UNIVERSITY OF LINCOLN

CMP9780, EGR3031 & BME3002

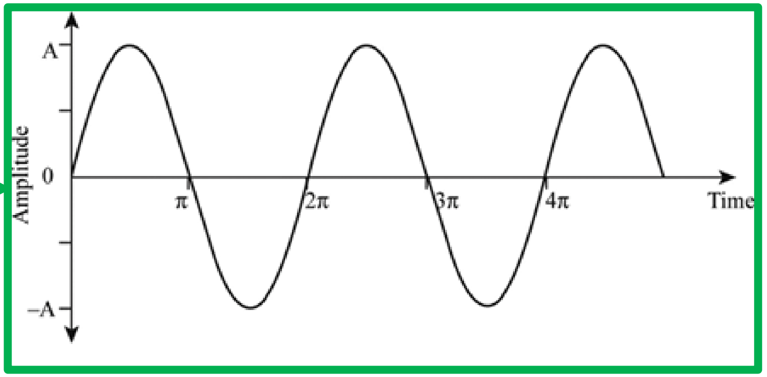
Lecture Week 2 – Introduction to Fourier Transforms

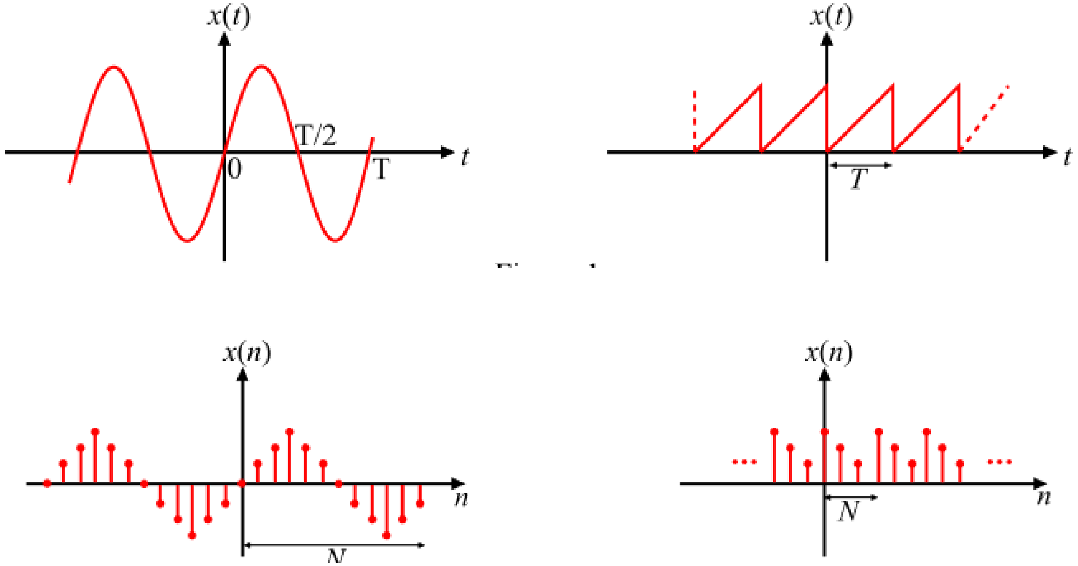
Contact Information

- Dr John Atanbori
  - jatanbori@lincoln.ac.uk
  - Isaac Newton Building, INB2217
  - Office Hours
    - Wed 11:00 – 13:00

Introduction to Fourier

- Introduced by **Joseph Fourier**, a French Mathematician
- Information as a function of time aka signal
- Periodic vs non-periodic signals
  - Periodic -> $F(x+p) = F(x)$, *p* is periodicity of function *F*(*x*).
- The sinusoidal waveform in the figure is repeating over a period of 2π.
  - It can be written as *sin(x+2π)=sin(x)*





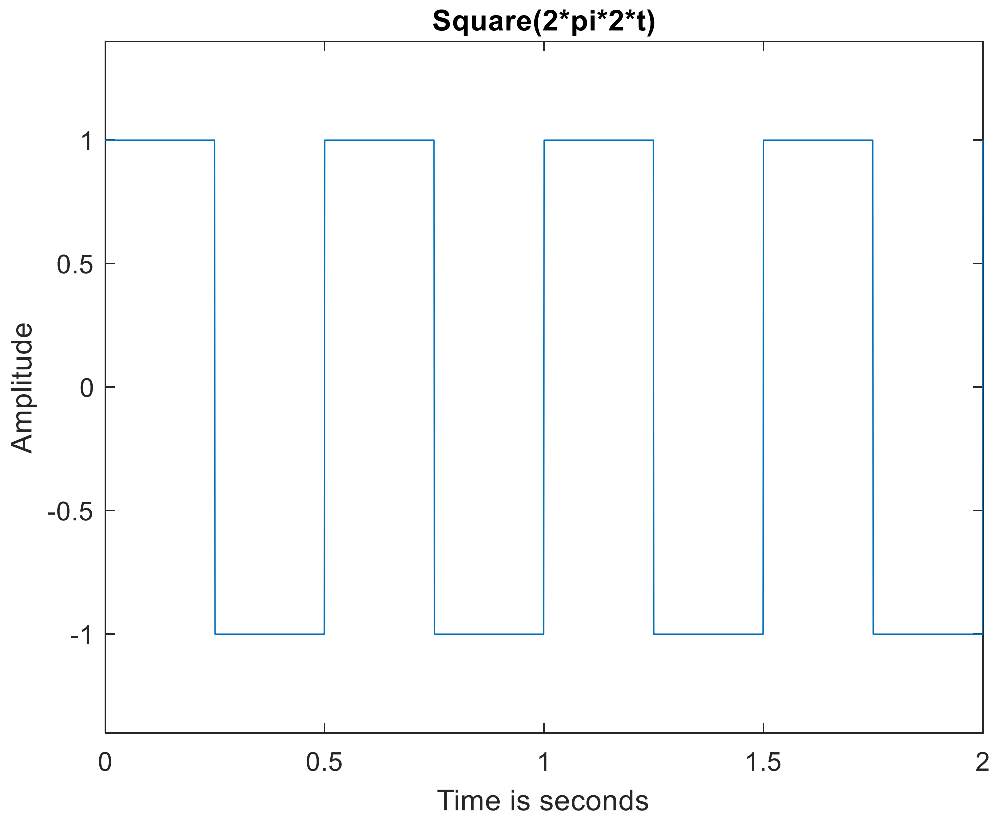
Square Wave

Is the signal periodic? What is the time period?

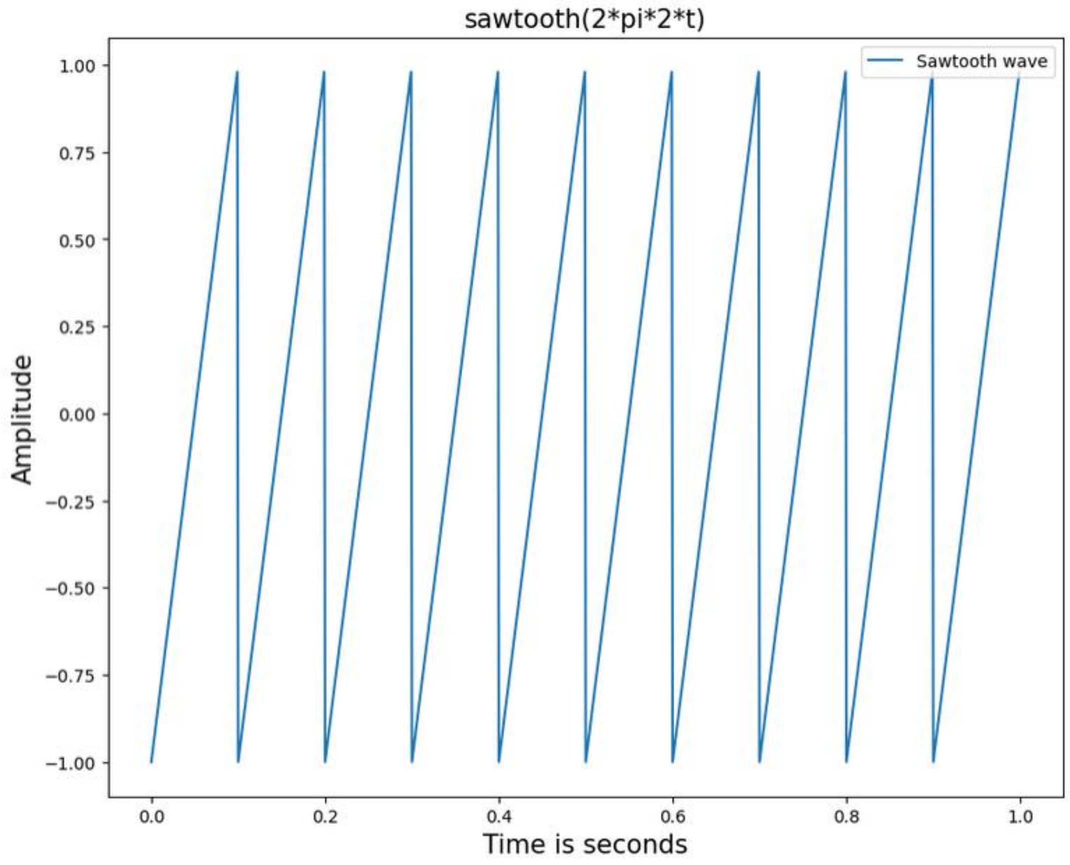
```python
import numpy as np
import matplotlib.pyplot as plt
from scipy import signal
```

```python
fs = 1000
dt = 1/fs
t = np.arange(0, 2, dt)
x = signal.square(2*np.pi*2*t)
```

```python
plt.figure(figsize=(10, 8))
plt.plot(t, x, label='Sine wave')
plt.xlabel('Time is seconds', fontsize=15)
plt.ylabel('Amplitude', fontsize=15)
plt.title('square(2*pi*2*t)', fontsize=15)
plt.legend(fontsize=10, loc='upper right')
```



Sawtooth / Triangular Wave



Is the signal periodic? What is the time period?

```
fs = 1000
dt = 1/fs
t = np.arange(0, 2, dt)
x = signal.sawtooth(2*np.pi*10*t)
```

