ANDROID APPLICATIONS USERS' GUIDE

(Off Line)

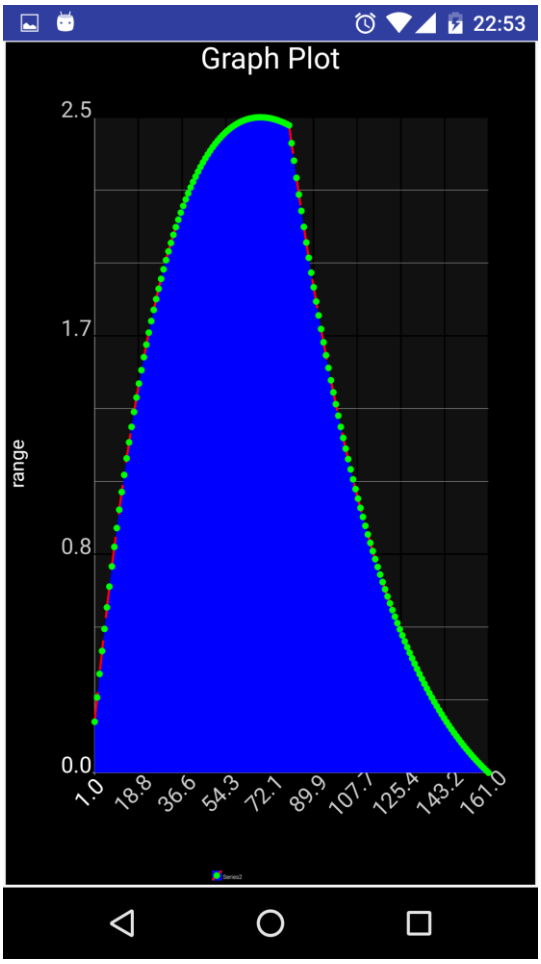
The document contains instructions for the user, to run the digital signal processing applications implemented on android smartphone. There are instructions for various applications which implement DSP concepts like convolution and its properties, FFT, modulation, demodulation, DTMF encoder. Also Echo cancellation, Doppler Effect, Music Synthesis application can be seen. Name of applications are written in the block letters, followed by the instructions to properly run the app on the android smartphone. Android smartphone used to develop was NEXUS 5 with latest operating system, android studio for IDE and MATHCODER to generate C programming files from MATLAB code, MATLAB version number: R2015b to run the MATLAB. Android plot website can be used as guide to understand the way graphs are implemented in all the applications. Results are displayed on home screen when you run the app, to verify the correctness of the solution. In the coming paragraphs, we shall consider each application at a time, read the instructions carefully and follow them to run the app on target, also plot the graph of the result obtained. Enjoy Signal Processing!!

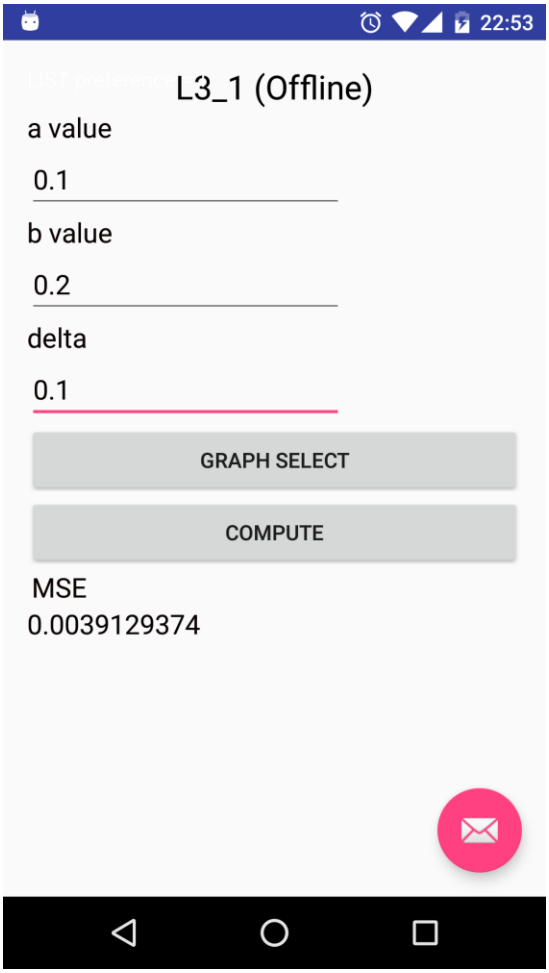
L3_1 (Offline)

This app implements convolution between the two exponent signals, analytically and using command, mean square error is recorded between the two methods. Also user can select the signal to plot. Follow the points given below to run the app.

1. 'a value' option lets user to select the exponent of the first signal (e^-a) and generate the signal.
2. 'b value' for exponent of second signal (e^-b).
3. 'delta' value will determine the size the input signal and impulse response, which affects the size of the output also.
4. 'GRAPH SELECT' button, when pressed four options pop up, 1. Output(y), convolution computed using command, 2. Output (y_ac), analytical convolution result 3. Input signal 4. Impulse response.
5. 'COMPUTE', when pressed convolution is carried out.
6. 'MSE': Displays the mean square error.

Typical values of ‘a’ would from 0.1 to 0.9, so of ‘b’. delta can be given from 0.001 to 8, if delta value is more than 8, it will chops off value to 8 and performs the operation. Below are snapshot of the application. Figure one shows the graph output using convolution command. Second figure shows the home screen of the app, where user can input ‘a’, ‘b’, ‘delta’ values and also see the MSE output.



