or "by line" form as shown below. The "by line" option provides an easy way to 'cut' and 'paste' coefficients from other sources.

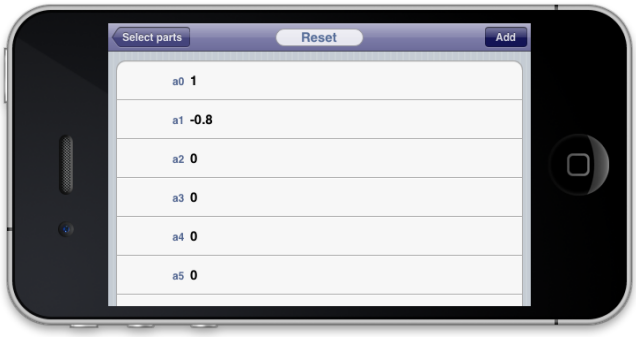
Pin	Signal	Domain
P3	*Filter Coeffs*	N/A
Pin Description		

Block Parameters:

Filter coefficients, namely a_i's and b_i's in the equation

$$y(n) = \sum_{i=0}^{L} b_i x(n-i) - \sum_{j=1}^{M} a_j y(n-j)$$

.



Parameter Editor

Plot

Function Description:

This block primarily plots the signal at its input in an x-y axis coordinate system. It can also display values in text form and calculate some basic signal statistics. The magnitude, magnitude squared, real part, imaginary part, and phase of the input signal can be examined.

Pin	Signal	Domain
P0	*Input signal*	Time
P4	*Input signal*	Time
Pin Description		

Block Parameters:

Plot Choices – Magnitude plot, Phase plot, real part plot, imaginary part plot.

Values of the signal sample can be chosen from the navigation bar as well and some sample plots of the magnitude of an exponential signal are shown below:

Plot Options

Parameter Editor

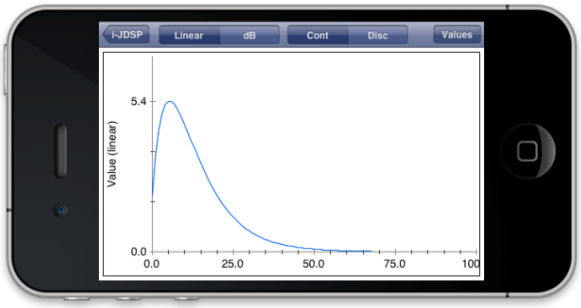


Fig: Continuous plot with the "linear" option.

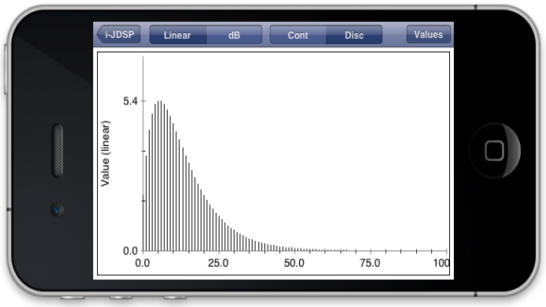


Fig: Discrete plot with the "linear" option.

Frequency Response (Freq Resp)

Function Description:

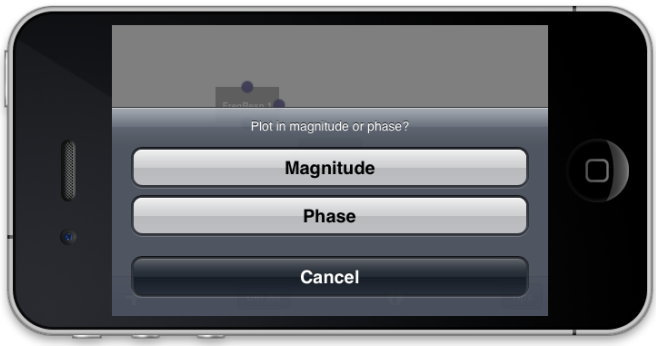
This block calculates and displays the frequency response of a filter. It can be connected to any block that can generate filter coefficients. In its dialog window, the top plot displays the magnitude in dB or linear scale and the bottom plot shows the phase.

