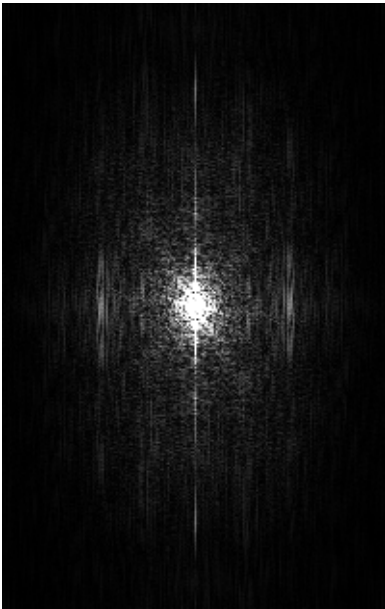


Fourier Transform and Applications

- Part 1

ECE generic

Han Q. Le (c)

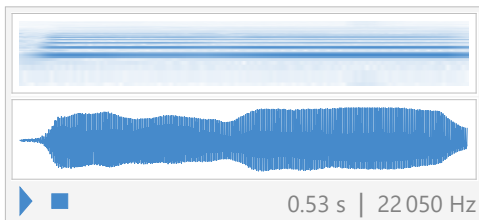


Jean-Baptiste-Joseph Fourier
(1768-1830)

1. Introduction: numerical discovery approach

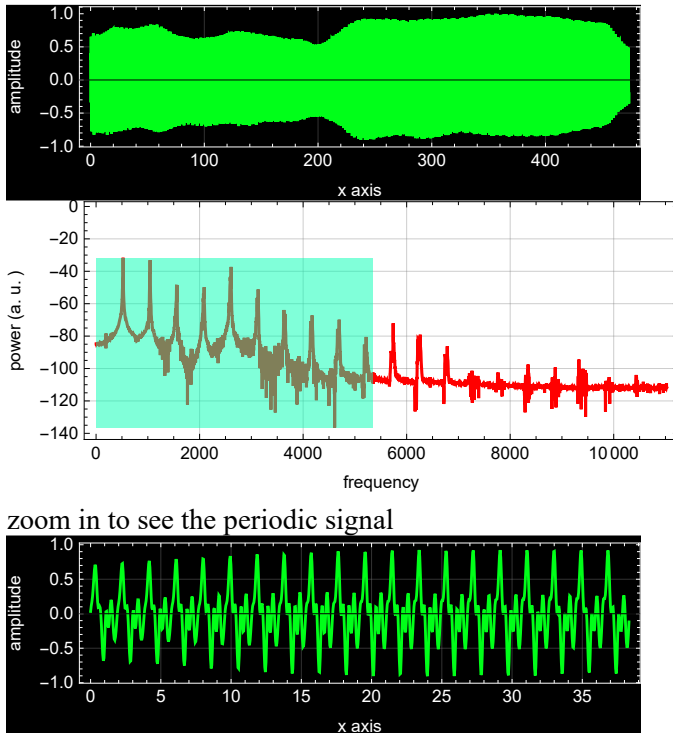
1.1 Run the app demo

We have made music and see a sound note can be a “fairly” periodic function:



sound signal
form

Fourier trans-

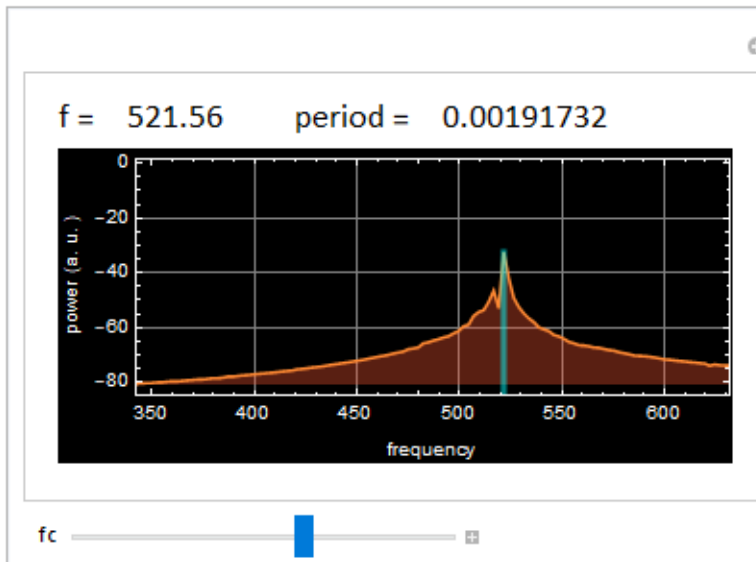
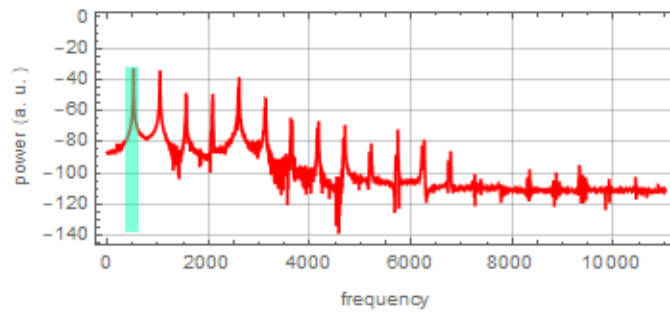


What is most remarkable is that the Fourier spectrum shows many peaks that have periodic spacing. It means that the sound signal can be written as:

$$x[t] \sim \sum_{m=-p}^p X_m e^{i m 2 \pi f t} \quad (1.1.1)$$

where f is the fundamental frequency found below:

Linear dB

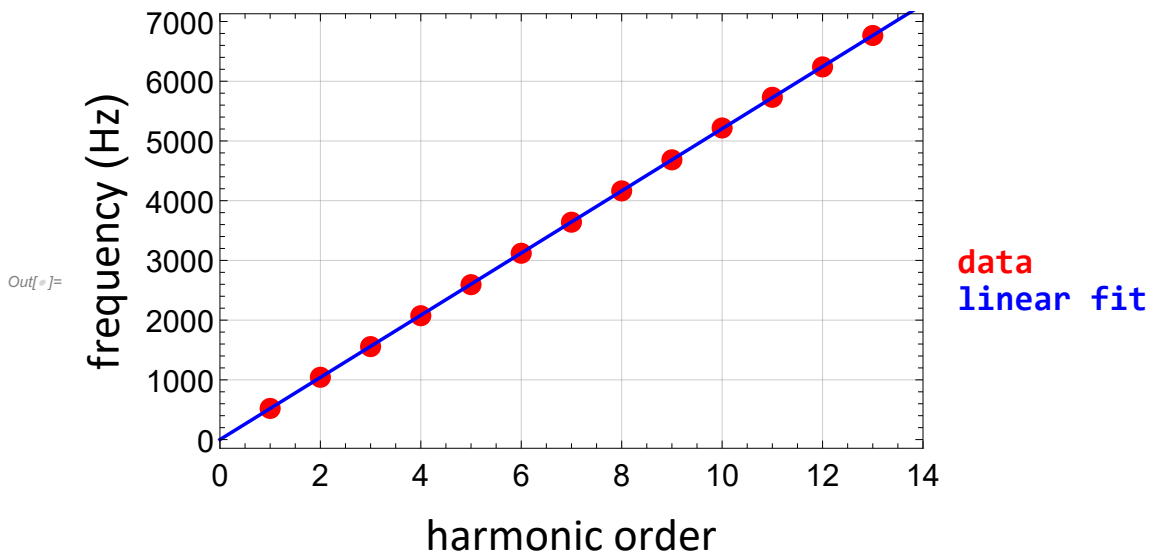


and p are the number of peaks that are above the “fuzzy” noise floor.
 Are those peaks are really regularly spaced? We can check by plotting:

```

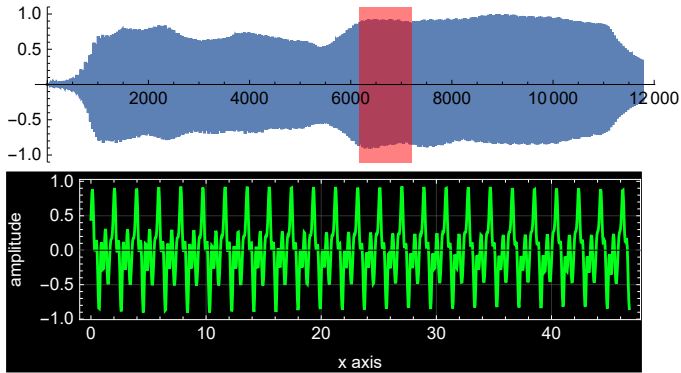
In[ ]:= fharmonic = {521.73`, 1043.46`, 1556.74`, 2074.`, 2594.`,
  3120.`, 3640.`, 4164.`, 4684.`, 5217.31`, 5731.`, 6240.`, 6766.};
ffit = LinearModelFit[Transpose[{Range[Length[fharmonic]], fharmonic}],
  x, x, IncludeConstantBasis -> False];
Show[ListPlot[fharmonic, PlotStyle -> {PointSize[0.03], Red},
  PlotRange -> {{0, Length[fharmonic] + 1}, All}
  , Frame -> True, GridLines -> Automatic
  , FrameLabel ->
    (Style[#, 24, Black, FontFamily -> "Calibri"] & /@ {"harmonic order", "frequency (Hz)"}
  , LabelStyle -> {16, Black, FontFamily -> "Arial"}
  , PlotLegends -> Style["data", Red, 18, Bold, FontFamily -> "Arial"]
  , Plot[ffit[x], {x, 0, Length[fharmonic] + 1}, PlotStyle -> {Thickness[0.005], Blue}
  , PlotLegends -> Style["linear fit", Blue, 18, Bold, FontFamily -> "Arial"]
  , ImageSize -> 450]

```

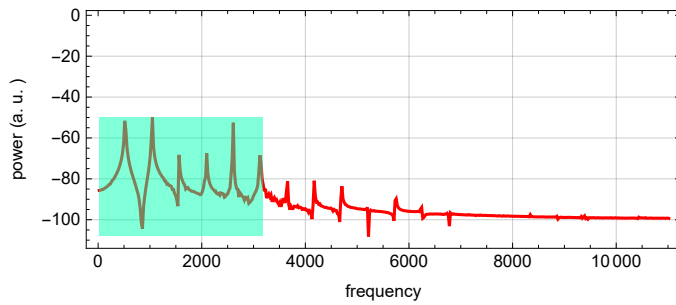


1.2 Construction of the signal

Let's use Eq. (1.1.1) to reconstruct the signal. Since the signal is over a long duration with some slight variation, we'll choose a narrow interval as show below:

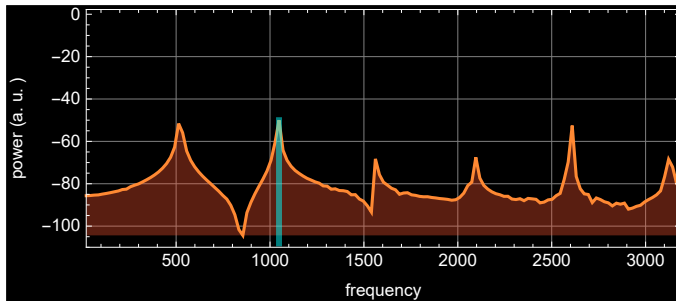


"Linear" "dB"