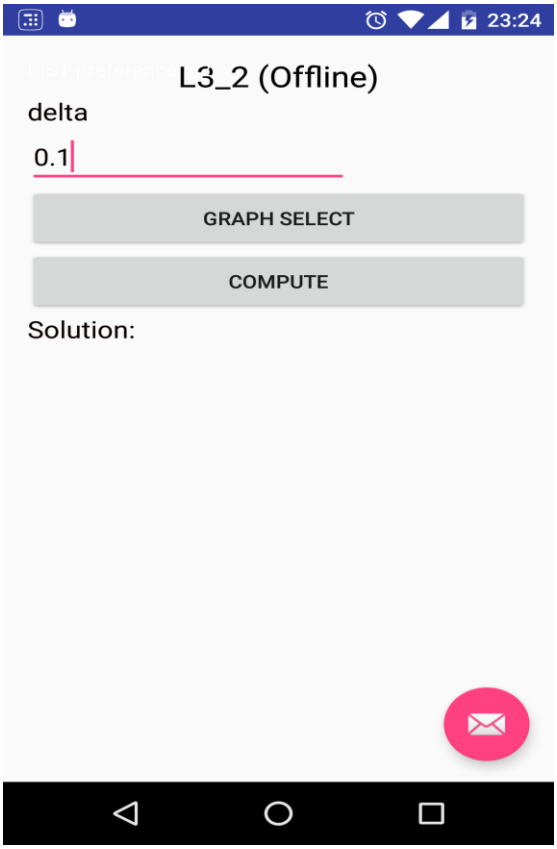


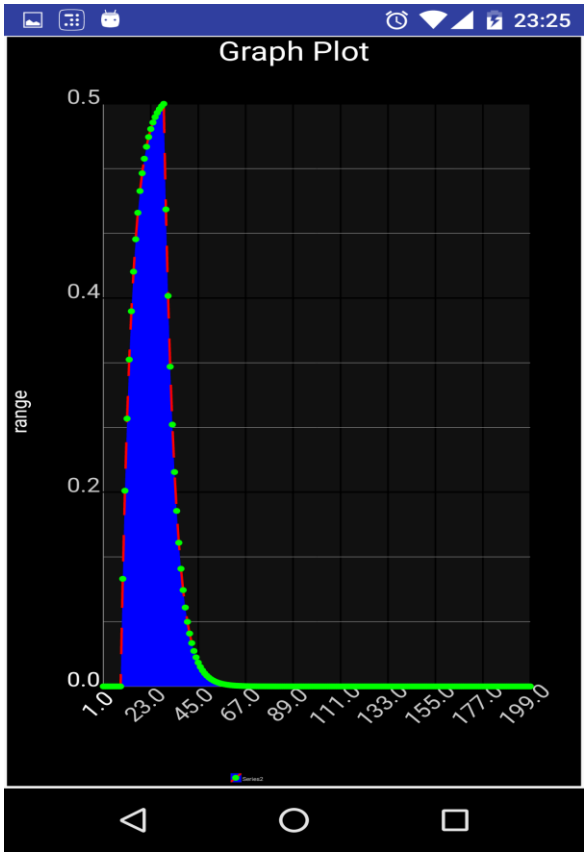
L3_2 (Offline)

This app implements convolution between the input signal and impulse response, user is allowed to select the delta value, which determines the length of the input signal and impulse response. User can also select the signal to plot. Follow the points given below to run the app.

1. ‘delta’ value will determine the size the input signal and impulse response, which affects the size of the output also.
2. ‘GRAPH SELECT’ button, when pressed 3 options pop up, 1. Output(y), convolution output, 2. Input 3. Impulse response.
3. ‘COMPUTE’, when pressed convolution is carried out.
4. ‘Solution’: Displays values of the output, which can be used to verify the solution of the problem.

Typical values of ‘delta’ can be given from 0.001 to 8, if delta value is more than 8, it will chops off value to 8 and performs the operation. Below are some snapshot of the app. Figure 1. Home Screen. Figure 2. Output Graph.



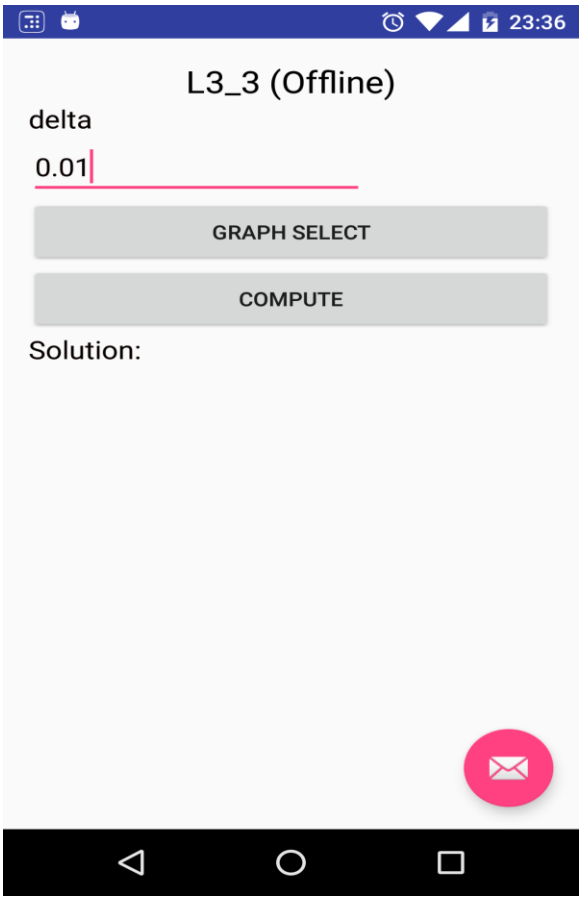


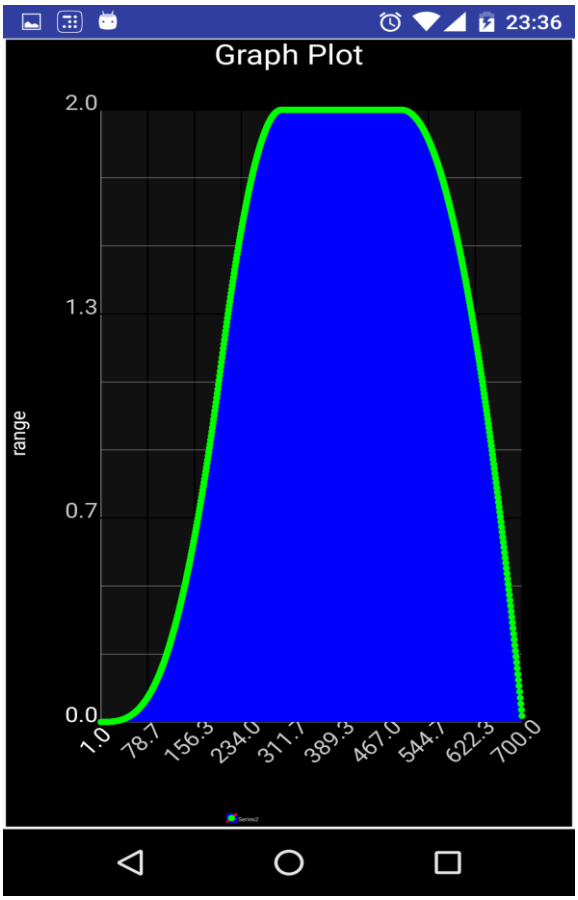
L3_3 (Offline)

This app implements convolution between the two input signals, user is allowed to select the delta value, which determines the length of the input signals. User can also select the signal to plot. Follow the points given below to run the app.