Pin	Signal	Domain
P2	*Filter Coeffs*	N/A
P3	*Filter's output signal*	Frequency
P4	*Signal*	Frequency
Pin Description		

Block Parameters:

Plot Choices – Magnitude plot, Phase plot.

Some sample plots of the magnitude of an exponential signal are shown below:



Frequency Response Plot Options

The values of the frequency response can be looked at by clicking the "Values" button in the navigation bar on the top of the screen.

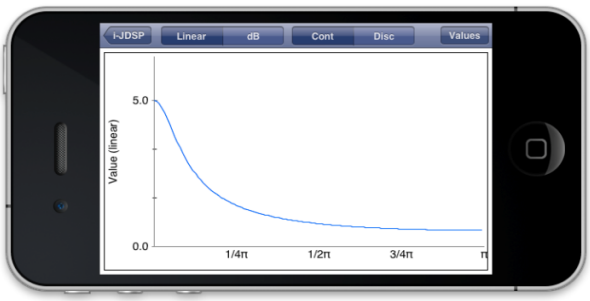


Fig: Continuous plot with the "linear" option.

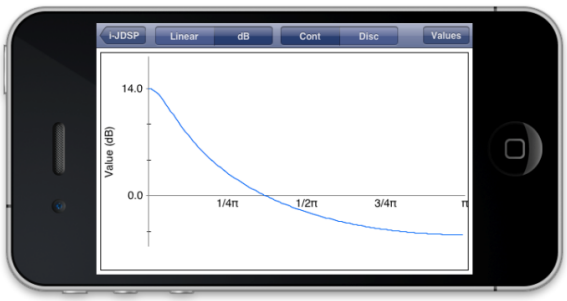


Fig: Continuous plot with the "dB" option.

Adder

Function Description:

This block adds two input signals and gives the sum of the same signals as the output.

Pin	Signal	Domain
P0	$x_1(n)$	Time
P1	$x_2(n)$	Time
P4	$y(n)$	Time
Pin Description		

Block Parameters:

Plot Choices - '+' or '-' for the two input signals.

Equations implemented:

$$y(n) = x_1(n) + x_2(n)$$

where

$x_1(n) = input\ signal$

$x_2(n) = input\ signal$

$y(n) = output\ signal$

Adder Choices

Window

Function Description:

This block performs a windowing operation on the input signal.

Pin	Signal	Domain
P0	*x(n)*	Time
P4	*y(n)*	Time
Pin Description		

Block Parameters:

- Window type – Hamming, Hanning, rectangular, Bartlett, Blackman, and Kaiser.
- Length – 0 to 256 samples.

Equations implemented:

$$y(n) = x(n).w(n)$$

where

$$x(n) = input\ signal$$

$$w(n) = window$$

$$y(n) = output\ signal$$

Parameter Editor

FFT(Fast Fourier Transform)

Function Description:

This implements the Cooley-Tukey Fast Fourier Transform algorithm. The algorithm converts time domain signals into frequency domain representation.

Pin	Signal	Domain
P0	*x(n)*	Time
P4	*X(k)*	Frequency
Pin Description		

Block Parameters:

- FFT length - Possible lengths are 8, 16, 32, 64, 128 , 256.

Equations implemented:

$$X(k) = \sum_{n=0}^{N-1} x(n) e^{-j2\pi kn/N}$$

where

$x(n) = input\ signal$

$X(k) = output\ signal$

$N = FFT\ length$

Parameter Editor

Pole-Zero Plot (PZ Plot)

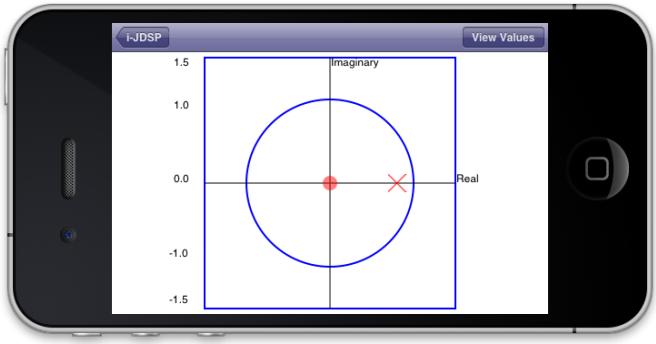
Function Description:

This block calculates and displays the poles and zeros of a transfer function in the z-plane. The block accepts filter coefficients at its input.