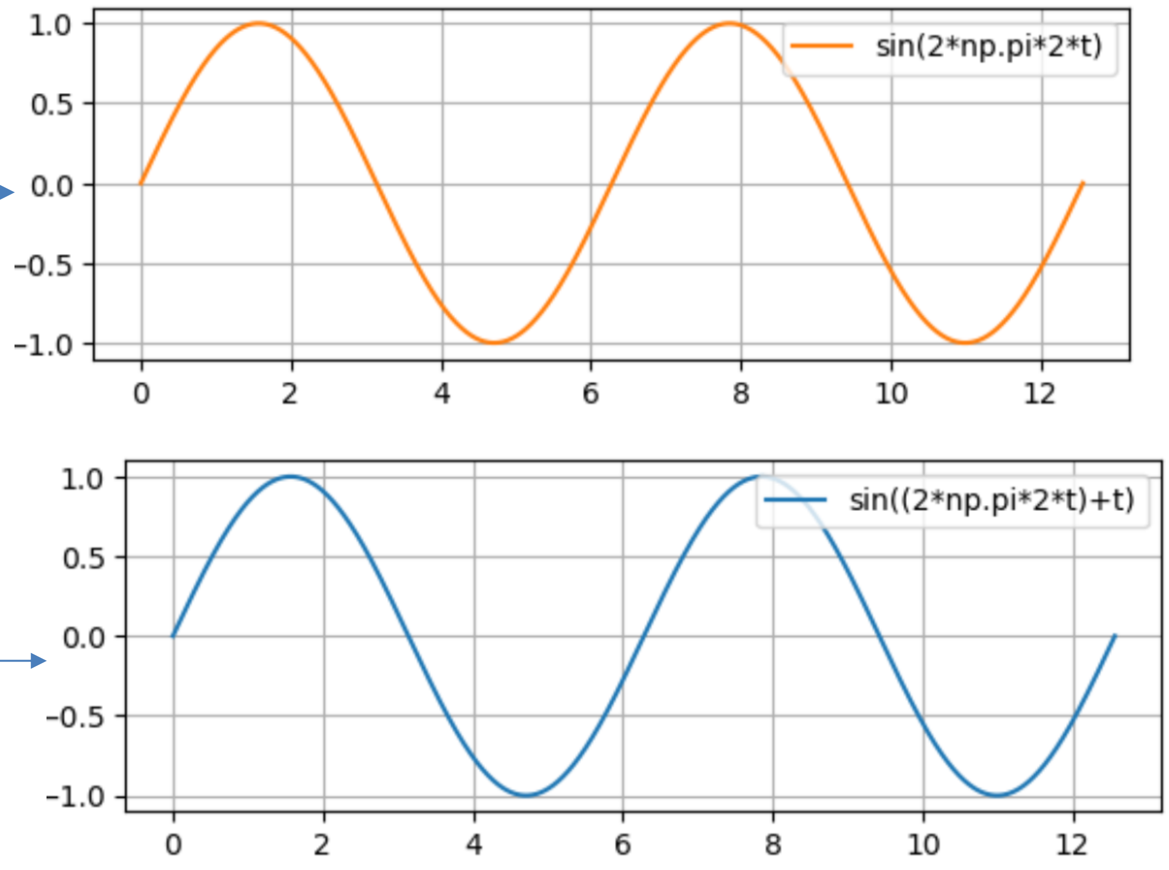
```
plt.figure(figsize=(10, 8))
plt.plot(t[0:1000], x[0:1000], label='Sawtooth wave')
plt.xlabel('Time is seconds', fontsize=15)
plt.ylabel('Amplitude', fontsize=15)
plt.title('sawtooth(2*pi*2*t)', fontsize=15)
plt.legend(fontsize=10, loc='upper right')
```

Periodic Signals

$$x(t + T) = x(t) \text{ for } -\infty < t < \infty$$

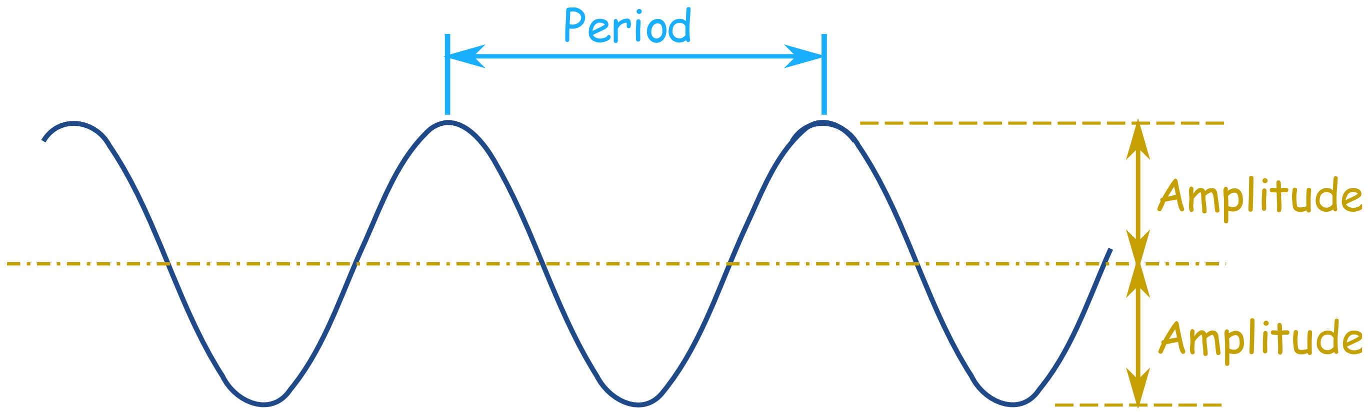
```
fs = 1000
dt = 1/fs
t = np.arange(0, 4*np.pi, dt)

X1 = np.sin(t)
X2 = np.sin((2*np.pi) + t)
```

```
fig, axs = plt.subplots(2)

axs[0].plot(t, X1, 'tab:orange', label='sin(2*np.pi*2*t)')
axs[0].legend(fontsize=10, loc='upper right')
axs[0].grid()

axs[1].plot(t, X2, label='sin((2*np.pi*2*t)+t)')
axs[1].legend(fontsize=10, loc='upper right')
axs[1].grid()
```

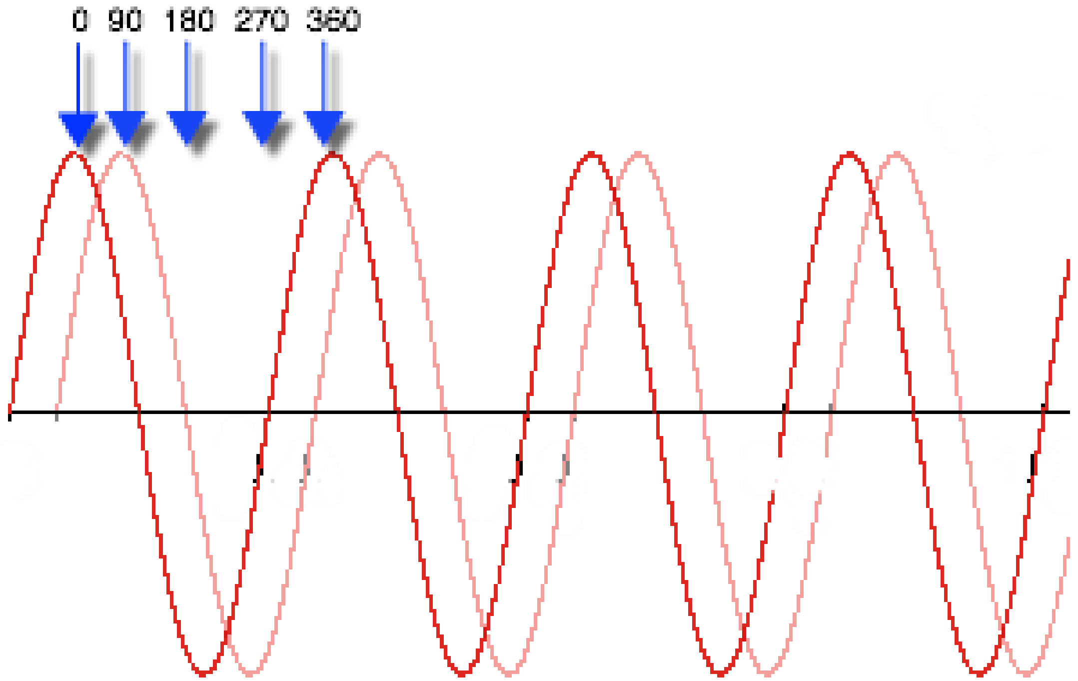


Introduction to Fourier

- Periodic signals repeat themselves at a specific **time** interval.
  – This interval is called the **period** of the signal.
- **Amplitude**
  – "Amplitude is the height, force or power of the wave“
  – Maximum Displacement



Introduction to Fourier



- **Phase**
  - Same **frequency**, and **same cycle**
  - But the wave forms are not exactly aligned together.

Introduction to Fourier

Signal that composes of **multiple sinusoids**, **amplitudes** and **phases**

$$x(t) = A_0 + \sum_{k=1}^{N} A_k \cos(2\pi f_k t + \phi_k)$$

$$x(t) = A_0 + \sum_{k=1}^{N} A_k \cos(2\pi f_k t + \phi_k)$$

Remember these relations

$$\sin\theta = \cos(90 - \theta)$$

$$\cos\theta = \sin(90 - \theta)$$

Introduction to Fourier

Fourier In complex form

$$x(t) = A_0 + \sum_{k=1}^{N} A_k \cos(2\pi f_k t + \phi_k)$$

$$= X_0 + \mathrm{Re}\left\{ \sum_{k=1}^{N} X_k e^{j2\pi f_k t} \right\}$$

where here $X_0 = A_0$ is real, $X_k = A_k e^{j\phi_k}$ is complex, and f_k is the frequency in Hz

Introduction to Fourier

- Euler's formula, by renowned mathematician Leonhard Euler
- **Euler's formula demonstrates the fundamental relationship between the <u>trigonometric functions</u> and the <u>complex exponential function</u>.**
- Euler's formula states that for any real number x:
  - $e^{ix} = cosx + isinx$
  - $e^{-ix} = cosx - isinx$
- Hence:
  - $cosx = \frac{e^{ix} + e^{-ix}}{2} = \frac{1}{2}\, e^{ix} + \frac{1}{2}\, e^{-ix}$
- and:
  - $sinx = \frac{e^{ix} - e^{-ix}}{2i} = -\frac{1}{2}\, ie^{ix} + \frac{1}{2}\, ie^{-ix}$