save

"f = " fc " period = " $\frac{1}{fc}$



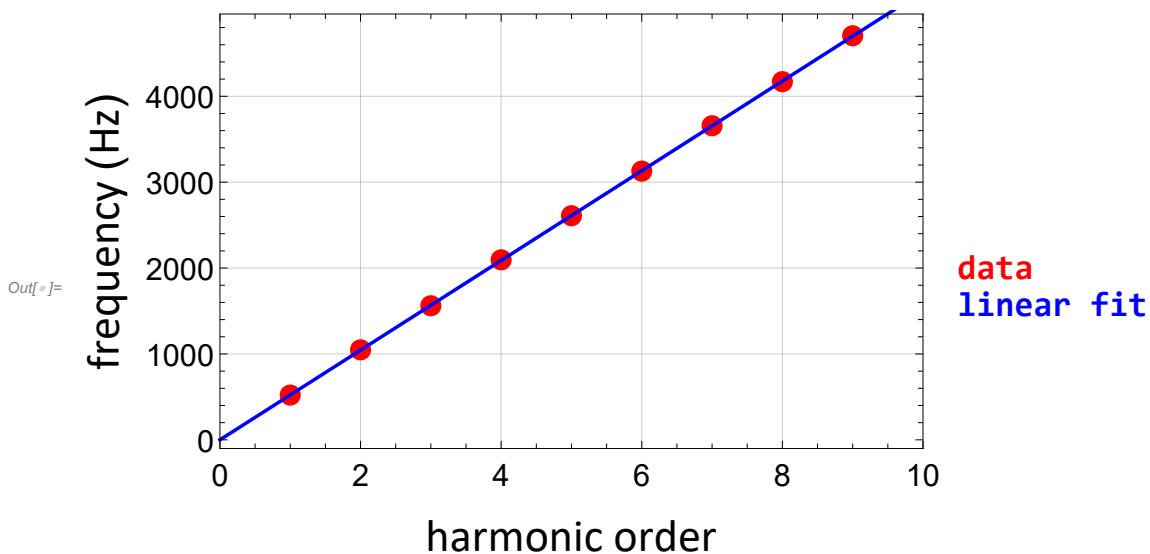
Over this range of signal:

```

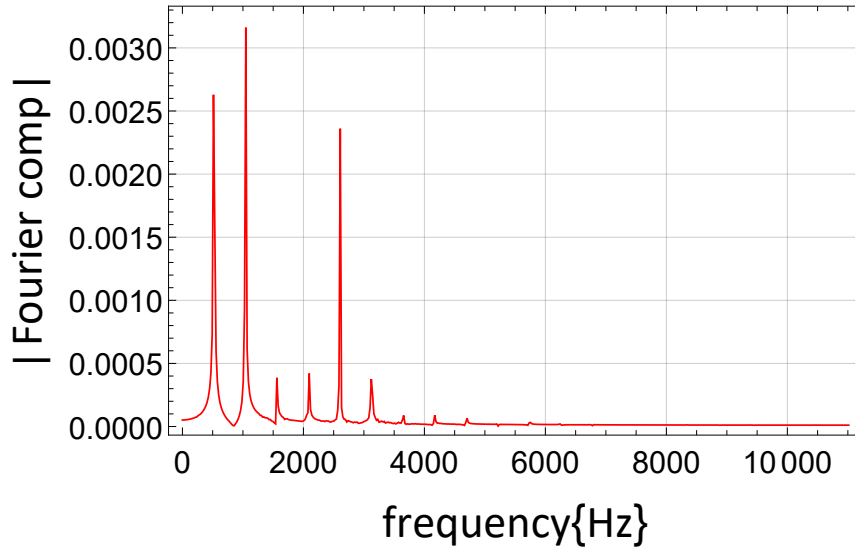
In[ ]:= fharm0 = {521.`, 1047.963143`, 1561.251212`, 2095.926285`,
  2609.214355`, 3128.5`, 3657.177498`, 4170.465567`, 4705.14064`};
ffit0 = LinearModelFit[Transpose[{Range[Length[fharm0]], fharm0}],
  x, x, IncludeConstantBasis -> False];
Grid[{"Best-fitted fundamental frequency (Hz) ", f0 = ffit0[[1, 2, 1]]}
, Background -> Hue[0.15, 1, 1], Frame -> True]
Show[ListPlot[fharm0, PlotStyle -> {PointSize[0.03], Red},
  PlotRange -> {{0, Length[fharm0] + 1}, All}
, Frame -> True, GridLines -> Automatic
, FrameLabel ->
  (Style[#, 24, Black, FontFamily -> "Calibri"] & /@ {"harmonic order", "frequency (Hz)"}
, LabelStyle -> {16, Black, FontFamily -> "Arial"}
, PlotLegends -> Style["data", Red, 18, Bold, FontFamily -> "Arial"]
, Plot[ffit0[x], {x, 0, Length[fharm0]}, PlotStyle -> {Thickness[0.005], Blue}
, PlotLegends -> Style["linear fit", Blue, 18, Bold, FontFamily -> "Arial"]
, ImageSize -> 450]

```

Out[]:= Best-fitted fundamental frequency (Hz) 522.146



Let's look at the Fourier components:



Here are the Fourier component X_m :

```
In[4]:= Xfourier = {0.00105554` - 0.0024075` i,
  -0.00050213` + 0.00312062` i, 0.00019292000000000002` - 0.00033453` i,
  0.00038492` + 0.00016834` i, -0.00167107` + 0.00166658` i,
  -0.00036519` + 0.00009031000000000001` i, -0.00007283` + 0.00004933000000000004` i,
  -0.00008946` + 9.14` *^-6 i, 0.00001506` + 0.00006433` i};
```

```
Length[Xfourier]
```

```
Out[5]= 9
```

Now, we just apply the formula:

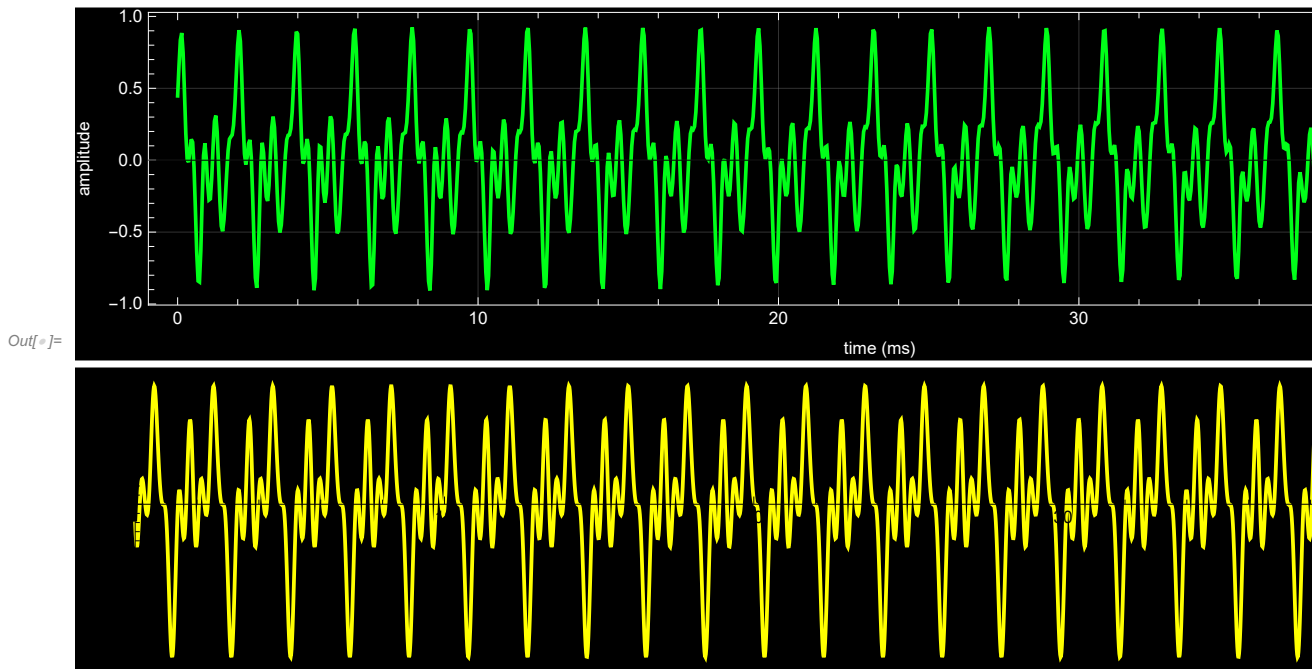
$$x[t] \sim \sum_{m=-p}^p X_m e^{i m 2 \pi f t} \quad (1.1.1)$$

One more thing: we don't have to sum from $-p$ to p like the equation says. Because the signal is real, we can just add the positive p components and take the Real part.

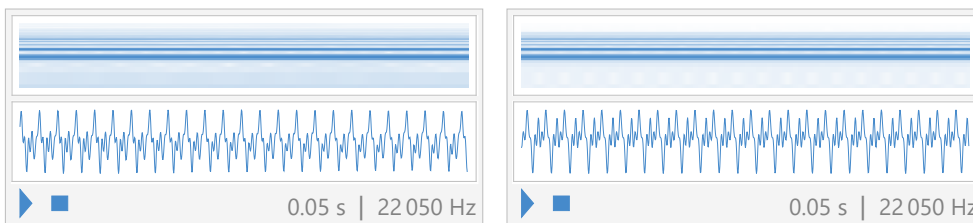
So, this is the synthesized signal:

```
In[6]:= p = 9;
xsynth[t_] := 2 Sum[Re[Xfourier[[m]] e^{i m 2 \pi f0 t}], {m, 1, p}];
```

Let's plot our synthesized signal (yellow) to compare with the original (green):



Pretty close, given that we use a mere 9 components out of 515. This what they sound like:



2. Fourier's theorem: an example

2.1 Discussion

Is it true that any periodic function can be expressed as a sum of harmonics with frequencies that are integral multiples of the fundamental frequency? Yes, that is exactly what Fourier's theorem says,...

which means that we can represent a periodic function as a series, known as Fourier's series

$$x[t] = \sum_{m=-\infty}^{\infty} X_m e^{i m 2\pi f_0 t} = \sum_{m=-\infty}^{\infty} X_m e^{i m \omega_0 t} ; f_0 = \frac{1}{T} \quad (2.1.1)$$

where T is the period, and the Fourier series coefficient is obtained as

$$X_m = \frac{1}{T} \int_0^T x[t] e^{-i m \omega_0 t} dt \quad (2.1.2)$$

It is quite easy to construct a periodic function. All we have to do is to define it over one period, and repeat it. In programming language, we use modulo function so that the variable repeating itself after each period.

Consider as an example this simple function:

$$x[t] = \text{UnitBox}\left[\frac{t}{\sigma}\right] \left(1 - 2 \text{Abs}\left[\frac{t}{\sigma}\right]\right) \text{ for } t \in [-0.5, 0.5]$$

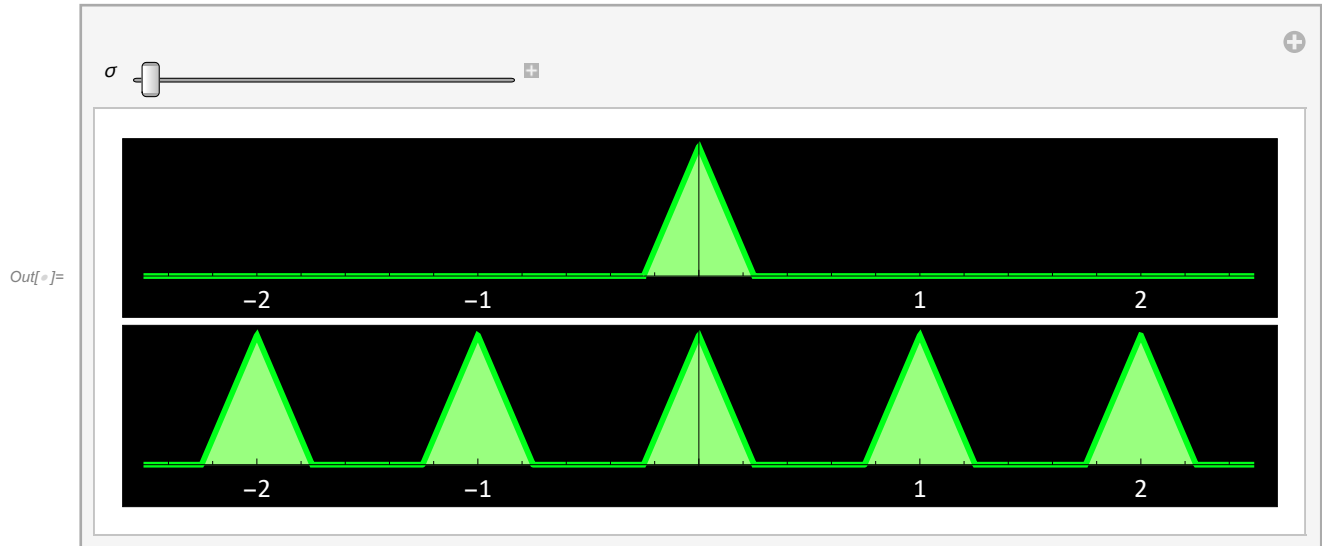
which can also be written as:

$$x[t] = \text{If}[t \leq -\frac{\sigma}{2}, 0, \text{If}[t \leq 0, \frac{2t+\sigma}{\sigma}, \text{If}[t < \frac{\sigma}{2}, \frac{\sigma-2t}{\sigma}, 0]]]$$

2.2 An example

2.2.1 Plot the function

■



2.2.2 Obtain Fourier coefficients

Let's apply Fourier's theorem to it. The Fourier's theorem says first, we should obtain the coefficient:

$$X_m = \frac{1}{T} \int_0^T x[t] e^{-im\omega_0 t} dt = \frac{1}{T} \int_0^T x[t] e^{-im2\pi t/T} dt$$

`In[]:= σ1 =.; T1 = 1; m =.;`

`xtriangle[t_, σ_] := If[t ≤ -σ/2, 0, If[t ≤ 0, (2t+σ)/σ, If[t < σ/2, (σ-2t)/σ, 0]]];`

$$\int_{-T1/2}^{T1/2} \text{xtriangle}[t, \sigma1] e^{-im2\pi t/T1} dt$$

$$\text{Out[]:= } \begin{cases} -\frac{e^{-im\pi\sigma1}(-1+e^{im\pi\sigma1})^2}{2m^2\pi^2\sigma1} & 0 < \sigma1 \leq 1 \\ \frac{e^{-im\pi}(-1+e^{im\pi})(1-e^{im\pi+i m\pi+i e^{im\pi} m\pi-i m\pi\sigma1-i e^{im\pi} m\pi\sigma1})}{2m^2\pi^2\sigma1} & \sigma1 > 1 \\ 0 & \text{True} \end{cases}$$

Hence, the Fourier coefficient is: $-\frac{e^{-im\pi\sigma1}(-1+e^{im\pi\sigma1})^2}{2m^2\pi^2\sigma1}$

But we can simplify it a bit:

$$\text{In[*]:= } - \left(\text{ExpToTrig} \left[\text{Expand} \left[e^{-i m \pi \sigma_1} (-1 + e^{i m \pi \sigma_1})^2 \right] \right] / (2 m^2 \pi^2 \sigma_1) \right)$$

$$\text{Out[*]:= } - \frac{-2 + 2 \text{Cos}[m \pi \sigma_1]}{2 m^2 \pi^2 \sigma_1}$$

$$\text{In[*]:= } \text{FullSimplify} \left[- \frac{-2 + 2 \text{Cos}[m \pi \sigma_1]}{2 m^2 \pi^2 \sigma_1} \right]$$

$$\text{Out[*]:= } \frac{1 - \text{Cos}[m \pi \sigma_1]}{m^2 \pi^2 \sigma_1}$$

$$\text{In[*]:= } \frac{1}{m^2 \pi^2 \sigma_1} \left(\text{Simplify} \left[1 - \text{Cos} \left[\frac{m \pi \sigma_1}{2} \right]^2 \right] + \text{Sin} \left[\frac{m \pi \sigma_1}{2} \right]^2 \right)$$

$$\text{Out[*]:= } \frac{2 \text{Sin} \left[\frac{m \pi \sigma_1}{2} \right]^2}{m^2 \pi^2 \sigma_1}$$

Hence, the Fourier coefficient is:

$$\text{Xfc}[m_, \sigma_1_] := \frac{\sigma_1}{2} \left(\frac{2 \text{Sin} \left[\frac{m \pi \sigma_1}{2} \right]}{m \pi \sigma_1} \right)^2 ;$$

or

$$\text{Xfc}[m_, \sigma_1_] := \frac{\sigma_1}{2} \text{Sinc} \left[\frac{m \pi \sigma_1}{2} \right]^2 ;$$

Why do we use Sinc function? just for this simple reason

$$\text{In[*]:= } \left\{ \frac{2 \text{Sin} \left[\frac{m \pi \sigma_1}{2} \right]}{m \pi \sigma_1}, \text{Sinc} \left[\frac{m \pi \sigma_1}{2} \right] \right\} /. m \rightarrow 0$$

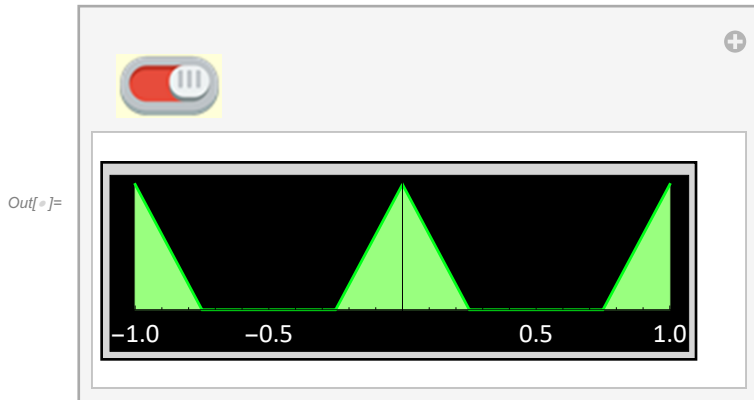
Power: Infinite expression $\frac{1}{0}$ encountered.