Pin	Signal	Domain
P2	*Filter Coeffs*	N/A
Pin Description		

Block Parameters:

- None.



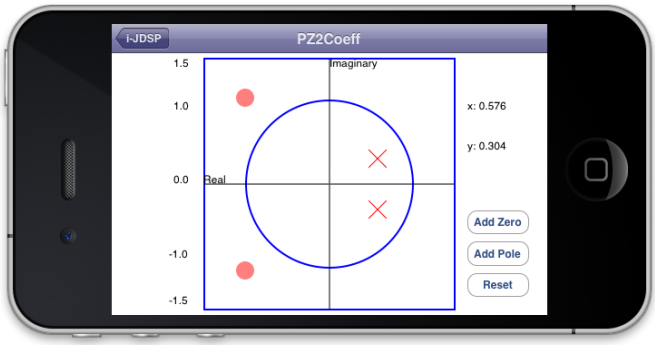
PZ Plot

Pole-Zero to Filter Coefficients (PZ2Coeff)

Function Description:

This block uses poles and zeros graphically placed by the user to generate appropriate filter coefficients to generate the required frequency response.

Pin	Signal	Domain
P3	*Filter Coeffs*	N/A
Pin Description		



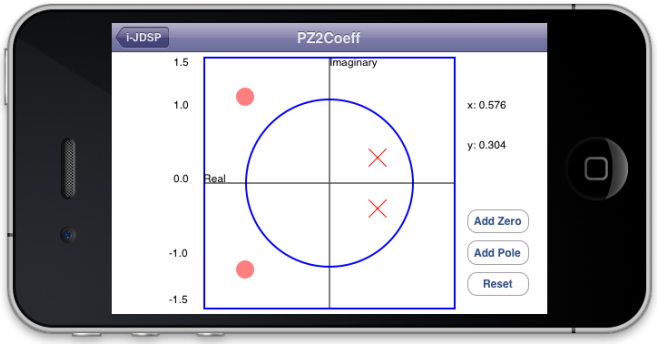
PZ placement Plot

Pole-Zero Placement (PZ Placement)

Function Description:

This block allows the user to enter poles and zeros representing a filter. The corresponding filter coefficients are passed to the output. Poles and zeros are added as conjugate pairs, and no more than 10 (5 pairs) can be entered. They can be placed graphically on the plot. Graphical manipulation of poles and zeros is achieved through buttons that allow placing, moving and resetting to clear the poles.

Pin	Signal	Domain
P3	*Filter Coeffs*	N/A
Pin Description		



PZ placement Plot

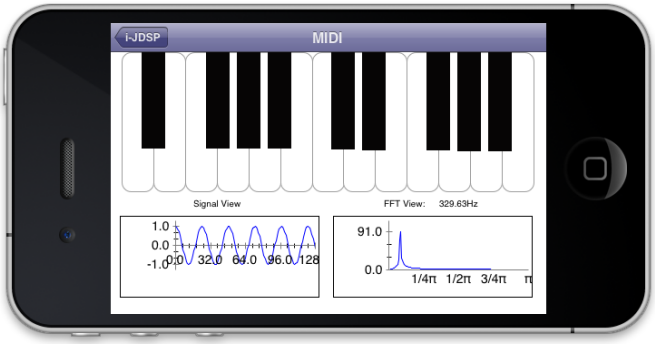
MIDI

Function Description:

Simulates a piano keyboard and generates Musical Instrument Digital Interface (MIDI) sounds at the frequencies described by the MIDI standard. The MIDI block can generate a single tone of length: 256 (1 frame), 1280 (5 frames) and 8192 (32 frames) samples. It can also generate a sequence of pre-recorded tones. The sampling frequency is 8KHz. All the tones can be used in a DSP simulation and are audible using the iJDSP provided sound player.

Pin	Signal	Domain
P4	*Audio output*	Time
Pin Description		

Shown in this display of the view of the MIDI block is the outcome of pressing a single key. The left hand plot pane shows the time domain output of the MIDI player and the right hand plot pane shows the frequency spectrum of the MIDI block's output.