1. ‘delta’ value will determine the size the both input signals, which affects the size of the output also.
2. ‘GRAPH SELECT’ button, when pressed 3 options pop up, 1. Output(y), convolution output, 2. Input_1 3. Input_2.
3. ‘COMPUTE’, when pressed convolution is carried out.
4. ‘Solution’: Displays values of the output, which can be used to verify the solution of the problem.

Typical values of ‘delta’ can be given from 0.001 to 8, if delta value is more than 8, it will chops off value to 8 and performs the operation. Below are some snapshot of the app. Figure 1. Home Screen. Figure 2. Output Graph.



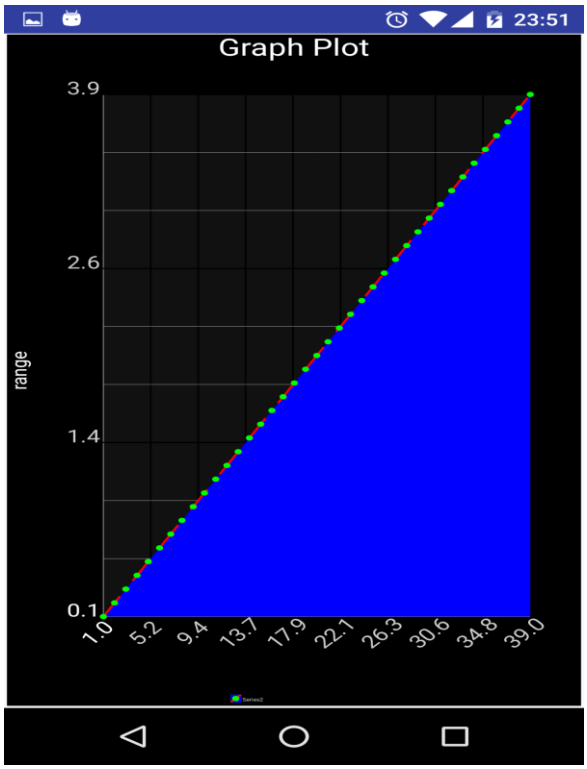


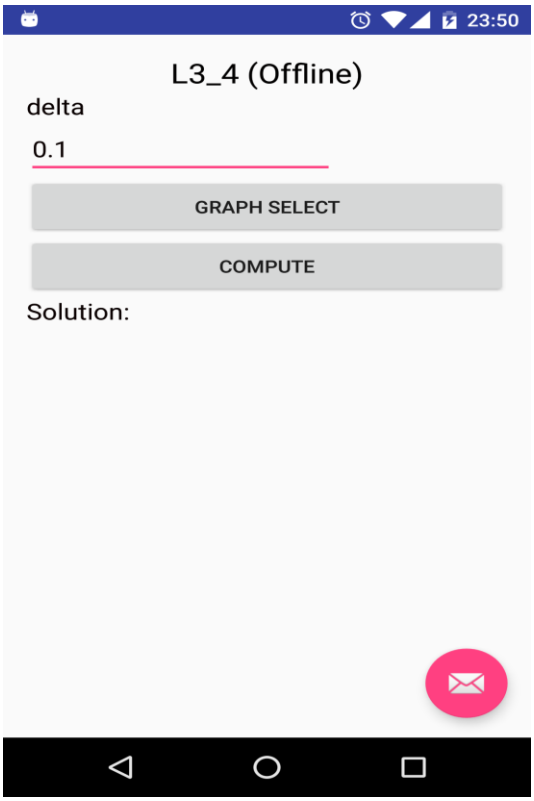
L3_4 (Offline)

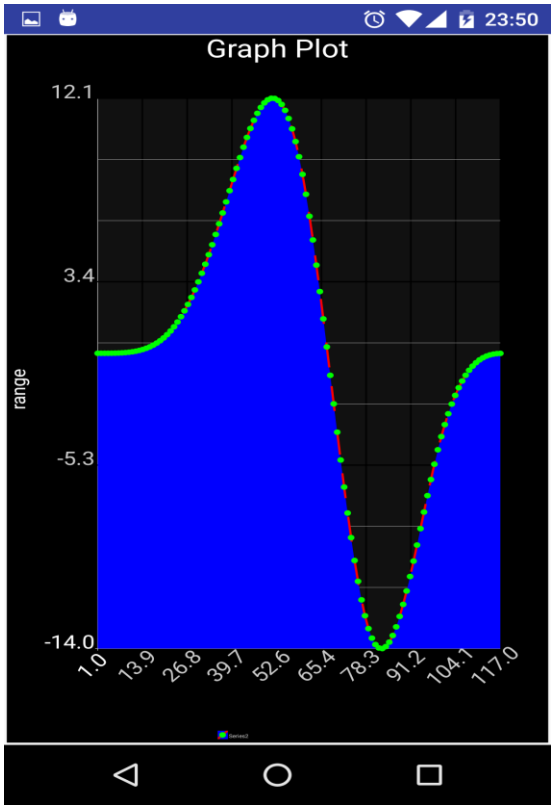
This app implements various properties of convolution between the input signal and two impulse response, user is allowed to select the delta value, which determines the length of the signals. User can also select the signal to plot. Follow the points given below to run the app.