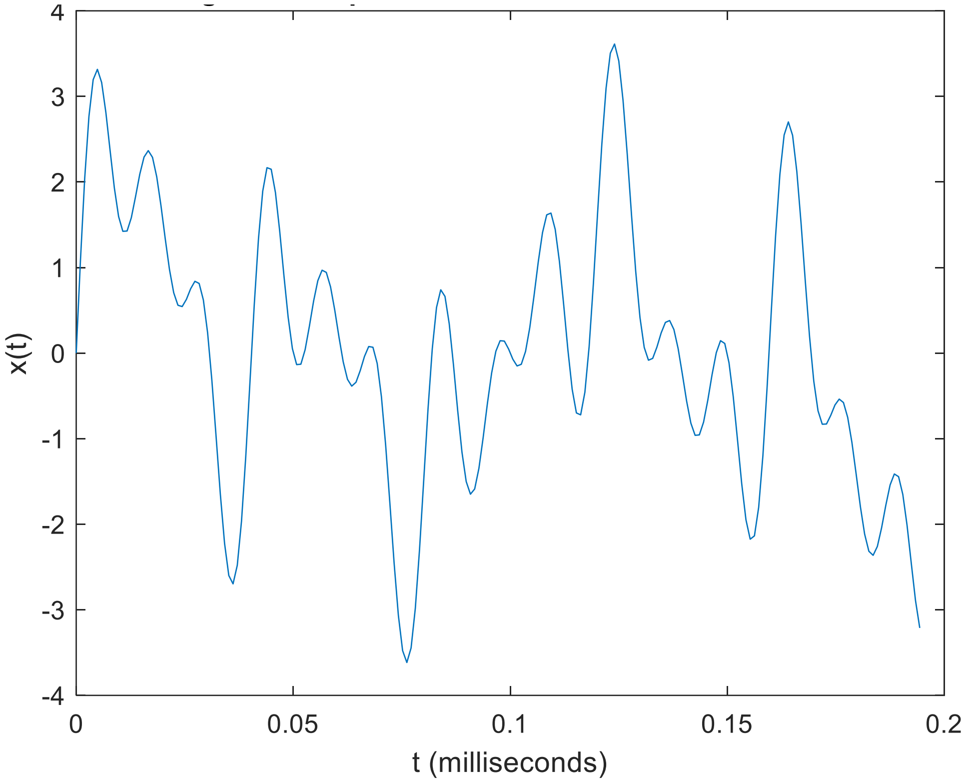
- **Maths might become simpler if we use e^{ix} instead of *cosx* and *sinx***

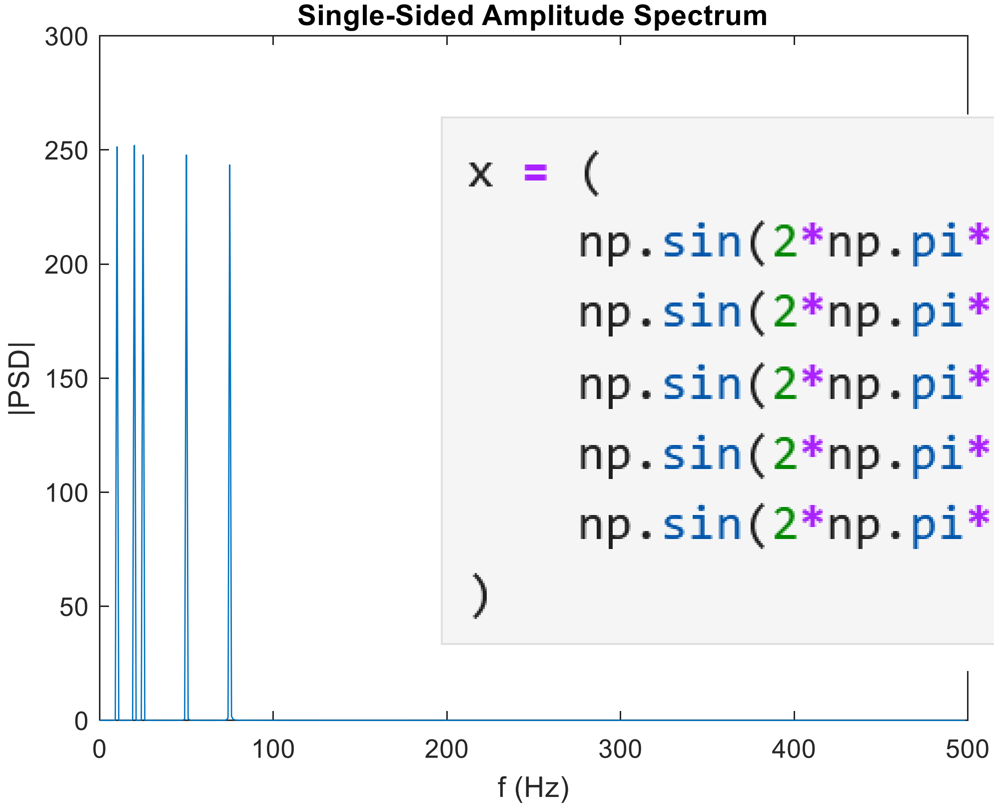
Introduction to Fourier

- **Signal Spectra**
  - By Fourier theory, any waveform can be represented by a summation of a (possibly infinite) number of sinusoids, each with a particular amplitude and phase.
  - Such a representation is referred to as the **signal's spectrum** (or it's frequency-domain representation).

Summation of 5 Sine Waves

- The signal (Left) and Spectrum (right)

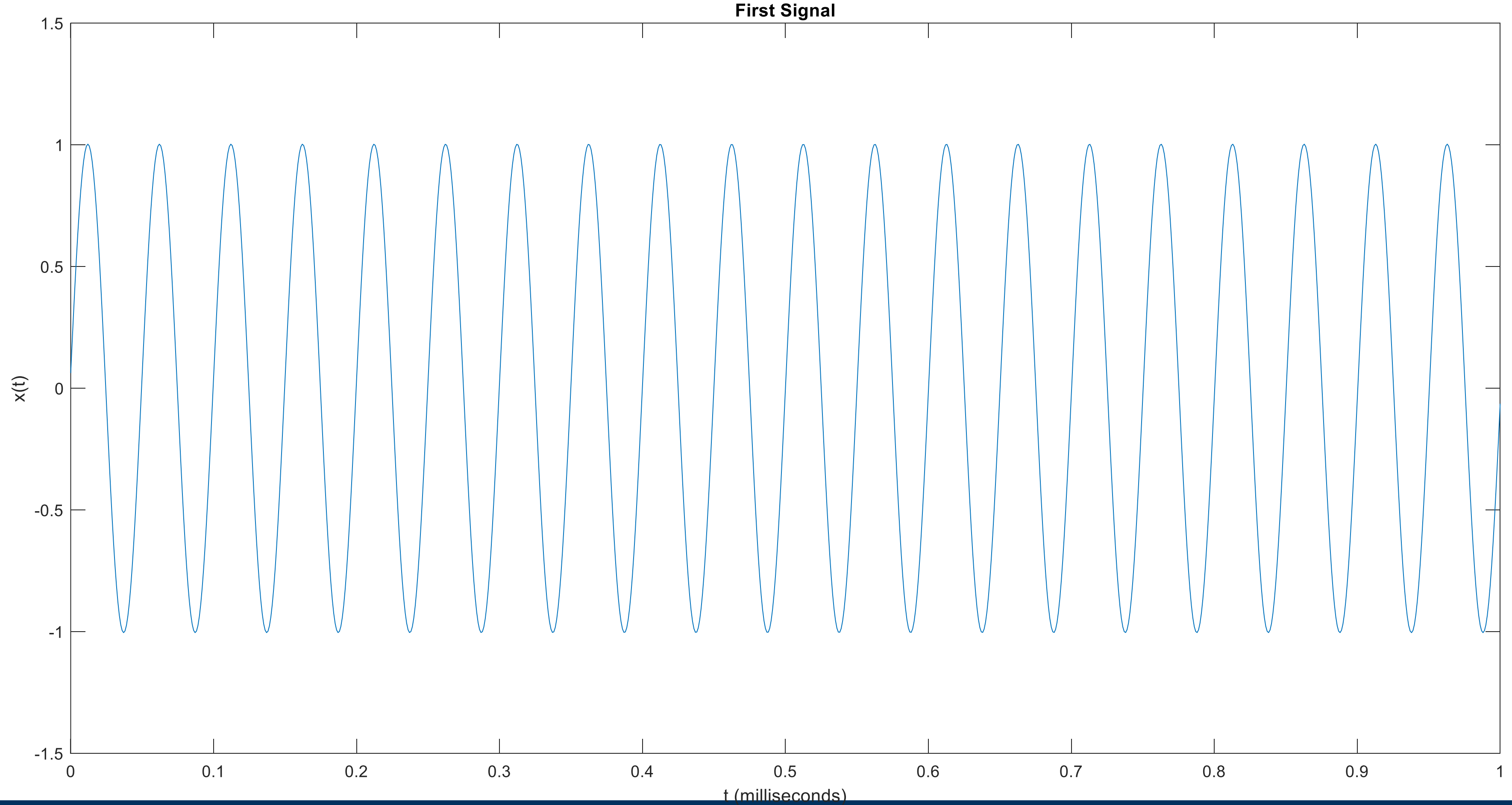





```python
x = (
    np.sin(2*np.pi*10*t) +
    np.sin(2*np.pi*20*t) +
    np.sin(2*np.pi*25*t) +
    np.sin(2*np.pi*50*t) +
    np.sin(2*np.pi*75*t)
)
```