MIDI View

DTMF Tones

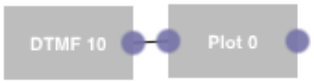
Function Description:

This block generates Dual-Tone-Multi-Frequency (DTMF) tones used in landline telephony applications. This block generates a single tone of length: 256 (1 frame), 1280 (5 frames) and 8192 (32 frames) samples. It also generates a sequence of pre-recorded tones. The sampling frequency is 8KHz. The tones can be played back using the iJDSP provided sound player, and used in a DSP simulation.

Pin	Signal	Domain
P4	*Audio output*	Time
Pin Description		

MIDI View

The sample block diagram for viewing the output of the DTMF block in a plot is given below, along with a display of the plot. The key 1 has been pressed in this example in the DTMF block before displaying the plots.



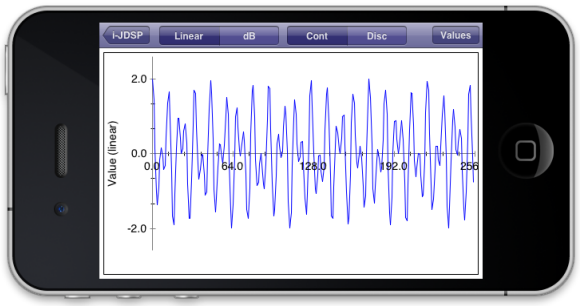


Fig. : Time domain plot of the DTMF output.

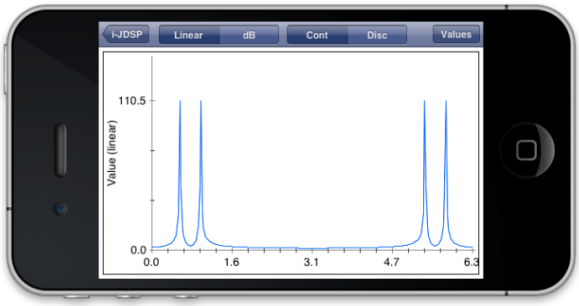


Fig. : Frequency spectrum of the DTMF output.

Sound Recorder

Function Description:

This block provides for real-time recording of a sound signal of up to ten seconds. It can then parse the data into multiple frames of size 256 samples each. The output of this block can be used by blocks such as FFT or Filter. However, the output signal of this block cannot be plot because it has multiple frames but can be replayed using the Sound Player function block.

Pin	Signal	Domain
P4	*Audio output*	Time
Pin Description		

Block Parameters:

- None.

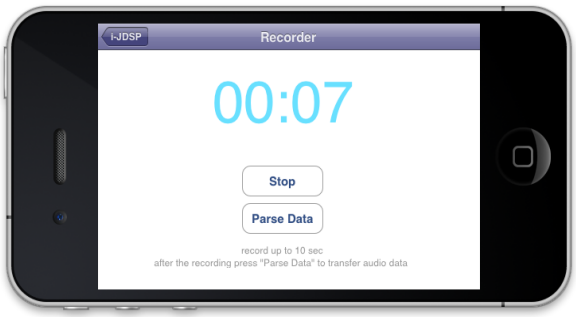


Fig. : View to record an input sound signal

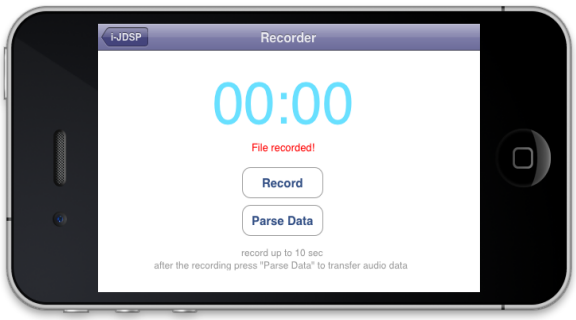


Fig. : View after a sound signal has been recorded

Sound Player

Function Description:

This block can be used to play an input sound signal on a frame-by-frame basis.

Pin	Signal	Domain
P4	*Audio input*	Time
Pin Description		

Block Parameters:

None.

Frequency Response Demo (Freq Resp Demo)

Function Description:

The Frequency Response Demo block shows samples of three pre-configured filters, namely: low-pass filter, high-pass filter and band-pass filter, to provide a basic understanding of filters and its effect on the corresponding signal output.