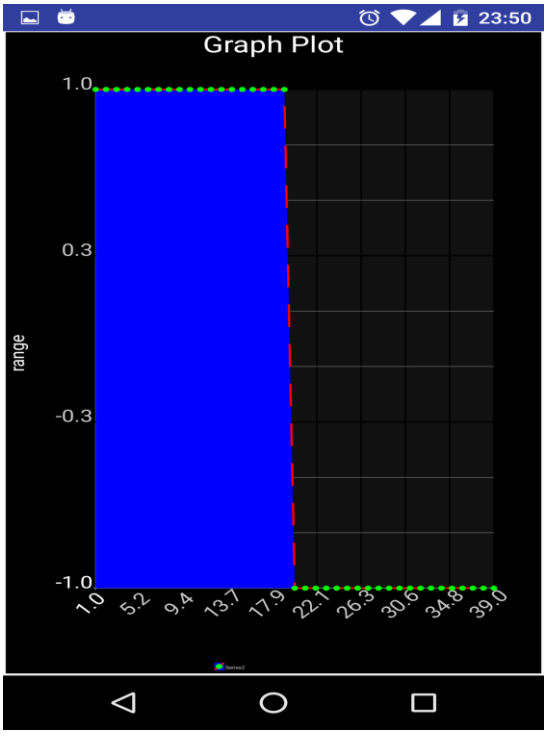
1. ‘delta’ value will determine the size the signals, which affects the size of the output also.
2. ‘GRAPH SELECT’ button, when pressed 9 options pop up, 1. 6 different outputs (y), calculated each in a different way, 2. Input 3. Impulse response_1 and 4. Impulse response_2.
3. ‘COMPUTE’, when pressed convolution is carried out.
4. ‘Solution’: Displays values of the output, which can be used to verify the solution of the problem.

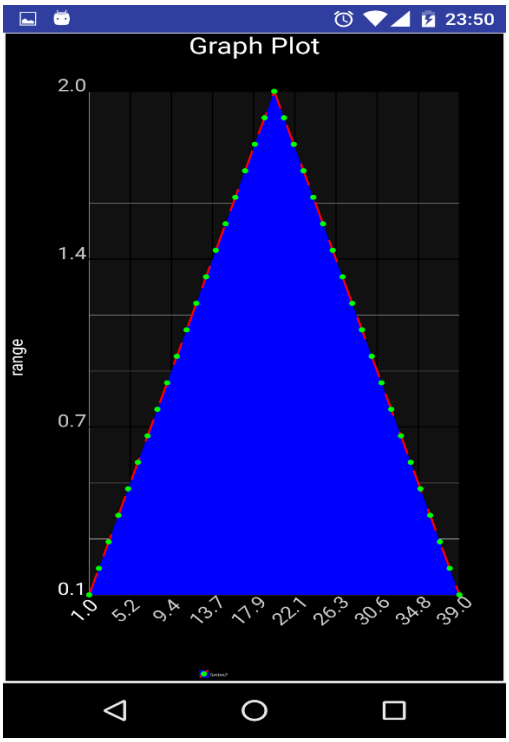
Typical values of ‘delta’ can be given from 0.001 to 8, if delta value is more than 8, it will chops off value to 8 and performs the operation. Below are some snapshot of the app. Figure 1. H2 2. Home Screen. Figure 3. Output Graph. Figure 4. Input Figure 5. H1




23:50
L3_4 (Offline)
delta
0.1
GRAPH SELECT
COMPUTE
Solution:


23:50
Graph Plot
12.1
3.4
-5.3
-14.0
range
1.0
13.9
26.8
39.7
52.6
65.4
78.3
91.2
104.1
117.0


23:50
Graph Plot
1.0
0.3
-0.3
-1.0
range
1.0
5.2
9.4
13.7
17.9
22.1
26.3
30.6
34.8
39.0

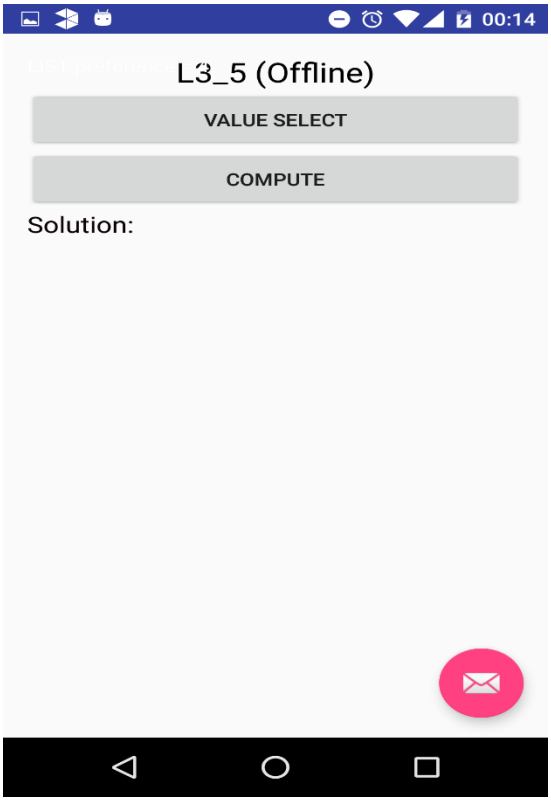

23:50
Graph Plot
2.0
1.4
0.7
0.1
range
1.0
5.2
9.4
13.7
17.9
22.1
26.3
30.6
34.8
39.0

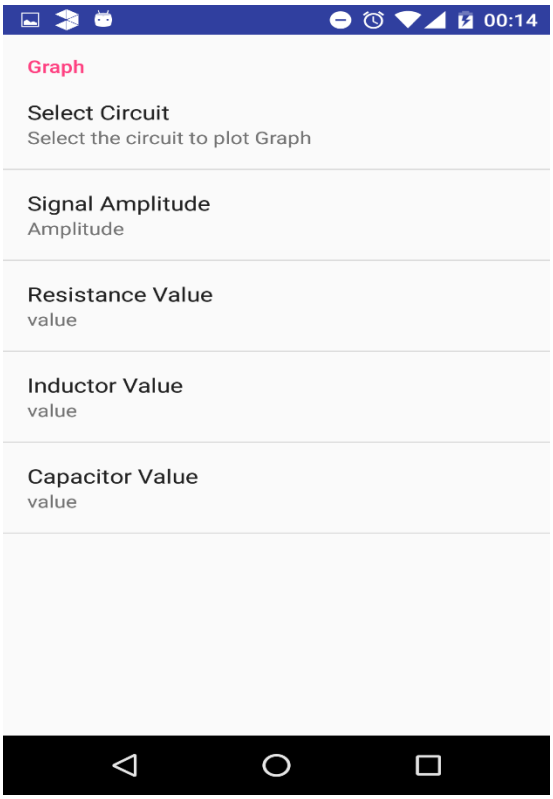
L3_5 (Offline)