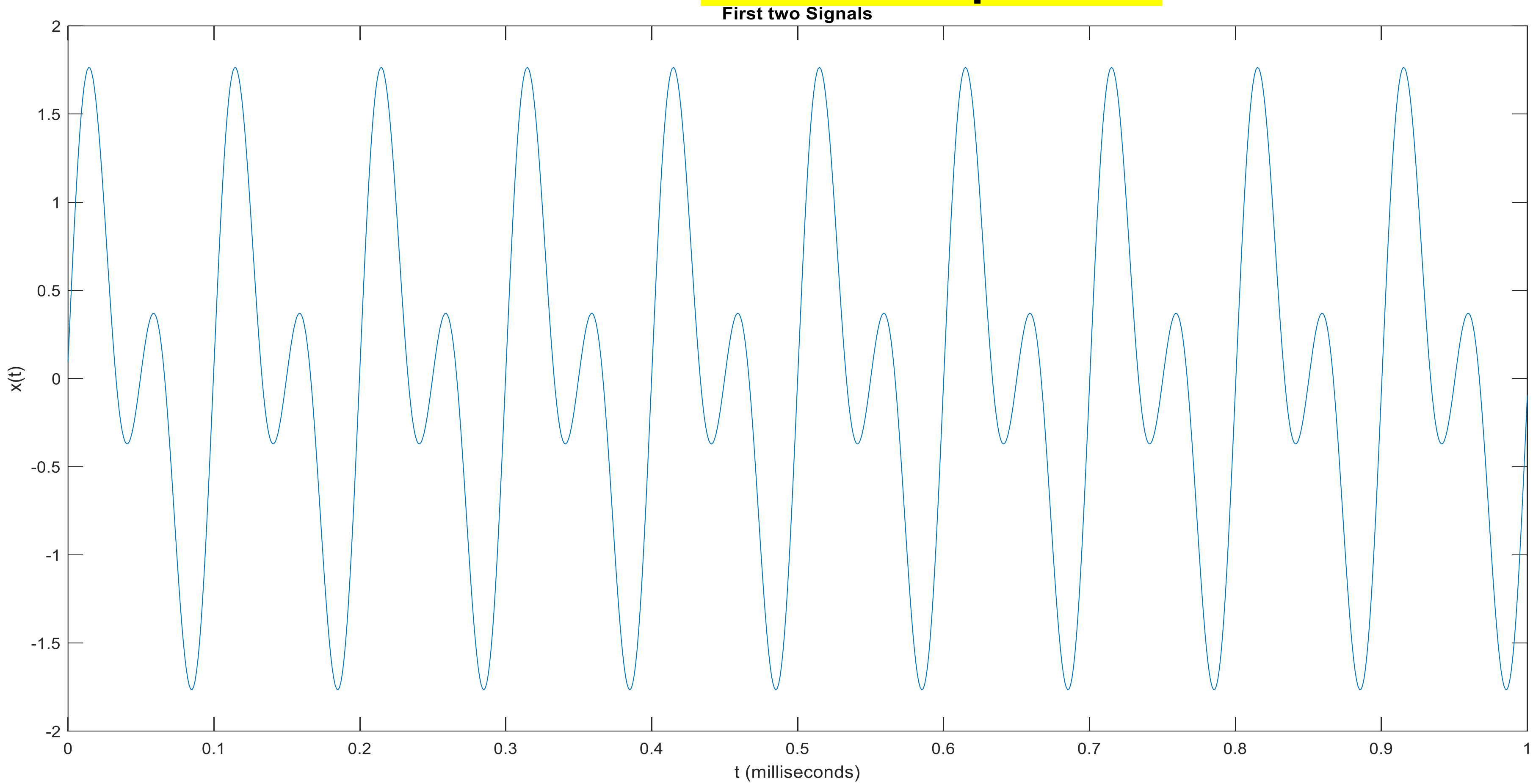
Summation of 5 Sine Waves

- Signal reconstructed with first peak



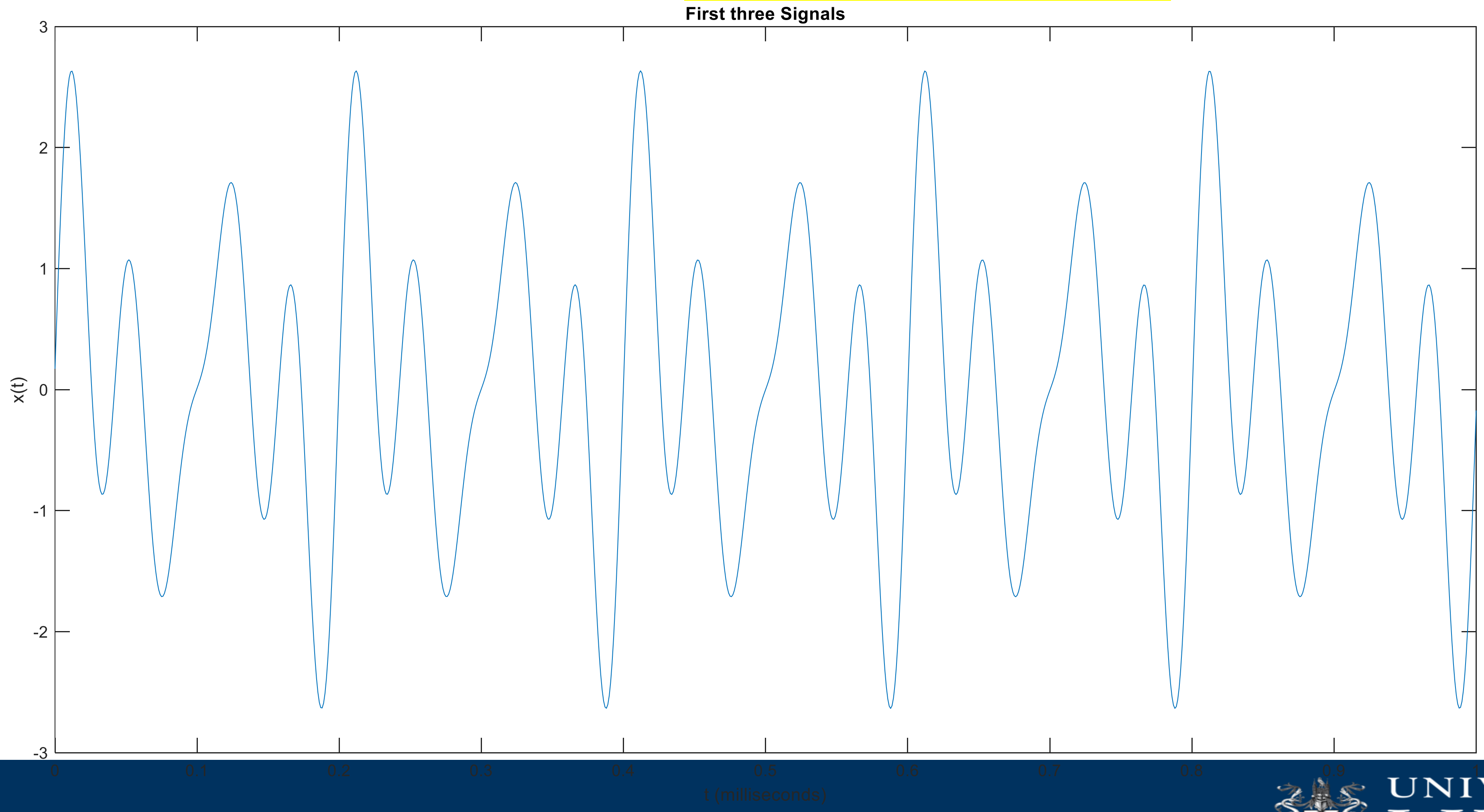
Summation of 5 Sine Waves

- Signal reconstructed with first two peaks



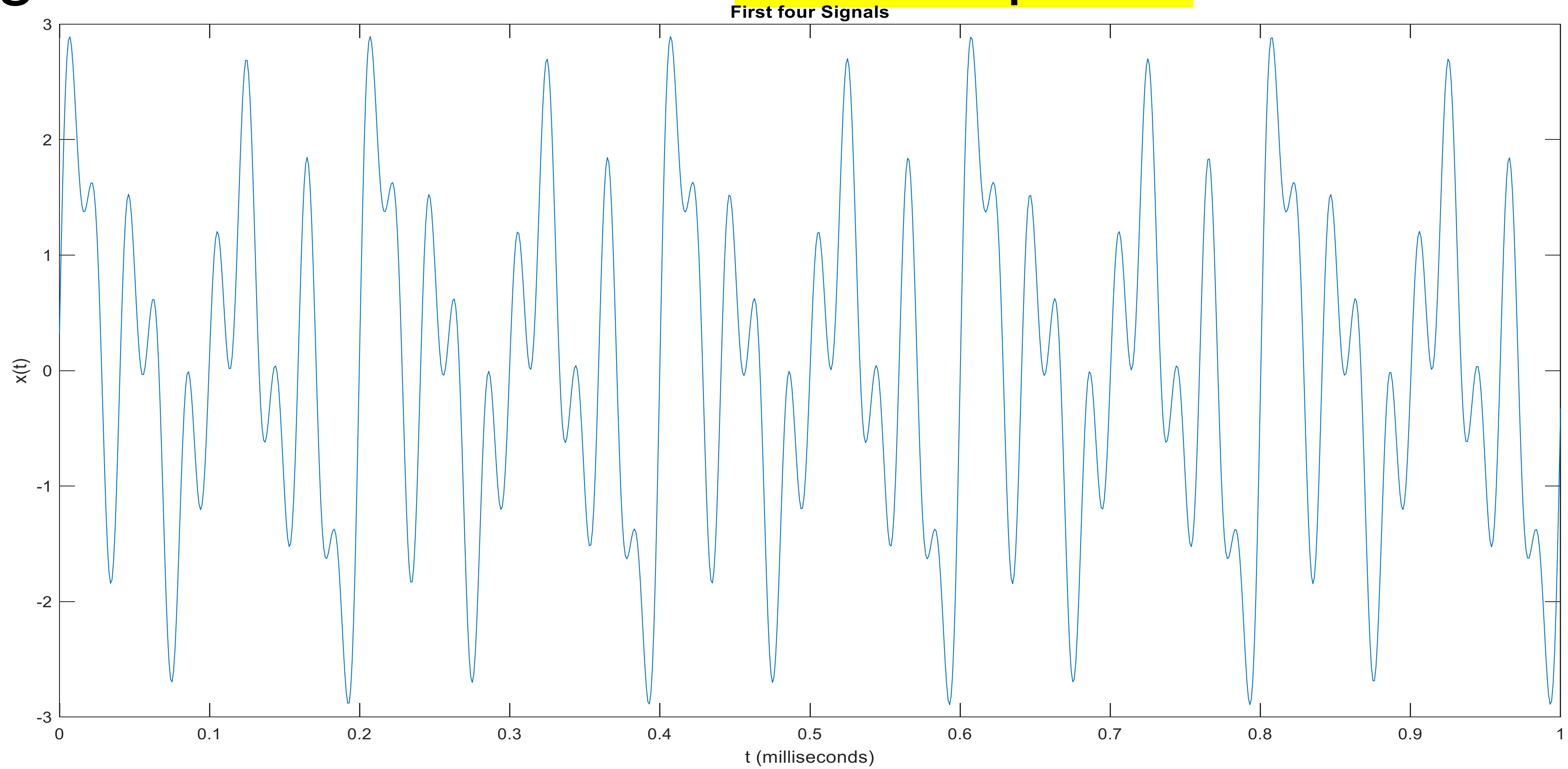
Summation of 5 Sine Waves

- Signal reconstructed with first three peaks



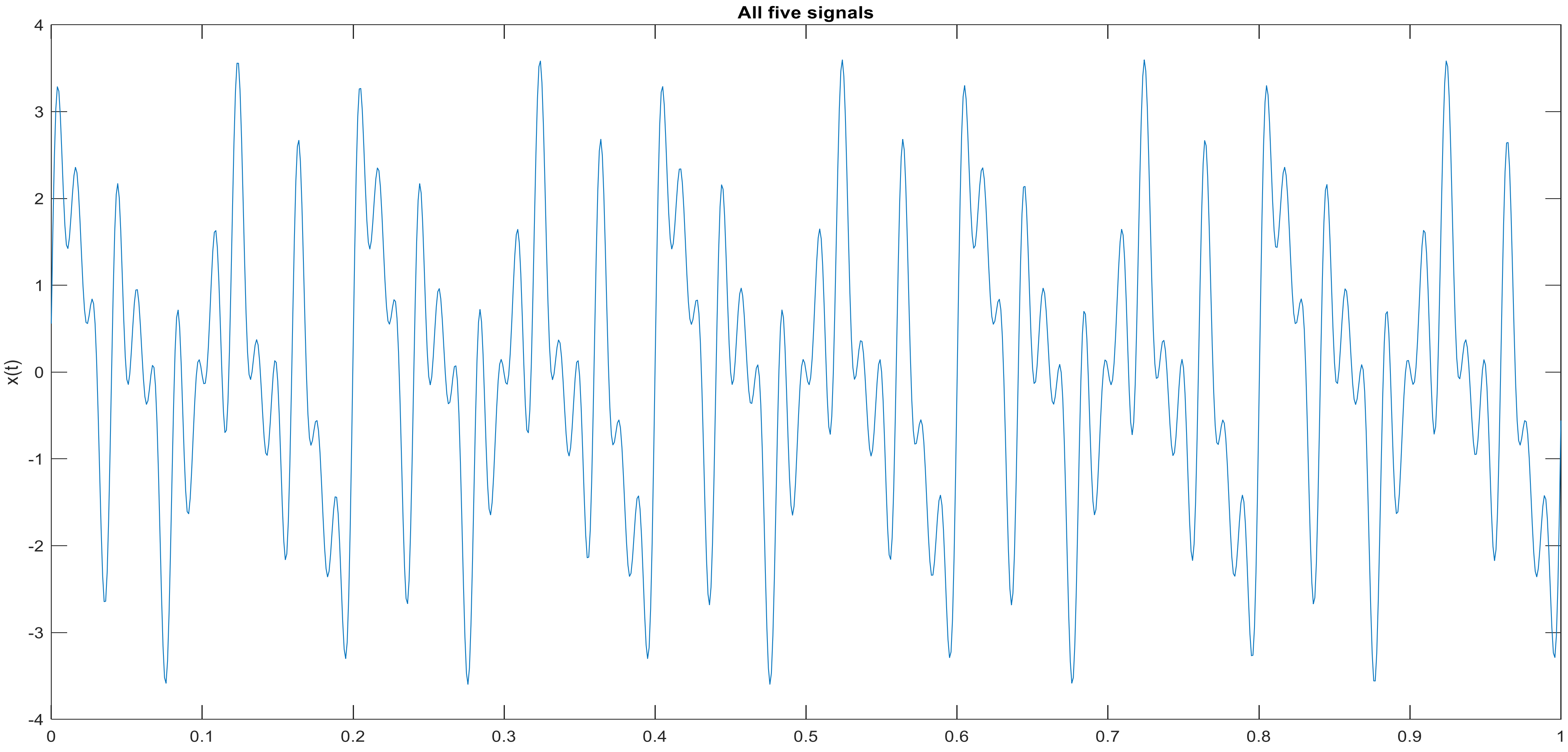
Summation of 5 Sine Waves

- Signal reconstructed with first four peaks



Summation of 5 Sine Waves

- Signal reconstructed with all five peaks

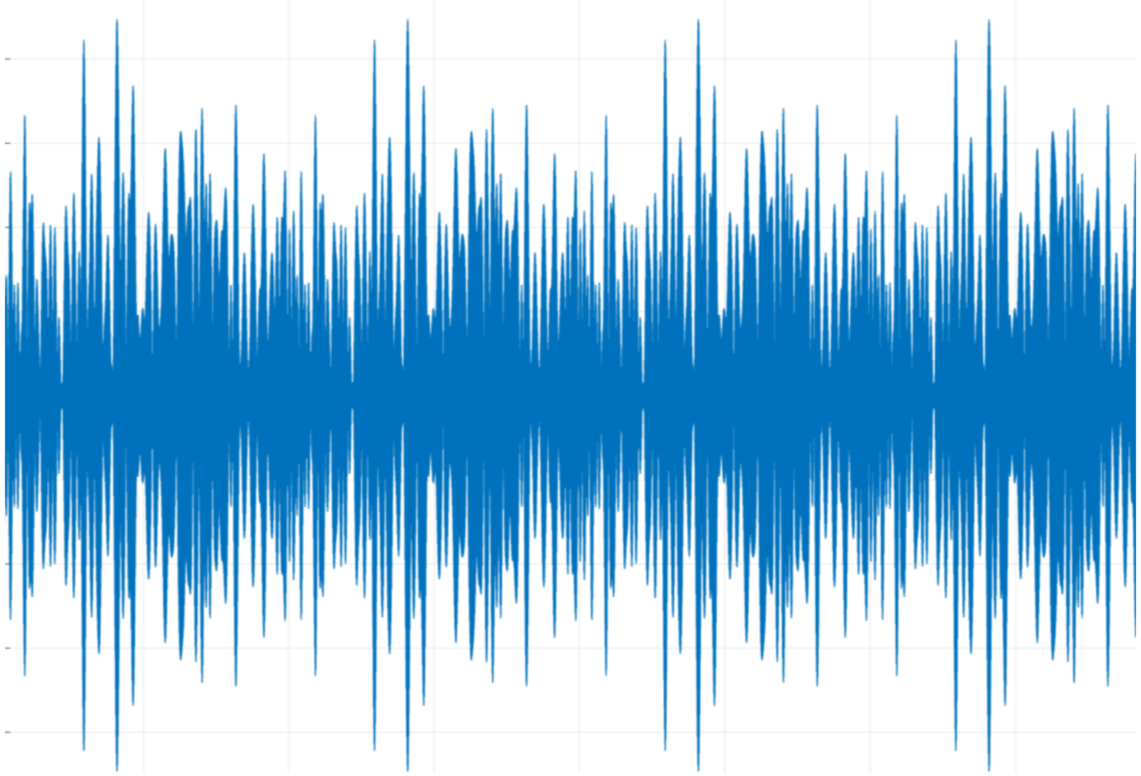


Question

- Since the different peaks can be used to reconstruct a signal, what conclusions can be drawn from the previous demonstrations regarding signal noise and compression?

Introduction to Fourier

- Many physical systems that resonate or oscillate produce quasi-sinusoidal motion
- The speech signal can be decomposed into multiple **sinusoids**
- **In short, most signals can be decomposed into sinusoidal components, and the <u>sum of the individual components</u> forms the same signal.**



Introduction to Fourier

Time schedule:

- Monday: put trash outside (plus do your homework)
- Tuesday: clean kitchen (plus do your homework)
- Wednesday: put trash outside (plus do your homework)
- Thursday: clean your bedroom (plus do your homework)
- Friday: put trash outside (plus do your homework)
- Saturday: clean your bedroom (plus do your homework)
- Sunday: (do your homework)

Spectral schedule:

- Always: do your homework
- Once a week: clean kitchen
- Twice a week: clean your bedroom
- Three times a week: put trash outside

Time Dimension

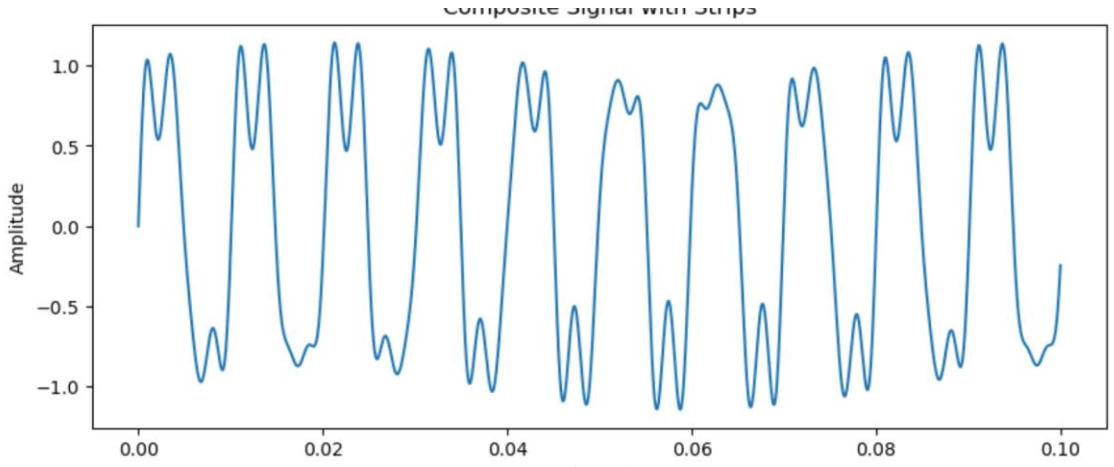
Instead of looking how these patterns are scheduled over time, we can also order these activities, in the frequency they occur

Frequency Dimension, losing the time dimension

Introduction to Fourier

Are these signals periodic?

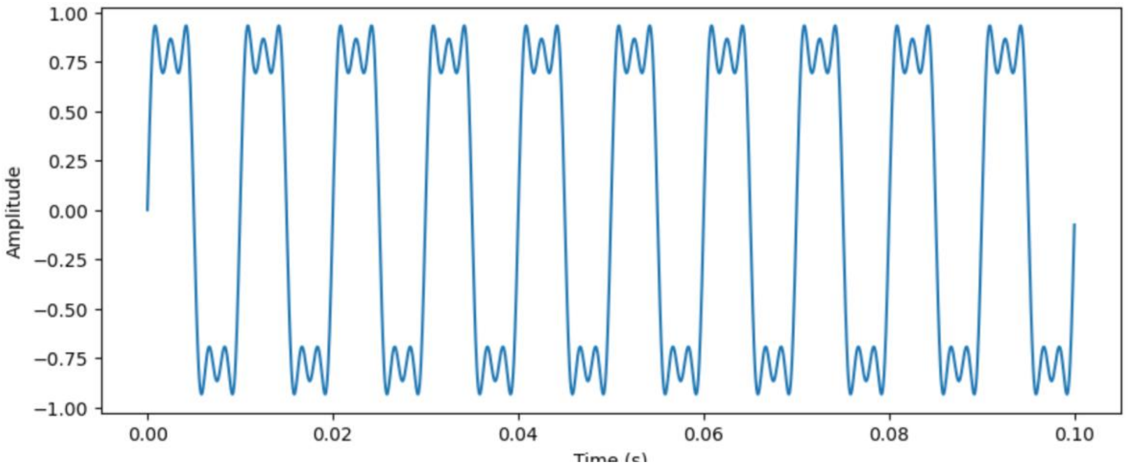
A




```python
# Define the sampling frequency and time array
fs = 50 * 500  # Sampling frequency
t = np.arange(0, 0.1, 1/fs)  # Time from 0 to 0.1 seconds w
```

```python
x_nper = (np.sin(2 * np.pi * 100 * t) +
          (1/3) * np.sin(2 * np.pi * np.sqrt(89999) * t) +
          (1/5) * np.sin(2 * np.pi * np.sqrt(149999) * t))
```