Block Parameters:

- None.

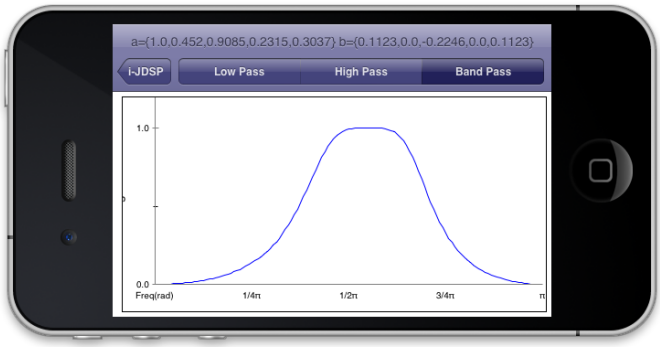


Fig. : Band Pass filter response.

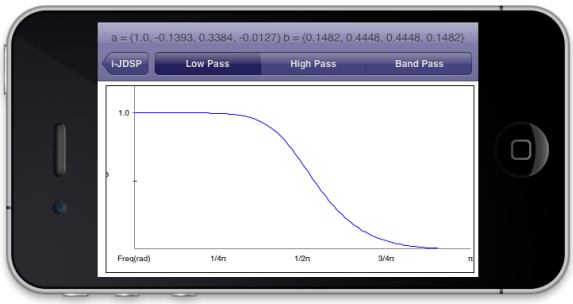


Fig. : Low Pass filter response.

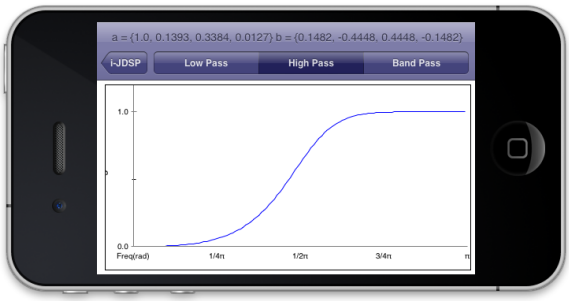


Fig. : High Pass filter response.

Convolution

Function Description:

This block performs a convolution operation between its input signals.

Pin	Signal	Domain
P0	$x_1(n)$	Time
P1	$x_2(n)$	Time
P4	$y(n)$	Time
Pin Description		

Block Parameters:

- None.

Equations Implemented:

$$y(n) = x_1(n) * x_2(n), \qquad y(n) = \sum_{m=0}^{N-1} x_1(m)\, x_2(n-m)$$

$x_1(n) = input\ signal$
$x_2(n) = input\ signal$
$y(n) = output\ signal$

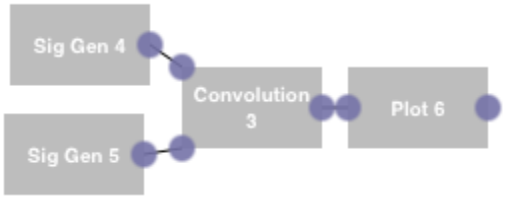


Fig. : Simulation block diagram for convolution.

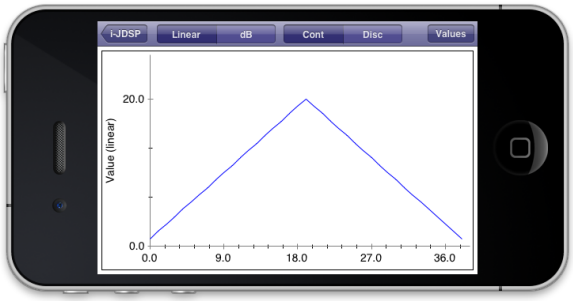


Fig. : Output plot of the resulting convolved signal for two rectangular signal inputs.

Convolution Demo (Conv Demo)

Function Description:

The Convolution Demo block shows a graphical visualization of the convolution process involving two signals for both continuous-time and discrete-time signals. Other than providing a set of pre-defined sequences, discrete convolution allows users create their own signals by holding and dragging each sample to the desired amplitude.