Infinity: Indeterminate expression $\frac{0 \text{ComplexInfinity}}{\pi \sigma_1}$ encountered.

$$\text{Out[*]:= } \{\text{Indeterminate}, 1\}$$

2.2.3 Apply the Fourier's theorem numerically

■ Code



3. Fourier signal synthesis

Can we use Fourier's theorem to synthesize a signal? Certainly!

3.1 Example 1: signal synthesis concept

UNIVERSITY of HOUSTON App by Han Q. Le ©

ECE generic – Fourier signal synthesis

RUN STATUS/CONTROL OFF

Out[=]=

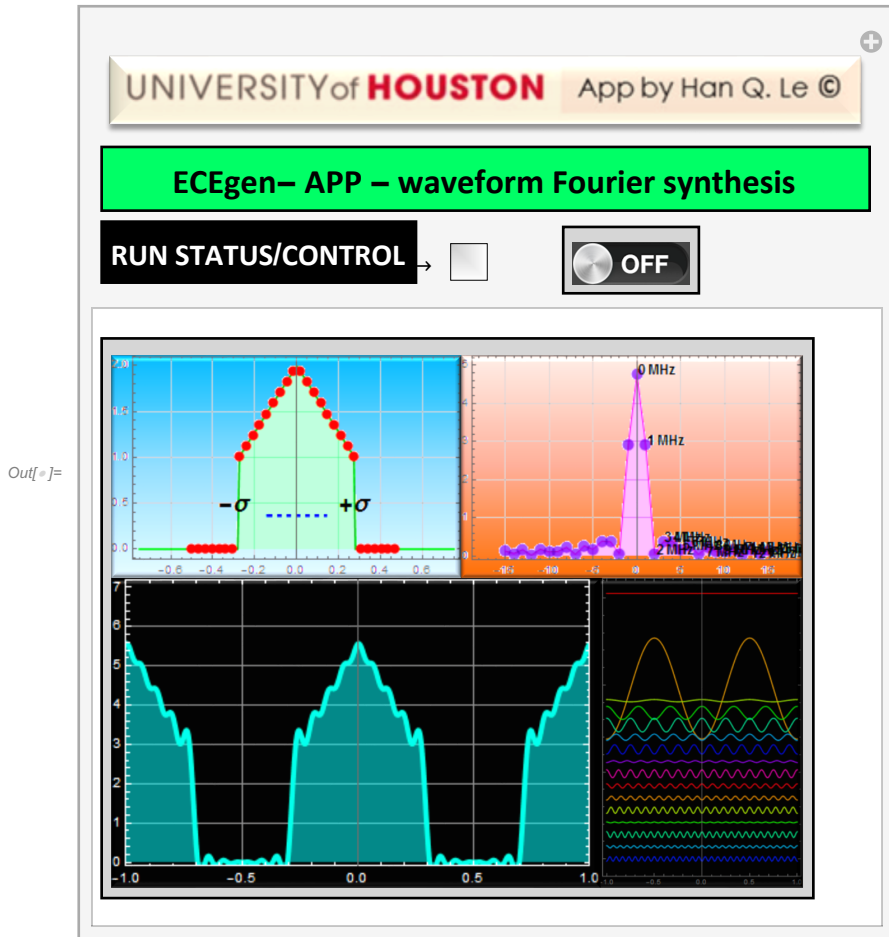
Synthesize a pulse signal by adding 8 channels