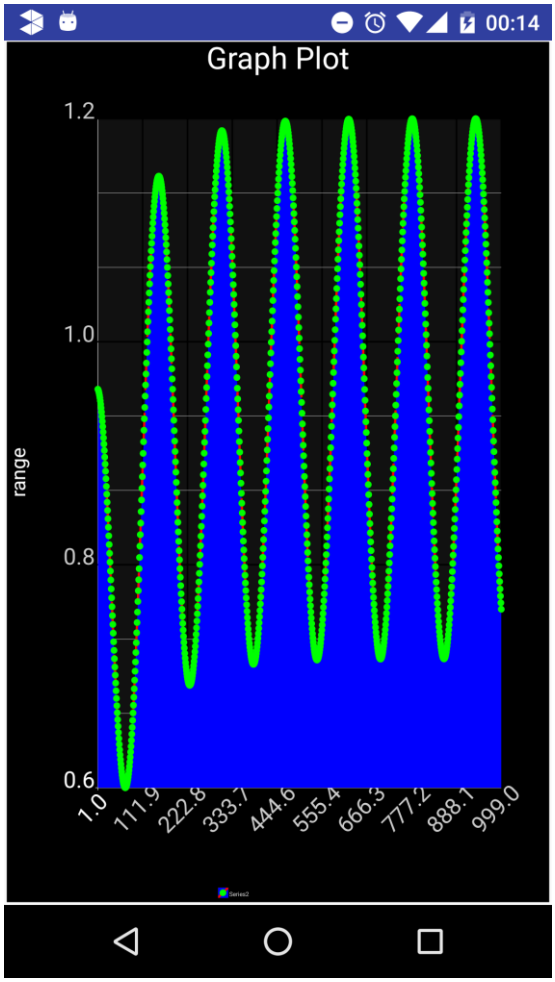
This app solves the electrical circuit using convolution technique, user is allowed to select the circuit either RL or RC, and also user selects delta value, which determines the length of the signals. Select the signal to plot graph. Follow the points given below to run the app.

1. 'VALUE SELECT' button, takes user to one more page, where he/she can select the type of circuit, values of the circuit components, amplitude for the input signal, type of input signal.
2. 'COMPUTE', when pressed convolution is carried out.
3. 'Solution': Displays values of the output, which can be used to verify the solution of the problem.

For this app, only output is plotted for different circuits and various input signals. Typical values of 'Amplitude' can be given from 1 to any real life value. Values of the circuit component are also any real life values. Input signals used are sine and step function. Below are some snapshot of the app. Figure 1. Home Screen. Figure 2. Value select screen. 3. Output of sin wave RC circuit.






00:14
Graph Plot
1.2
1.0
0.8
0.6
range
1.0
111.9
222.8
333.7
444.6
555.4
666.3
777.2
888.1
999.0

Chapter 4.

Here we are verifying the fourier series for various periodic signal inputs like sine wave, square wave, sawtooth, exponential and triangular wave. User can select the type of signal and number of fourier coefficient he/she wants to generate. All the codes return the fourier coefficients and the signal generated using those coefficients. User can verify the whether the reconstructed and input signal match. From theory we know with increase in fourier coefficients, signal reconstructed tends to resemble the input signal. Here coefficients are discrete, but graph plotted are continuous, hence it is requested to read them as discrete.

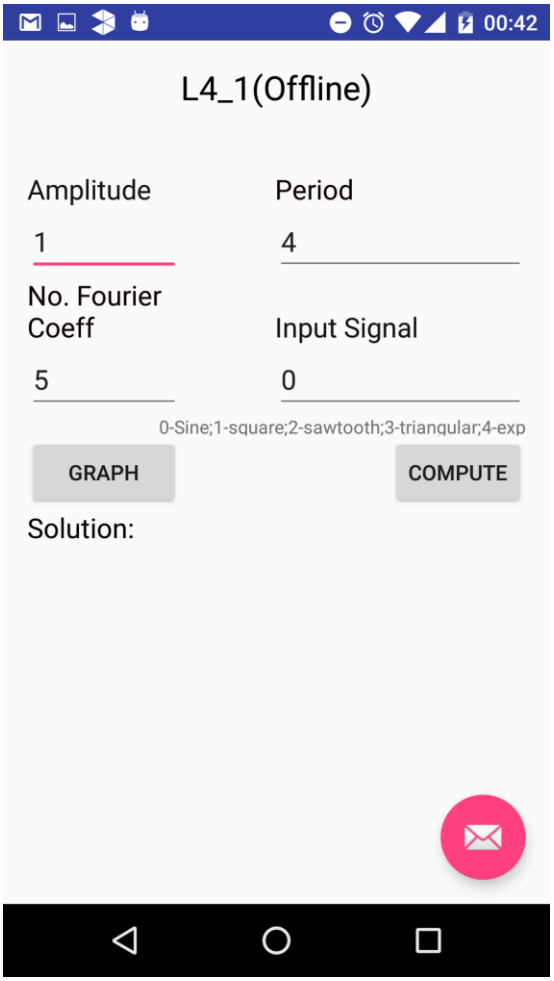
Note: L4_3 is same as L4_1 with extra exponential signal, hence it is combined and implemented.