Block Parameters:

- The parameters that can be chosen vary for both the Discrete-time and Continuous-time signal options.
- Discrete-time Signals – The signal type can be selected to be Rectangular, Triangular, Delta, Exponential or Sinc.
- Continuous-time Signals – The causality of the signal can be selected to be either Causal or Non-causal apart from the signal type as listed above.

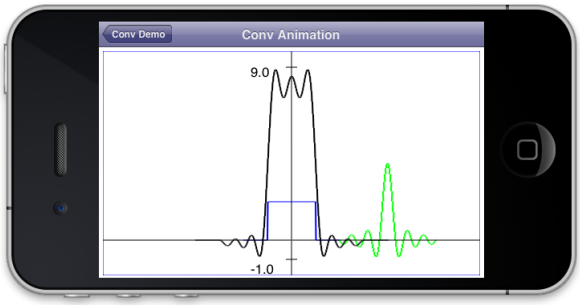


Fig. : Continuous-time convolution demo.

Fig. : Continuous-time convolution parameters.

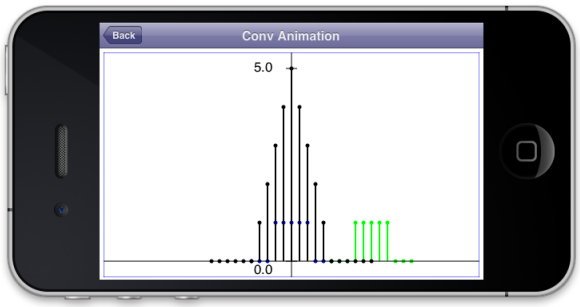


Fig. : Discrete-time convolution demo.

Fig. : Discrete-time convolution parameters.

Kaiser

Function Description:

This block designs Kaiser FIR filters based on the windowing method. The design process involves calculating the Fourier series of the ideal filter and then multiplying it with a Kaiser window that best fits the filter specifications.

Pin	Signal	Domain
P3	*Filter Coeffs*	N/A
Pin Description		

Block Parameters:

- Filter type - can be low-pass, high-pass, stop-band or pass-band.
- Wp_1, Ws_1 – pass-band and stop-band edge cut-off frequencies respectively.
- Wp_2, Ws_2 – second pass-band and stop-band edge cut-off frequencies respectively (for pass-band filters).
- PB, SB – pass-band and stop-band tolerances in dB.

Equations Implemented:

$$N = \frac{A-8}{2.285\Delta\omega} \; and \; \beta = \begin{cases} 0.1102(A-8.7) & , \quad A > 50 \\ 0.5842(A-21)^{0.4} + 0.07886(A-21) & , \quad 21 \le A \le 50 \\ 0 & , \quad A < 21 \end{cases}$$

$\Delta\omega$ is the transition band of the filter and A is equal to the smaller of PB and SB.

Fig. : Kaiser filter block parameters.

Finite Impulse Response Design (FIR Design)

Function Description:

Designs a finite impulse response (FIR) filter based on the windowing method. The windowing FIR filter design method is a straightforward technique implemented by expanding the frequency response of an ideal filter in a Fourier series and then truncating and smoothing the response using a window.

Pin	Signal	Domain
P3	*Filter Coeffs*	N/A
Pin Description		

Block Parameters:

- Window type - Hamming, Hanning, Blackman, Bartlett, rectangular or Kaiser
- Filter order - (maximum is 64)
- Type - low-pass, high-pass, pass-band, or stop-band.
- Cut-off frequencies (*fc)* - take values from 0 to 1, where *fc = 1* corresponds to half-the-sampling frequency.

Fig. : FIR filter design block parameters.

Parks-McClellan

Function Description:

This block designs FIR filters using the Parks-McClellan algorithm with min-max design.

Pin	Signal	Domain
P3	*Filter Coeffs*	N/A
Pin Description		

Block Parameters:

- Filter type - can be low-pass, high-pass, stop-band, or pass-band.
- Wp_1, Ws_1 – pass-band and stop-band edge cut-off frequencies respectively.
- Wp_2, Ws_2 – second pass-band and stop-band edge cut-off frequencies respectively (for pass-band filters)
- PB, SB – pass-band and stop-band tolerances in dB.