amplifier/attenuator to normalize signal

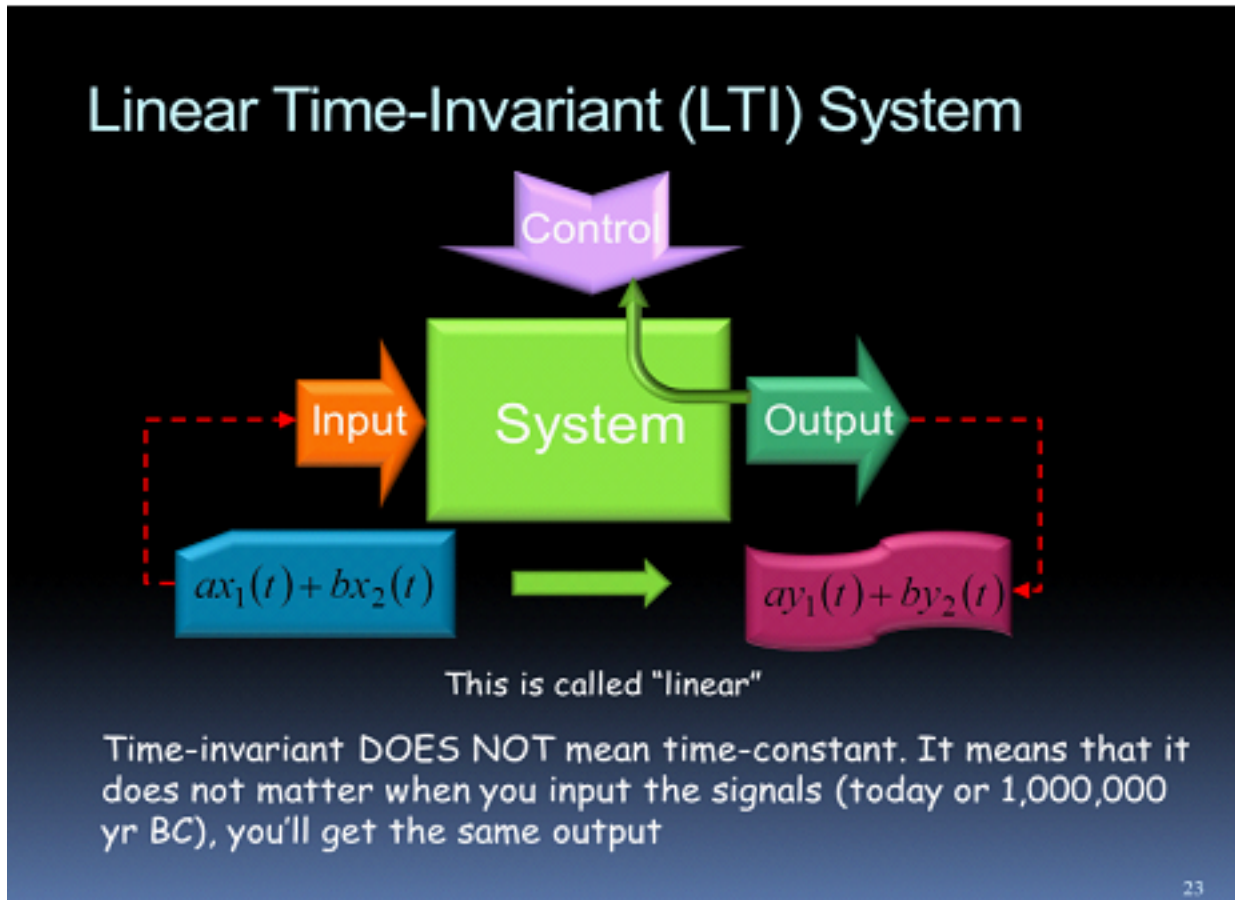
output signal

- as narrow pulse width as possible
- 50% period between pulse: ripple must be within $\pm 0.5\%$ of peak.
- pulse negative spike must be $< 0.5\%$ peak