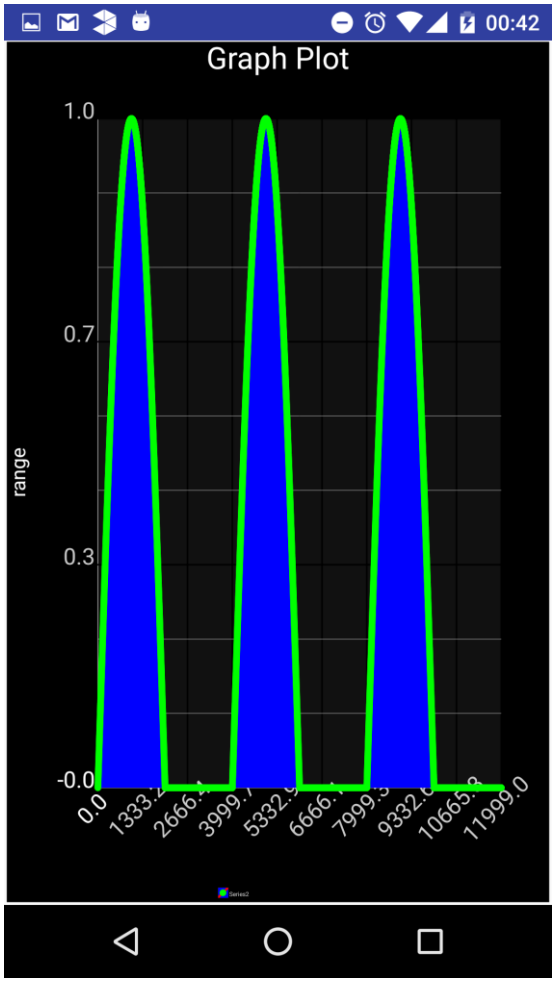
L4_1 (Offline)

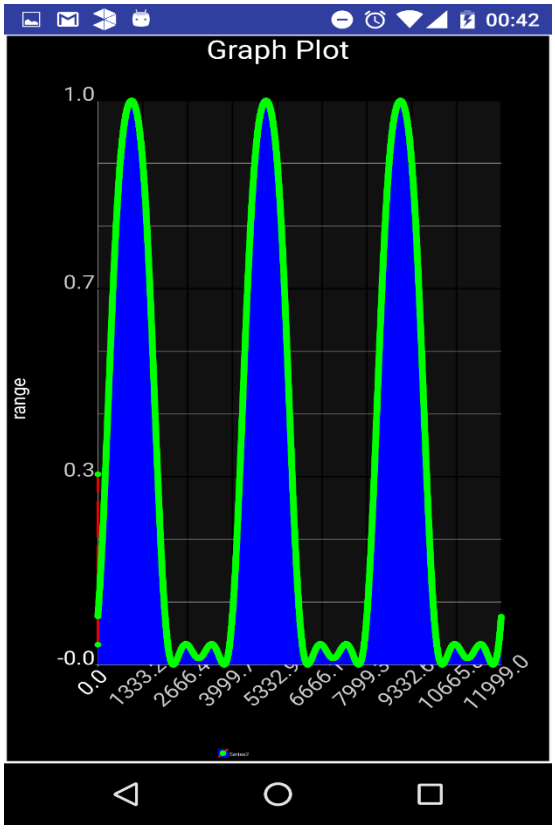
This app talks about fourier series implementation, user is allowed to selected the type of input and its parameter to generate. Select the signal to plot graph. Follow the points given below to run the app.

1. 'Amplitude' button, amplitude for the input signal.
2. 'Period', period for the input signal.
3. 'No. of Fourier Coefficients': *self-explanatory*.
4. 'Input Signal', type of input signal. In small fonts app shows what value denotes what signal.
5. 'Graph', allows you to plot input, fourier coefficients and signal constructed using fourier coefficients.
6. 'Compute', lets us solve the problem.
7. 'Solution': we can see values, help user to verify the solution.

Typical values of 'Amplitude', 'period', 'no. of fourier coeff' can be given from 1 to any real life value. Below are some snapshot of the app. Figure 1. Home Screen. Figure 2. Input signal. 3. Reconstructed signal.


00:42
L4_1(Offline)
Amplitude
Period
1
4
No. Fourier Coeff
Input Signal
5
0
0-Sine;1-square;2-sawtooth;3-triangular;4-exp
GRAPH
COMPUTE
Solution:


00:42
Graph Plot
1.0
0.7
0.3
-0.0
range
0.0
1333.2
2666.4
3999.7
5332.9
6666.1
7999.3
9332.6
10665.8
11999.0

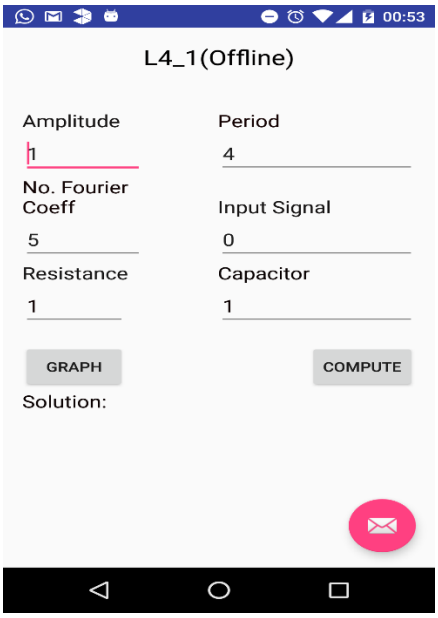

00:42
Graph Plot
1.0
0.7
0.3
-0.0
range
0.0
1333.2
2666.4
3999.7
5332.9
6666.1
7999.3
9332.6
10665.8
11999.0

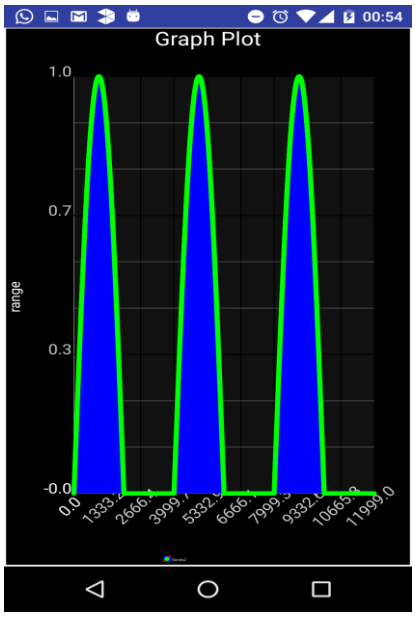
L4_2 (Offline)

