EEL 4514C: Communication Systems and Components (Fall 2025)

Lab 1: Fourier Series

Deliverables Due: At 11:59pm on September 19, 2025.

Pre-Lab

Learning GRC:: Start by completing these two quick tutorials on variables in GnuRadio Companion Flowgraphs:

- Python Variables in GRC: `https://wiki.gnuradio.org/index.php?title=Python_Variables_in_GRC`
- Variables in Flowgraphs: `https://wiki.gnuradio.org/index.php?title=Variables_in_Flowgraphs`
- Signal Data Types: `https://wiki.gnuradio.org/index.php?title=Signal_Data_Types`

You do not need to document the completion of these tutorials, but you will need these concepts to be successful in the lab.

Understanding Fourier Series of a bipolar square wave: In this lab, we will use GnuRadio Companion to calculate the first few coefficients of the Fourier Series of a periodic waveform and to reconstruct the approximation to that waveform using those first few coefficients.

For a real signal periodic $s(t)$ with period T_0, the functions

$$a_n(t) = \cos\left(\frac{2\pi nt}{T_0}\right), \text{ and}$$
$$b_n(t) = \sin\left(\frac{2\pi nt}{T_0}\right)$$

form a basis for $s(t)$ (provided $s(t)$ has a finite number of discontinuities per period).

Then we can find a signal-space representation for $s(t)$ using the coefficients

$$a_n = \frac{\langle\ s(t),\ a_n(t)\ \rangle}{\|a_n(t)\|^2}, \text{ and}$$
$$b_n = \frac{\langle\ s(t),\ b_n(t)\ \rangle}{\|a_n(t)\|^2}.$$

Note that (with the exception of $\|a_0(t)\|^2$ and $\|b_0(t)\}$, the values of $\|a_i(t)\|^2$ and $\|b_i(t)\|^2$ are equal and independent of i, so we will just lump them into a constant from here on.

We can reconstruct $s(t)$ as

$$s(t) = \sum_{n=0}^{\infty} \left[a_n a_n(t) + b_n b_n(t)\right],$$

where $b_0 = 0$ and a_0 is the DC term.

For a finite number of coefficients, we have an approximation of $s(t)$,

$$\hat{s}(t) = \sum_{n=0}^{N} \left[a_n a_n(t) + b_n b_n(t) \right].$$

1. Referring to Lecture 5, write down an equation for the complex exponential Fourier Series coefficients for the waveform in Fig. 2.20 in the textbook. Show that if you consider the reconstruction of the portion of the signal at f_n and $-f_n$ that you get the values shown in (2.86).

2. Review Example 2.9 to see how we can use a linear transformation to make the unipolar square wave into a bipolar one. Write down the equation for this linear transformation (it is given in the book).

3. Now make a table of the coefficients of the first 7 even and odd harmonics based on (2.88). Write the value both as fraction involving π and as a decimal value (2 digits after the decimal point is sufficient).

4. Explain using symmetry arguments why are all the sine are terms zero.

5. Draw a picture or create a plot that shows the product of the square wave and the cos wave at the fundamental frequency. We have a common EE term for the action of the square wave on the cos wave in this case. What is that?

6. Draw a picture or create a plot that illustrates why the Fourier series coefficient is zero for the cos harmonic at twice the fundamental frequency. Explain why this will be true for all harmonics at even multiples of the fundamental frequency.

7. If a system has a sampling rate of 48000 and a periodic signal of 400 Hz is generated in that system, how many samples are used to represent each period of the signal?

You are encouraged to complete as much of the following as you can before arriving to the lab!

Lab Part 1: Aligning Square Wave with Cosine Wave

Blocks Needed:

1	Variable
2	Signal Source
1	Delay
1	Multiply
3	Throttle
1	Audio Sink
1	QT GUI Time Sink

You should use **real** signals in all of these blocks; we will use the `float` signal type for real signals. The default signal type in GRC is `complex`. Because of this, you will need to change every block's I/O type from `complex` to `float`. There are two ways to do this:

- You can double-click on each block and then change the signal types using the dropdown at the top of the General tab.
- You can select all the blocks (using Control-A or Command-A) and then press the Down arrow key on your keyboard to cycle through the signal types.