Fig. : Parks-Mc-Clellan filter parameters.

Infinite-impulse Response (IIR Design)

Function Description:

Designs an infinite (length) impulse response (IIR) filter based on the bilinear transformation. Butterworth, Chebyshev -I & -II, and Elliptic filters are supported.

Pin	Signal	Domain
P3	*Filter Coeffs*	N/A
Pin Description		

Block Parameters:

- Filter type - can be low-pass, high-pass or pass-band
- Wp_1, Ws_1 – pass-band and stop-band edge cut-off frequencies respectively
- Wp_2, Ws_2 – second pass-band and stop-band edge cut-off frequencies respectively (for pass-band filters)
- *PB*, *SB* – pass-band and stop-band tolerances in dB. Cut-off frequencies (f_c) – take values from 0 to 1, where $f_c = 1$ corresponds to half-the-sampling frequency.

Fig. : IIR filter design parameters.

Up Sampling

Function Description:

Up-samples the input signal by an integer factor *L*.

Pin	Signal	Domain
P0	*x(n)*	Time
P4	*y(n)*	Time
Pin Description		

Block Parameters:

- Sampling rate (L) – *L* is allowed to take values from 1 to 10.

Equations Implemented:

$$y(n) = x(n / L)$$

$x(n) = input\ signal$
$y(n) = output\ signal$
$L = sampling\ rate$

Fig. : Up Sampling block parameters.

Down Sampling

Function Description:

Down–samples the input signal by an integer factor M.

Pin	Signal	Domain
P0	*x(n)*	Time
P4	*y(n)*	Time
Pin Description		

Block Parameters:

- Sampling rate (M) – M is allowed to take values from 1 to 10.

Equations Implemented:

$$y(n) = x(nM)$$

$x(n) = input\ signal$
$y(n) = output\ signal$
$M = sampling\ rate$

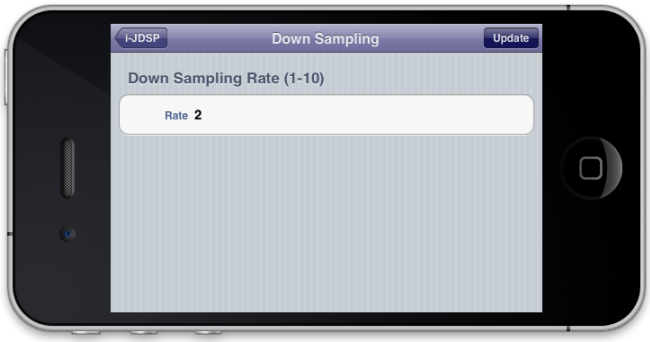


Fig. : Down Sampling block parameters.

Inverse Fast Fourier Transform (IFFT)

Function Description:

Implements the Inverse Fast Fourier Transform algorithm.

Pin	Signal	Domain
P0	*X(k)*	Frequency
P4	*x(n)*	Time
Pin Description		

Block Parameters:

- Inverse FFT size - 8, 16, 32, 64, 128, or 256

Equations Implemented:

$$x(n) = \frac{1}{N}\sum_{k=0}^{N-1} X(k)e^{j2\pi kn/N}, \qquad n = 0 \ldots N-1$$

$x(n) = output\ signal$
$X(k) = input\ signal$

Fig. : IFFT block parameters.

Peak Picking

Function Description:

Selects a specific number of peaks from a frequency-domain signal. The first set of peaks or the highest magnitude ones can be selected.

Pin	Signal	Domain
P0	$X(k)$	Frequency
P4	$X_c(k)$	Frequency
Pin Description		

Block Parameters:
Provides two different options for selecting the peaks, namely:

- First - selects the first 64 peaks of the input signal.
- Max – selects the largest 64 peaks in magnitude.

Fig. : Peak Picking block parameters.

Signal to Noise Ratio (SNR)

Function Description:

This block calculates the signal-to-noise ratio (SNR) value in 'dB' between two signals. The reference signal is given as input to the upper input pin and the processed signal is given as input to the lower pin.

Pin	Signal	Domain
P0	$s(n)$	Time
P1	$s'(n)$	Time
Pin Description		

Block Parameters:

- None.

Fig. : SNR block parameters.