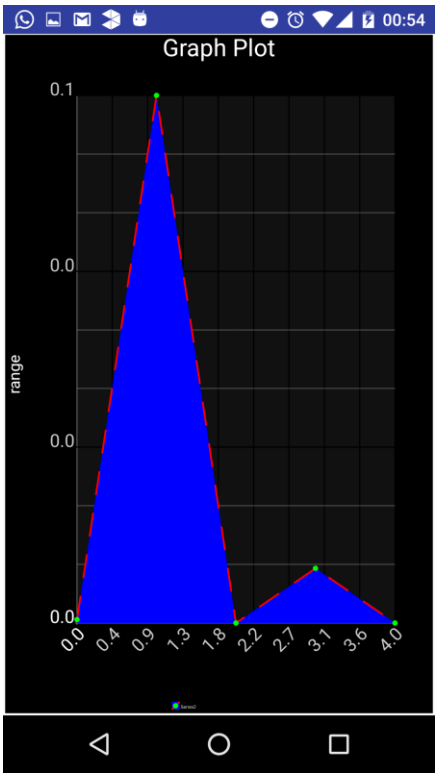
This app talks about fourier series implementation, to solve electrical circuit. For present problem we are talking about RC circuit. User is allowed to select the electrical component values. Select the signal to plot graph. Follow the points given below to run the app.

1. ‘Amplitude’ button, amplitude for the input signal.
2. ‘Period’, period for the input signal.
3. ‘No. of Fourier Coefficients’: *self-explanatory*.
4. ‘Input Signal’, type of input signal. In small fonts app shows what value denotes what signal.
5. ‘Resistance’ and ‘Capacitance’, user has to select R and C values.
6. ‘Graph’, allows you to plot input, magnitude of cosine and sine component, phase of cosine and sine components.
7. ‘Compute’, lets us solve the problem.
8. ‘Solution’: we can see values, help user to verify the solution.

Typical values of ‘Amplitude’, ‘period’, ‘no. of fourier coeff’ can be given from 1 to any real life value. Below are some snapshot of the app. Figure 1. Home Screen. Figure 2. Input signal. 3. Magnitude of cosine component. (read it as discrete values, as plot shown is continuous)






00:54
Graph Plot
0.1
0.0
0.0
0.0
range
0.0
0.4
0.9
1.3
1.8
2.2
2.7
3.1
3.6
4.0

Chapter 5.

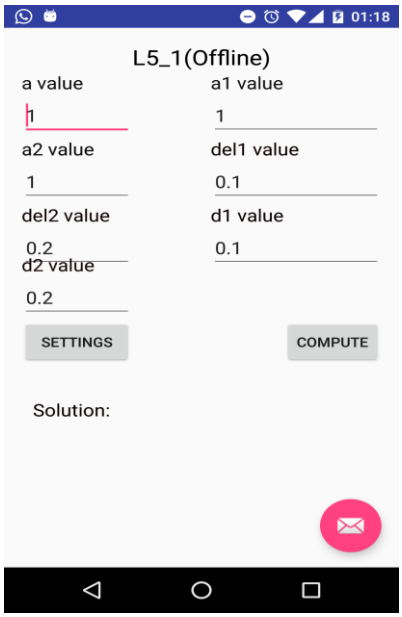
This chapter talks about the continuous time fourier transform for various signal inputs. User can select the type of signal and operation that needs to be performed on the signal in frequency domain and convert back to time domain. There are totally 3 apps for this chapter.

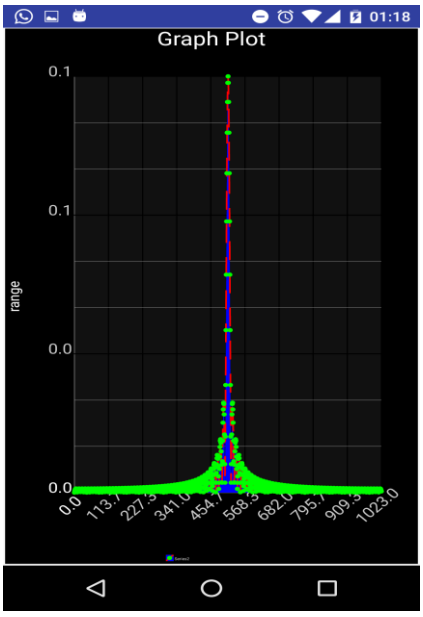
L5_1 (Offline)

This app talks about fourier transform implementation. For present problem we are talking about converting time domain to frequency domain and do some operations on it and get back to time domain using inverse fourier transform. User is allowed to select values. Select the signal to plot graph. Follow the points given below to run the app.

1. ‘a value’, ‘a1 value’, ‘a2 value’, ‘del1 value’, ‘del2 value’, ‘d1 value’, ‘d2 value’ allows user to input amplitude, size of the signal.
2. ‘Settings’ button takes user to next page and allows him to select inputs, their type and combination (add, convolve, multiply) and same case with output signal.
3. ‘Select input’, allows you to plot input, magnitude and phase of various output signals.
4. ‘Compute’, lets us solve the problem.
5. ‘Solution’: we can see values to verify.

Snaps from the app home page and graph (magnitude of y1).