- Set the standard **samp_rate** variable to 48000. (If you find your audio card does not support 48000 samples/s, you can try 44100.)
- Create a variable **f0** to represent the fundamental frequency and set it to 400 Hz.
- Use one signal source to create a bipolar square wave a frequency **f0** whose amplitude switches between 50 and -50.
- Use a second signal source to generate a cosine wave at frequency **f0**.
- The square wave that is generated in GRC is not aligned to have the middle of the positive pulse at time t=0. To get the signal to align with the cosine wave as in the textbook, we will need to introduce a delay. Delay the square wave and multiply the delayed square wave with the cosine signal. Adjust the delay until the output of the multiplier looks like a rectified cosine; be sure that the output does not look like the rectified cosine has a tilt or slant to it. (Note that the Delay block cannot adjust the delay according to a GUI Slider. So, you will have to try different delays and rerun the block. The necessary delay is some typical fraction of the number of samples in one period of the waveform. You will need to know this delay for the remaining parts of this lab.
- Take a screen capture of your block diagram and the rectified output for your lab report.

Lab Part 2: Determining Fourier Series Coefficients

Blocks Needed:

1	Variable
5	Signal Source
1	Delay
4	Multiply
4	Low Pass Filter
4	Throttle
4	QT GUI Time Sink

(You could instead use 1 QT GUI Time Sink with 4 inputs, in which case it may be easier or harder to read the necessary values.)

- Use the variable block for the fundamental frequency.
- Create a bipolar square wave with the same parameters as before and delay it by the amount from part 1.
- Add another Signal Source set to generate a cosine at the fundamental frequency and multiply it by the delayed square wave. Use a QT GUI Time Sink to verify that it is rectified.
- To integrate the product of the delayed square wave and the cosine wave, use a low pass filter with a cutoff frequency of 10 Hz and a transition width of 5 Hz. (These are chosen to pass DC. You can experiment with other values.)
- Use a Throttle block between the low pass filter and the time sink. After a small amount of settling time, the observed output of the low pass filter should converge to a constant. Note the value of that constant. This value is a constant times the Fourier Series coefficient – we will find that constant further below.
- Add additional cosine sources at 3, 5, and 7 times the fundamental frequencies, use a low pass filter as an integrator, and determine the output constants.
- Get a screen capture of your block diagram and of the GUI Time Sink(s) showing the values at the outputs of the low-pass filters.
- Create a table with all of the scaled Fourier Series coefficients, which are the constant values observed above. In another column of the table, list the corresponding values from part (a) of the Pre-Lab. Determine the scaling factor between these two sets of values.

Lab Part 3: Reconstruct Approximation of Square Wave Using Fourier Series Coefficients

Blocks Needed:

1	Variable
5	Signal Source
1	Delay
1	Add
3	QT GUI Chooser
2	Throttle
1	QT GUI Time Sink
1	Audio Sink

Note: you will need a lot of blocks that are the same as in the previous part of the lab. You may wish to use the Save As menu item to save your previous GRC file under a new name.

- Use the variable block for the fundamental frequency.
- Create a bipolar square wave with the same parameters as before and delay it by the amount from part 1. Connect the output to the QT GUI Sink and also to one input of the Audio sink. (If you did not bring headphones, then skip connecting the square wave to the audio sink.)
- Generate cosine waveforms at 1, 3, 5, and 7 times the fundamental frequency. Set the amplitude of each according to the values found in part 2 of this lab.
- Add together the different scaled cosine components and put the output into the QT GUI Sink and also into one input of the Audio sink.
- Compare the square wave and the approximated version.
- Now, add the 3 QT GUI Chooser blocks and set them to give either a value of 0 or 1 depending on which radio button is selected. These will be used to turn on or off the harmonics above the fundamental frequency, so give them reasonable IDs and labels. Have them default to 1. Change the amplitudes of the cosine sources so that they multiply by the corresponding IDs, allowing that signal to be attenuated to zero using a switch.
- Compare the accuracy of the approximation with the original square wave as you turn off the cosines from highest frequency to lowest frequency. How many ripples are present in each half period when you are using 4 cosines, 3 cosines, 2 cosines, 1 cosine? Take a screen capture for each of these cases to use in your lab report.

Lab Part 4: Plotting the Error Signal

Add the blocks necessary to plot the error signal between the square wave and its approximation. Introduce a GUI Range block so that you can scale up the approximation until the error signal looks minimized. Note that you may need to adjust the delay by ± 1 unit to get the best error signal.