B

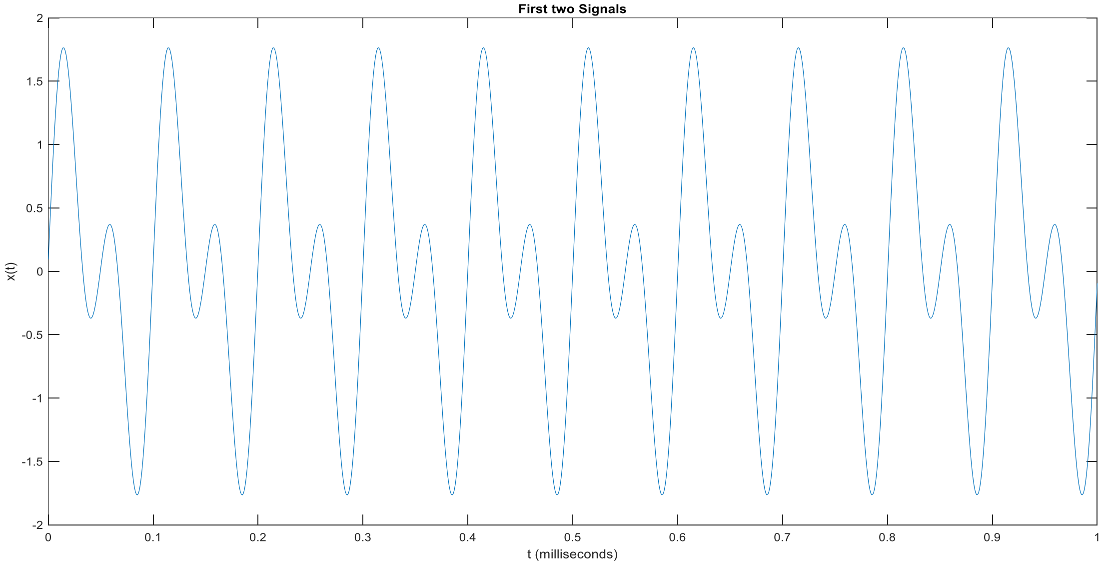



```python
# Define time array
fs = 50 * 500  # Sampling frequency
t = np.arange(0, 0.1, 1/fs)  # Time from 0 to 0.1 sec
```

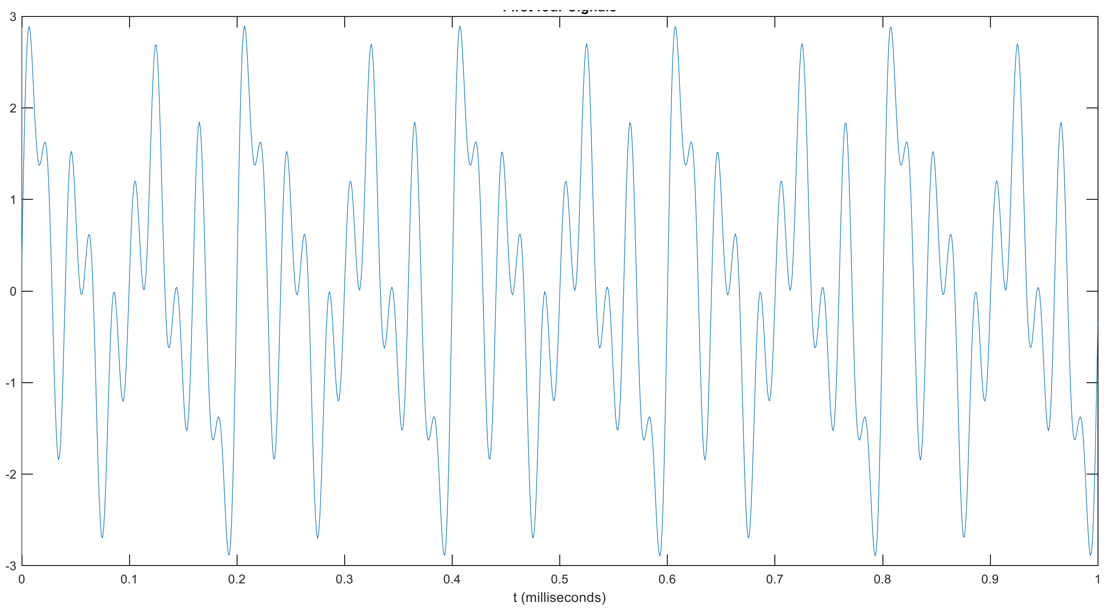
```python
# Define the periodic signal
x_per = (np.sin(2 * np.pi * 100 * t) +
         (1/3) * np.sin(2 * np.pi * 300 * t) +
         (1/5) * np.sin(2 * np.pi * 500 * t))
```

Introduction to Fourier

- **y(t)** = sin(2*pi*10*t) + sin(2*pi*20*t)

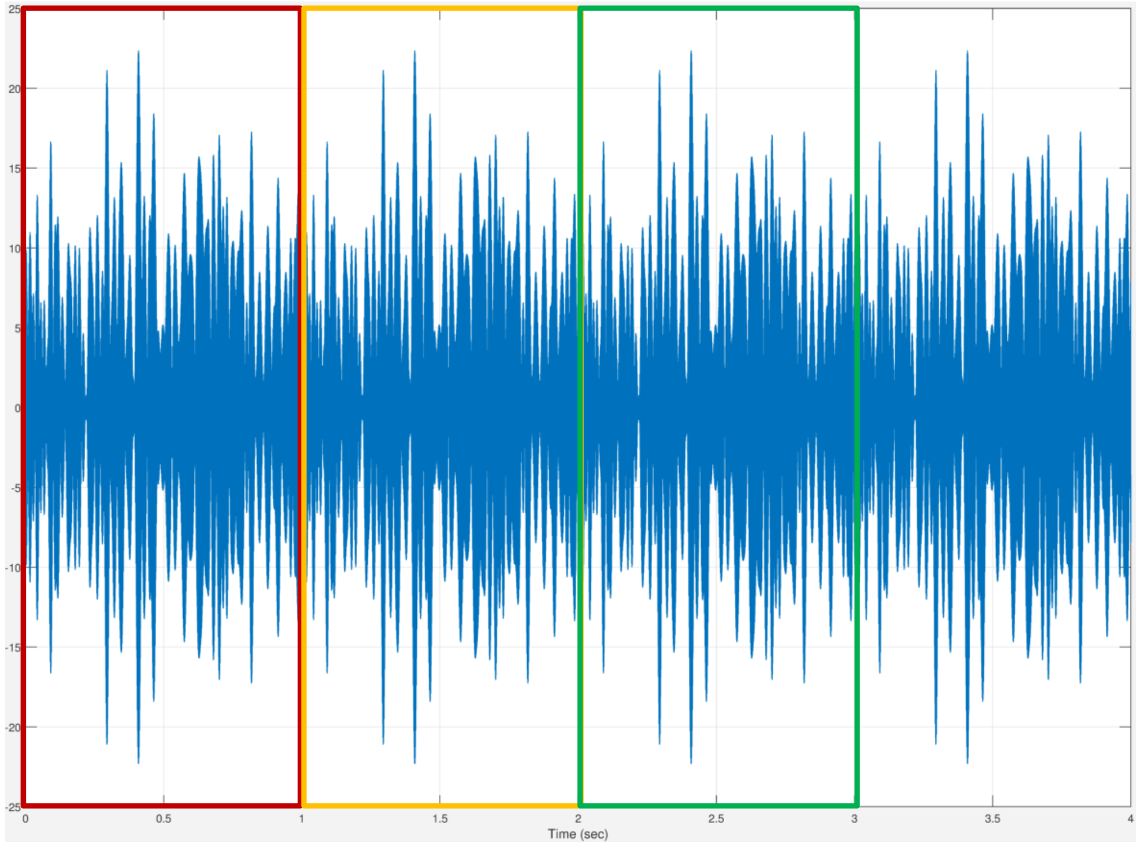


- **x(t)** = sin(2*pi*10*t) + sin(2*pi*20*t) + sin(2*pi*25*t) + sin(2*pi*50*t) + + sin(2*pi*75*t);



Are these signals periodic?

Introduction to Fourier



- **What about this signal? Periodic or Non-Period?**
- Sometimes it is all but impossible to figure out the period from a graph
- The signal below has a period of 1 second

Introduction to Fourier

- The mathematical definition of a signal is given below.

$$x(t) = \cos(2\pi 100t) + \sin(2\pi 150t)$$

- Check if the signal is periodic at 40ms

$$x(t+0.04) = \cos(2\pi 100(t+0.04)) + \sin(2\pi 150(t+0.04))$$
$$x(t+0.04) = \cos(2\pi 100t + 2\pi 100(0.04)) + \sin(2\pi 100t + 2\pi 100(0.04))$$
$$x(t+0.04) = \cos(2\pi 100t + 2\pi 4)) + \sin(2\pi 100t + 2\pi 6)$$
$$x(t+0.04) = \cos(2\pi 100t)) + \sin(2\pi 100t)$$
$$x(t+0.04) = x(t)$$

Fourier Series

- In the study of Fourier series, we learn how any **periodic continuous signal** can be represented as a **sum of harmonically related sinusoids**

The **synthesis formula** is:

$$x(t) = \sum_{k=-\infty}^{\infty} a_k e^{j(2\pi/T_0)kt}$$

where T_0 is the period

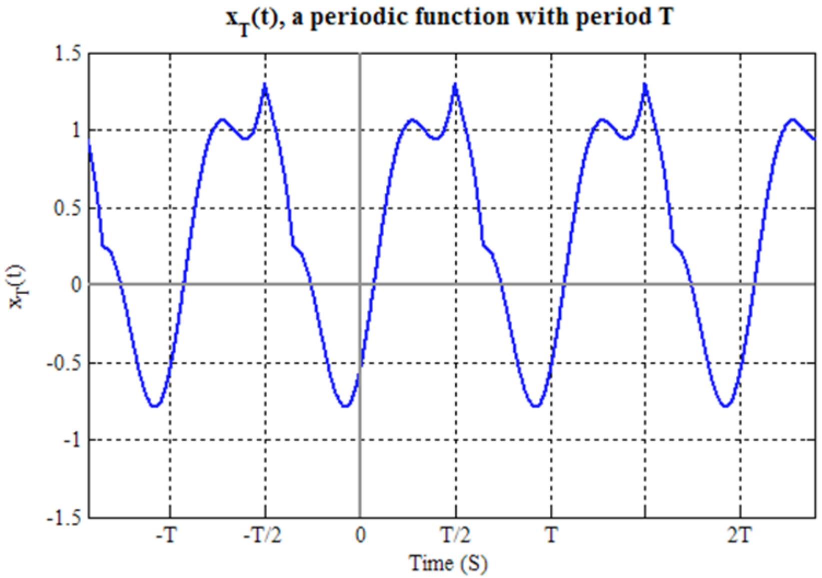
Fourier *coefficient* /
Complex-valued
coefficient ➔ **amplitude**
and phase combined

A Fourier series is an expansion of a periodic function f(x) in terms of an infinite sum of sines and cosines.