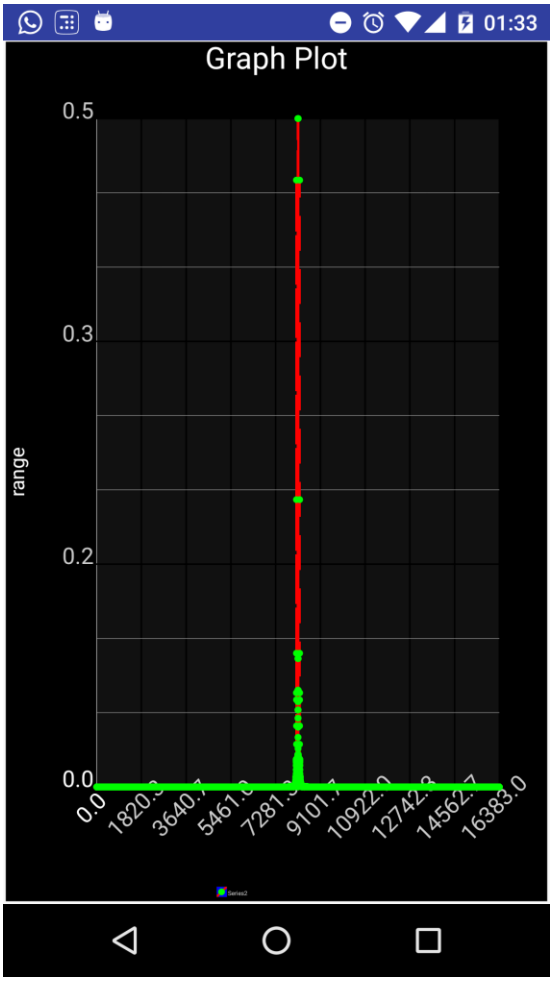


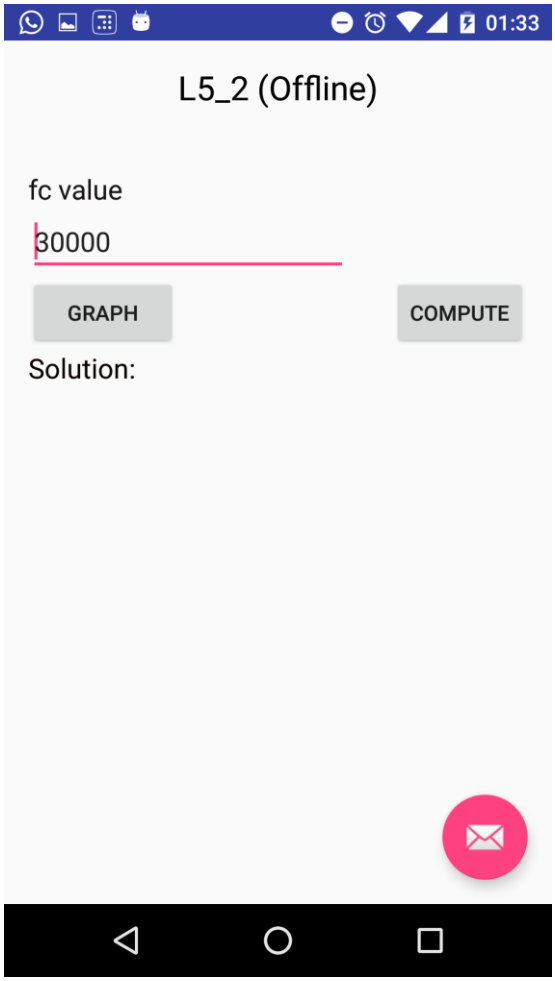
L5_2 (Offline)

Here app is taking the input as sampling frequency. Let's user to plot original input, modulated signal, signal along with the noise, and then signal with reduced noise.

1. 'fc value', allows user to input the carrier frequency.
2. 'GRAPH', allows user to select the waveform that he/she wants to implement like message signal, modulated signal, with noise, demodulated in both time and frequency domain.
3. 'Compute', lets us solve the problem.
4. 'Solution': we can see values to verify.

Snaps from the app home page and graph (frequency domain message signal).





Chapter 6.

This chapter talks digital signals and their transform. Also about sampling and quantization and reconstruction.

L6_1 (Offline)

This app talks about fourier transform implementation. For present problem we are talking about converting time domain to frequency domain and do some operations on it and get back to time domain using inverse fourier transform. User is allowed to select values. Select the signal to plot graph. Follow the points given below to run the app.

1. ‘amplitude’ allows user to input amplitude of the signal.
2. ‘Signal freq’, ‘sampling freq’, ‘phase’ button allows user to input frequency and phase.
3. ‘Quant Level’, quantization level for signal reconstruction.
4. ‘Graph’, allows user to select signal to plot. Input, sampled, reconstructed
5. ‘Compute’, lets us solve the problem.
6. ‘Solution’: we can see values to verify.

L6_2 (Offline)

This app talks about continuous time fourier transform and discrete fourier transform. Let’s user to select the signal, and performs depending on the input signal.

1. ‘Continuous or discrete’ allows user to input type of signal.
2. ‘Graph’, allows user to select signal to plot. Analog, discrete, FS or DFT.
3. ‘Compute’, lets us solve the problem.
4. ‘Solution’: we can see values to verify.

L6_3 (Offline)

This app talks about dual tone multi frequency encoder, lets user to input key and shows corresponding output.

1. ‘Key’ to key in the button user has pressed. Input can be from 0-9.
2. ‘Tone’ asks user whether he wants busy tone or ringing tone.
3. ‘Graph’, allows user to select signal to plot. Tone heard when key is pressed or busy/ringing tone.
4. ‘Compute’, lets us solve the problem.
5. ‘Solution’: we can see values to verify.

Doppler Effect

The Doppler effect denotes the change in frequency and wavelength of a wave as perceived by an observer moving relative to the wave source. The Doppler effect can be demonstrated via time scaling of Fourier series. The observer hears the siren of an approaching emergency vehicle with different amplitudes and frequencies as compared to the original signal. As the vehicle passes by, the observer hears another amplitude and frequency. Application allows the user to input whether signal is approaching or passing by, then plots the relevant signal.

1. 'Freq increment' to key in scale of increased frequency of the signal
2. 'Freq Decrement' scale of decreased frequency of the signal.
3. 'Input Frequency' signal frequency.
4. 'Graph', allows user to select signal to plot. Both in time and freq domain for input, approaching and passing by signals
5. 'Compute', lets us solve the problem.
6. 'Solution': we can see values to verify.

Echo cancellation

Deconvolution technique is used to get back the original signal that is to perform the echo cancellation. User is allowed to input the amplitude and delay of the echo signal and matlab is used to model the echoed signal, later using deconvolution technique, original signal is recovered.

1. 'a value' amplitude of the echoed signal.
2. 'delay' delay of the echo signal.
3. 'Graph', allows user to select signal to plot. Original, delayed, recovered signals
4. 'Compute', lets us solve the problem.
5. 'Solution': we can see values to verify.

Music Synthesis

Here various techniques like additive synthesis, self modulation, frequency modulation. Algorithm were mentioned in the text book, app has implemented 3 algorithm, easy medium and complicated techniques. User can select the algorithm is wants and run the code to see the graphs.

1. ‘A1-A6’ amplitude of the signals.
2. ‘freq1-freq6’ freq of the signals
3. ‘Graph’, allows user to select signal to plot. Music signal, AN, BN
4. ‘Compute’, lets us solve the problem.

Note: For this particular app, if user wants to plot two parameters back to back app might crash. Hence it is advised to close the app after particular operation and run app again to check different parameter.