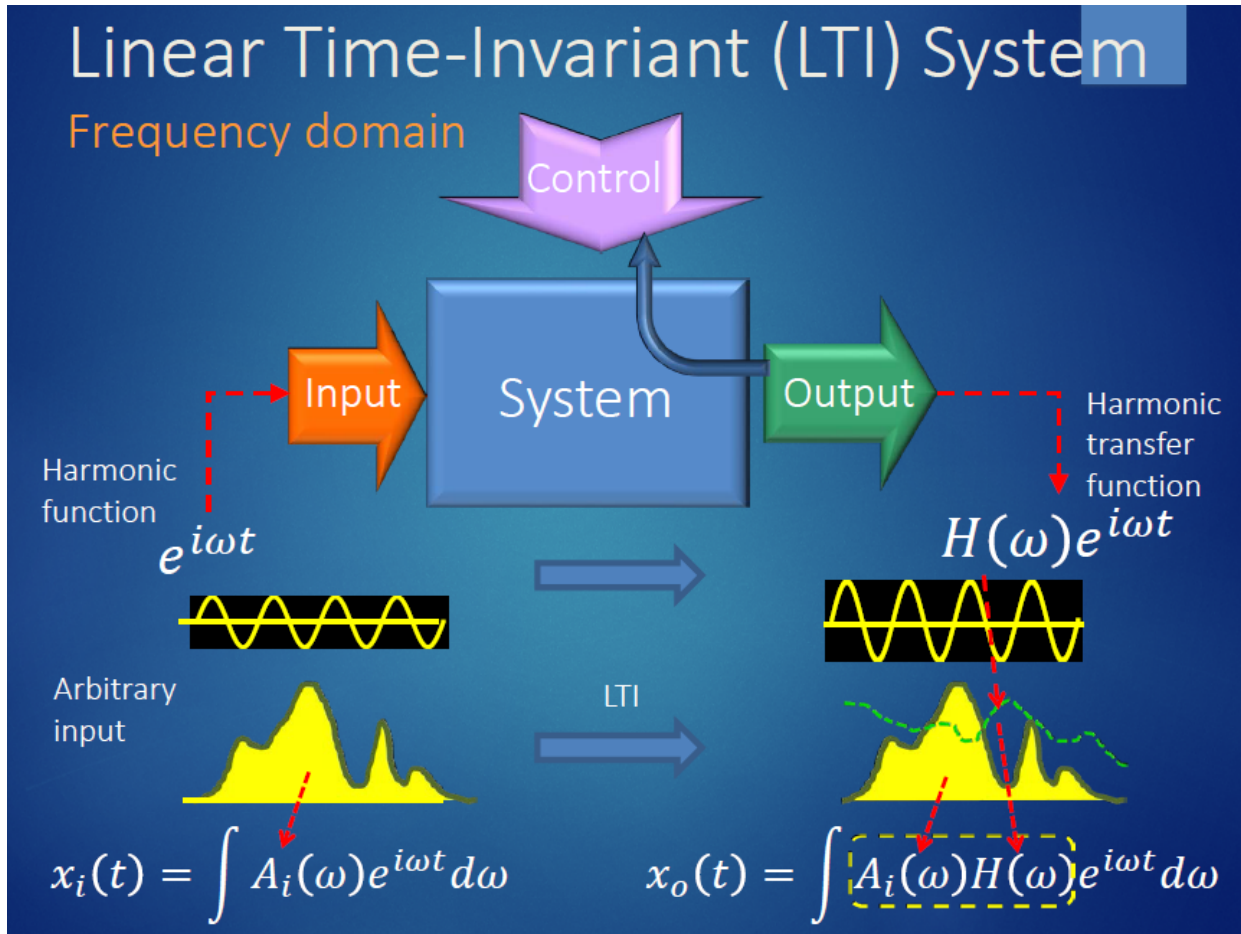
3.2 Example 2



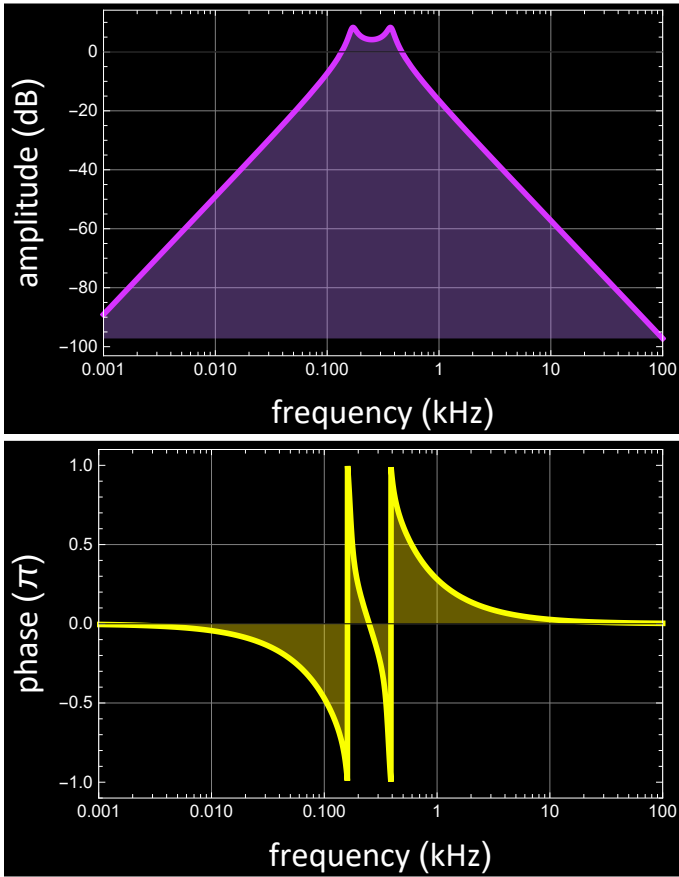
4. Basic concept of Fourier analysis for a LTI system



A periodic signal can be decomposed into a sum of its Fourier components

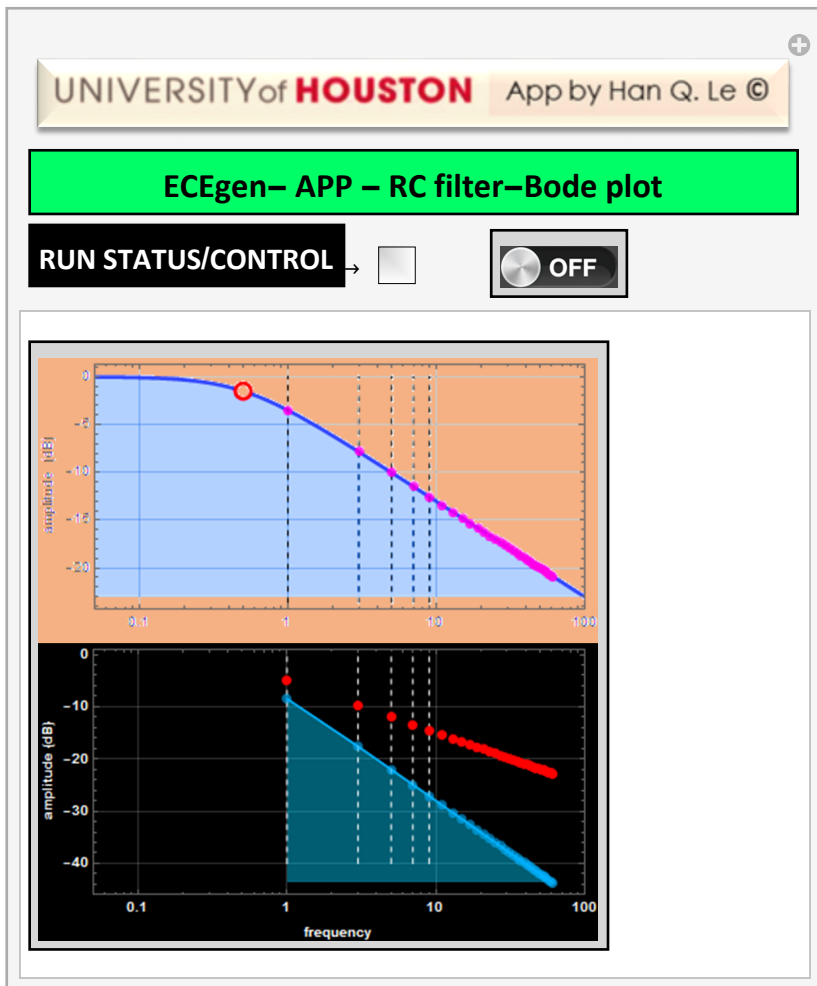


With known system response, we know the output of each component. Remember this?



All we have to do is to add them up! See illustration below

4.1 Illustration of passive RC with Bode plots

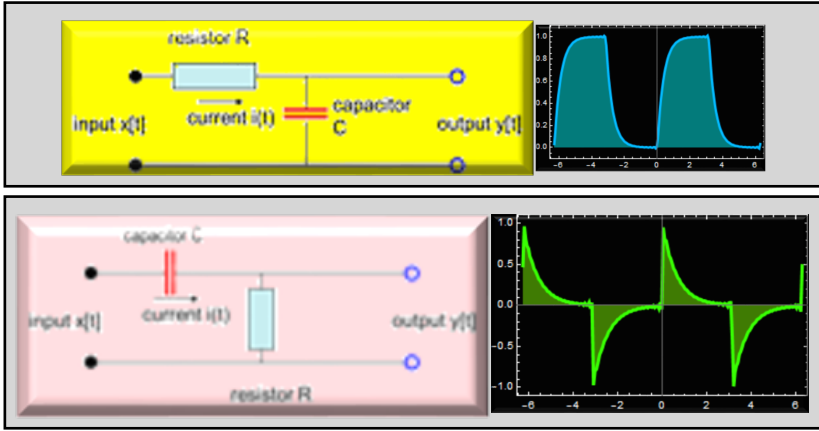


4.2 Illustration of passive RC with Fourier analysis

UNIVERSITY of HOUSTON App by Han Q. Le ©

ECEgen- APP – RC filter

RUN STATUS/CONTROL OFF



The screenshot displays the ECEgen- APP – RC filter interface. At the top, it shows the University of Houston logo and the app name. Below that, there is a green bar with the text "ECEgen- APP – RC filter". Underneath, there is a "RUN STATUS/CONTROL" section with a checkbox and a button labeled "OFF". The main area contains two circuit diagrams and their corresponding plots. The top diagram shows a resistor R in series with a capacitor C, with input x(t) and output y(t). The plot shows a blue step function. The bottom diagram shows a capacitor C in series with a resistor R, with input x(t) and output y(t). The plot shows a green step function.

To be continued to Part 2 - Fourier integral transform