It <u>**cannot**</u> represent any arbitrary function.

It can represent either**:**
(a) a periodic function, or
(b) a function that is defined over a finite-length interval only;
values produced by the Fourier series outside the finite interval are irrelevant.

Fourier Series



$$\boxed{x(t) = a_0 + \sum_{n=1}^{\infty} \left(a_n \cos(n\omega_0 t) + b_n \sin(n\omega_0 t)\right)} \quad \textit{Trigonometric}$$

$$= \boxed{\sum_{n=-\infty}^{\infty} c_n e^{jn\omega_0 t}} \quad \textit{Exponential}$$

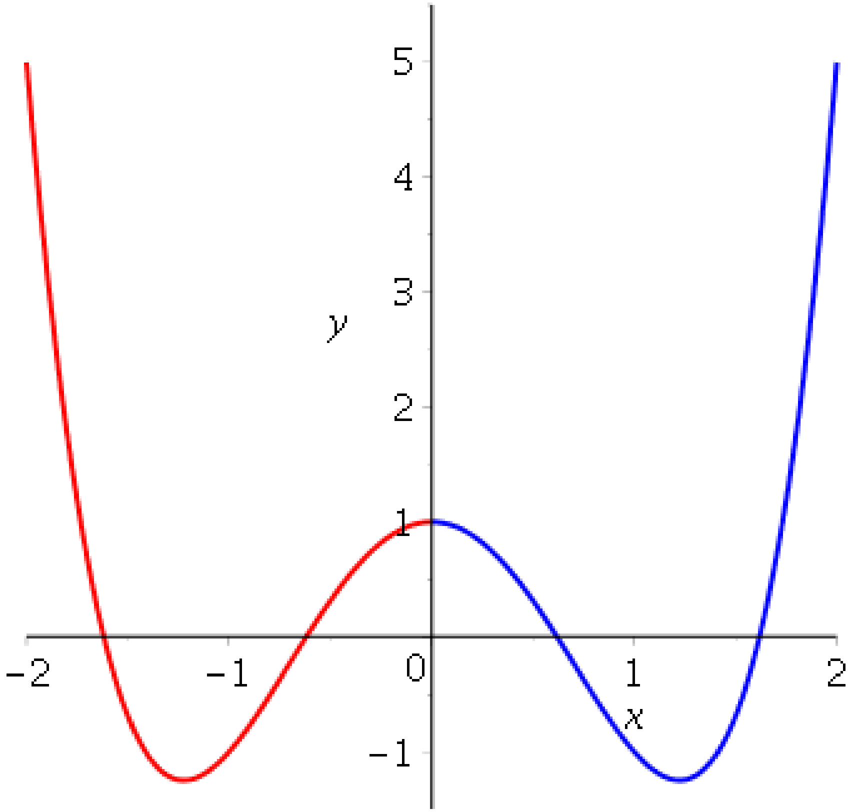
- Work by **Joseph Fourier** in the **early 1800's -** Fourier Series.
  - Determine that such a function can be represented as a series of sines and cosines.
  - In other words, as sum of sines and cosines of different frequencies, called a Fourier Series.
  - **There are two common forms** of the Fourier Series, "Trigonometric" and "Exponential." For easy reference the two forms are stated here.

Fourier Series

- A function is called even if $f(-x)=f(x)$, e.g. $\cos(x)$.
- A function is called odd if $f(-x)=-f(x)$, e.g. $\sin(x)$.

- These have somewhat different properties than the even and odd numbers:
  - Sum: Even + Even = **Even**, and Odd + Odd = **Odd**
  - Product: Even × Even = **Even**; Odd × Odd = **Even**; and Even × Odd = **Odd**

Question



- Determine from the graph whether the plotted function is an EVEN or ODD function.

a) Even
b) Odd

Fourier Series

The Fourier series expansion of an even function $f(x)$ with the period of 2π does not involve the terms with sines and has the form:

$$\boxed{f(x) = \frac{a_0}{2} + \sum_{n=1}^{\infty} a_n \cos nx,}$$

where the Fourier coefficients are given by the formulas

$$\boxed{a_0 = \frac{2}{\pi} \int_0^{\pi} f(x)\, dx,} \quad \boxed{a_n = \frac{2}{\pi} \int_0^{\pi} f(x) \cos nx dx.}$$

Fourier Series

Accordingly, the Fourier series expansion of an odd 2π-periodic function $f(x)$ consists of sine terms only and has the form:

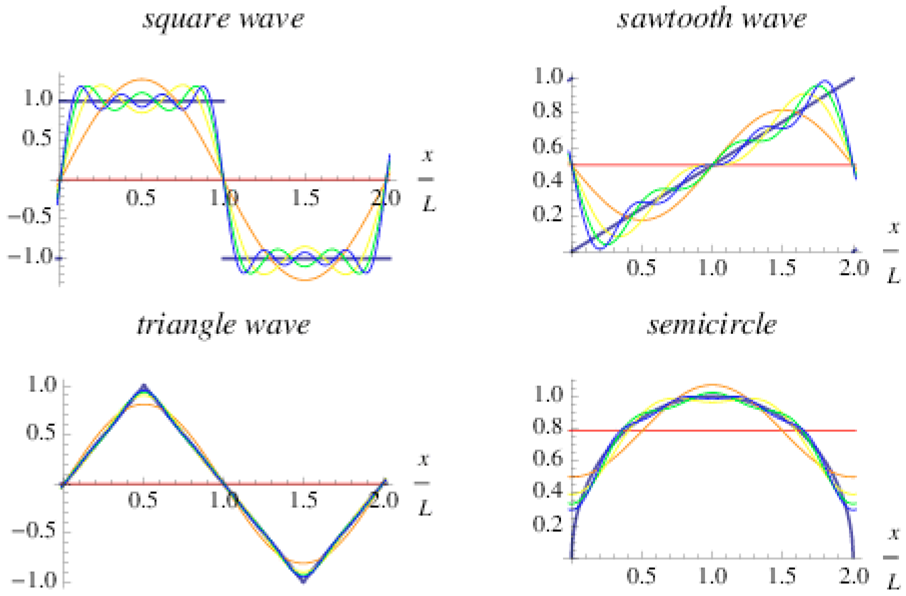
$$\boxed{f(x) = \sum_{n=1}^{\infty} b_n \sin nx,}$$

where the coefficients b_n are

$$\boxed{b_n = \frac{2}{\pi} \int_0^{\pi} f(x) \sin nx dx.}$$

Fourier Series

- The computation and study of Fourier series is known as **harmonic analysis**.
- **Harmonic analysis**
  - break up an arbitrary periodic function into a set of simple terms
  - Solved individually and then recombined to obtain the solution to the original problem or an approximation.
- Examples of successive approximations to common functions using Fourier series are illustrated next.

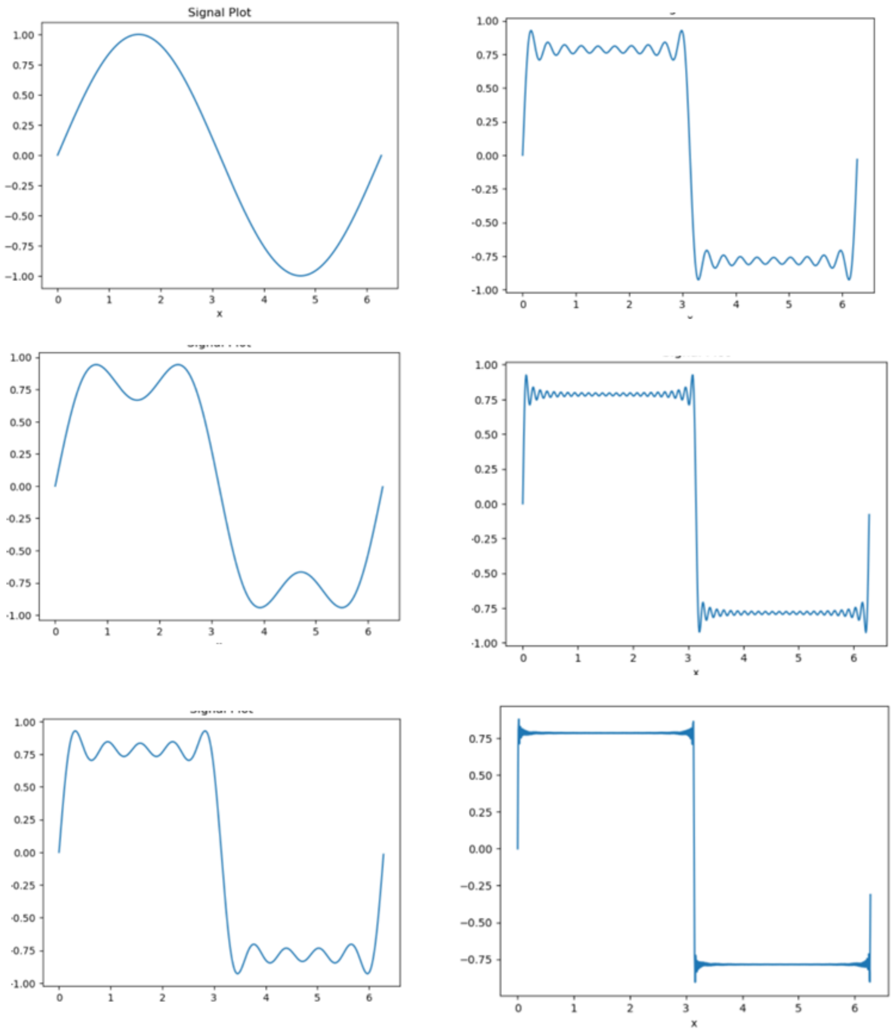


Square Wave Approximation Demo

N=2 N=20

N=4 N=50

N=10 N=200




```python
import numpy as np
import matplotlib.pyplot as plt

x = np.arange(0, 2 * np.pi, 0.01)
signal = np.zeros_like(x)

N = 9
for n in range(1, N, 2):
    y = (1/n) * np.sin(n * x)
    signal += y

plt.plot(x, signal)
plt.xlabel('x')
plt.ylabel('Signal')
plt.title('Signal Plot')
```

Fourier Series

To obtain a **Fourier series representation of periodic signal** *x(t)* we need to evaluate the Fourier integral

$$a_k = \frac{1}{T_0}\int_0^{T_0} x(t)e^{-j(2\pi/T_0)kt}\,dt$$

Where T_0 is the fundamental[3] period

[3]The **fundamental period** is the smallest positive real number T_0 for which the periodic equation $f(t+T) = f(t)$ holds true.

Discrete Fourier Transform

- **DFT is the equivalent of the Continuous Fourier Transform**
  - It has N samples, which are separated by sample time T
  - Finite sequence of data
- Let f(t) be the continuous signal
- Let the N samples be
  - f[0], f[1], f[2], … ,f[k], … , f[N-1]
- The Fourier transform of the original signal f(t) (**continuous**) is
- $$F(j\omega) = \int_{-\infty}^{\infty} f(t)e^{-j\omega t}dt$$

- We could regard each sample f[k] as an impulse having area f[k] .
- Then, since the integrand exists only at the sample points:

$$\int_{o}^{(N-1)T} f(t)e^{-j\omega t}dt$$

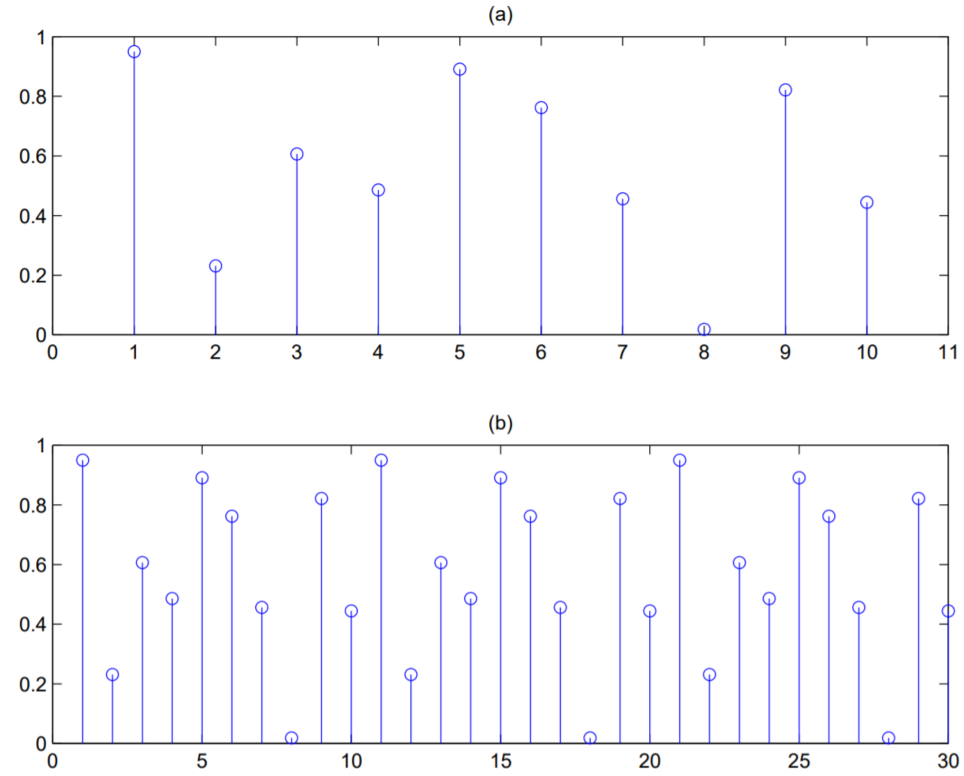
$$f[0]e^{-j0} + f[1]e^{-j\omega T} + \ldots + f[k]e^{-j\omega kT} + \ldots f(N-1)e^{-j\omega(N-1)T}$$

$$F(j\omega) = \sum_{k=0}^{N-1} f[k]e^{-j\omega kT}$$

Discrete Fourier Transform

- Since there are only a **finite number of input data points**, the DFT treats the data as if they were periodic (i.e. *f (N)* to *f (2N-1)* is the same as *f (0)* to *f (N-1))*.

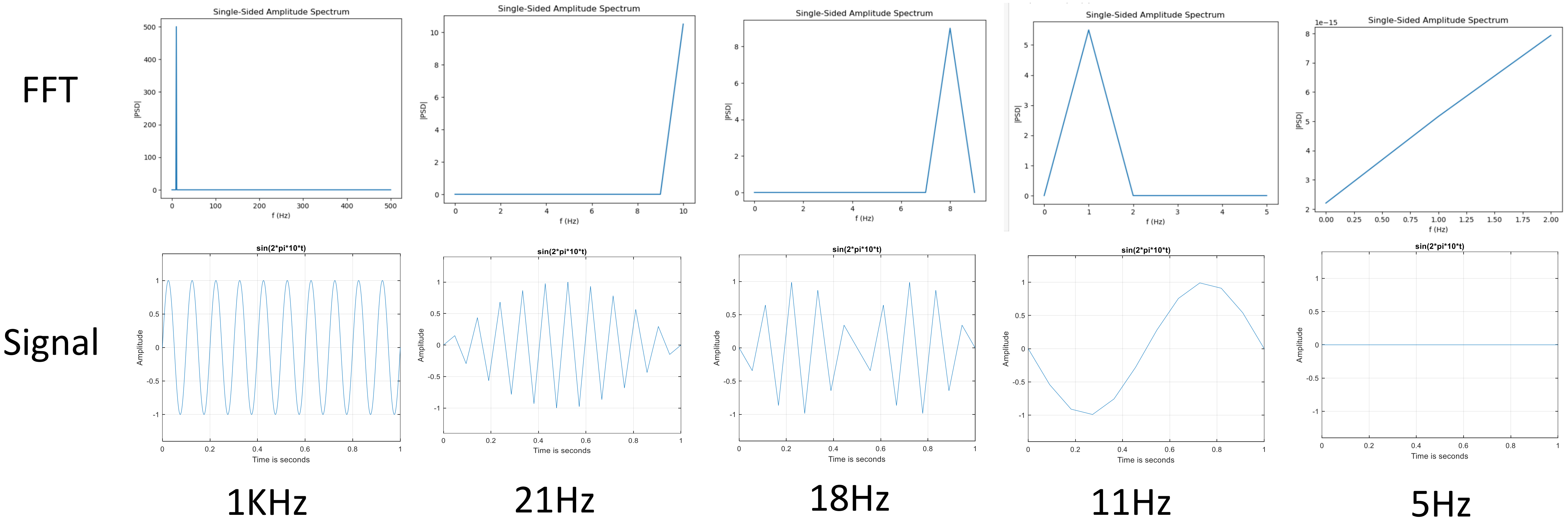
- **The Nyquist criterion is also important in DFT analysis**. When sampling at frequency fs, we obtain reliable frequency information only for frequencies less than fs/2. (Here, reliable means without aliasing problems.)



Plot (a) is considered to be one period (in a sequence of N = 10 samples) of the periodic sequence in plot (b) (implicit periodicity in DFT)

The 10Hz Sine Wave and its FFT

- Examined the characteristics of a 10 Hz sine wave sampled at frequencies of 1 kHz, 21 Hz, 18 Hz, 11 Hz, and 5 Hz on the frequencies in the FFT domain..



Discrete Fourier Transform

- In general

$$F[n] = \sum_{k=0}^{N-1} f[k] e^{-j\frac{2\pi}{N}nk} \qquad (n = 0 : N-1)$$

  - F[n] is the Discrete Fourier Transform of the sequence f[k]

Discrete Fourier Transform

We may write this equation in matrix form as:

$$\begin{pmatrix} F[0] \\ F[1] \\ F[2] \\ \vdots \\ F[N-1] \end{pmatrix} = \begin{pmatrix} 1 & 1 & 1 & 1 & \dots & 1 \\ 1 & W & W^2 & W^3 & \dots & W^{N-1} \\ 1 & W^2 & W^4 & W^6 & \dots & W^{N-2} \\ 1 & W^3 & W^6 & W^9 & \dots & W^{N-3} \\ \vdots & & & & & \\ 1 & W^{N-1} & W^{N-2} & W^{N-3} & \dots & W \end{pmatrix} \begin{pmatrix} f[0] \\ f[1] \\ f[2] \\ \vdots \\ f[N-1] \end{pmatrix}$$

where $W = \exp(-j2\pi/N)$ and $W = W^{2N}$ etc. $= 1$.

Discrete Fourier Transform

In the previous example and with Euler….

$$F[n] = \sum_{k=0}^{N-1} f[k] e^{-j\frac{2\pi}{N}nk} \qquad (n = 0 : N-1)$$

DFT can be rewritten as:

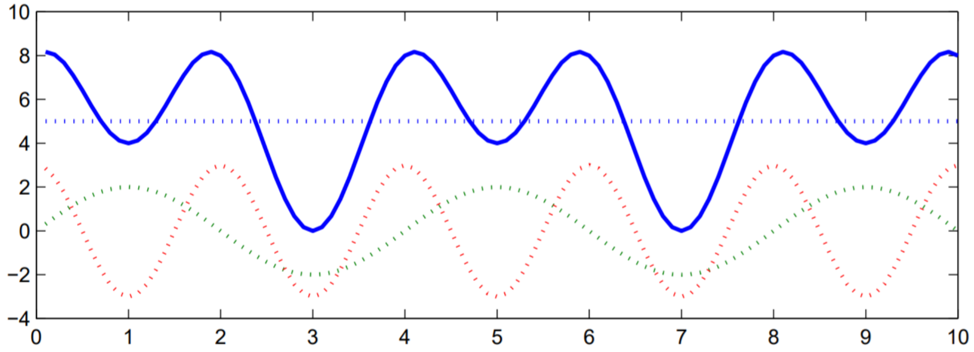
$$F[n] = \sum_{k=0}^{N-1} f[k] \cos\left(\frac{2\pi kn}{N}\right) - i \sum_{k=0}^{N-1} f[k] \sin\left(\frac{2\pi kn}{N}\right)$$

Discrete Fourier Transform

DFT – example

Let the continuous signal be

$$f(t) = \underbrace{5}_{\text{dc}} + \underbrace{2\cos(2\pi t - 90^o)}_{\text{1Hz}} + \underbrace{3\cos 4\pi t}_{\text{2Hz}}$$



Example signal for the Discrete Fourier Transform

Let us sample *f (t)* at *4* times per second (i.e. *fs = 4Hz)* from *t = 0* to $t = \frac{3}{4}$. The values of the discrete samples are given by:

$$f[k] = 5 + 2cos(\tfrac{\pi}{2}k - 90^o) + 3cos\pi k \qquad \text{by putting } t = kT_s = \tfrac{k}{4}$$

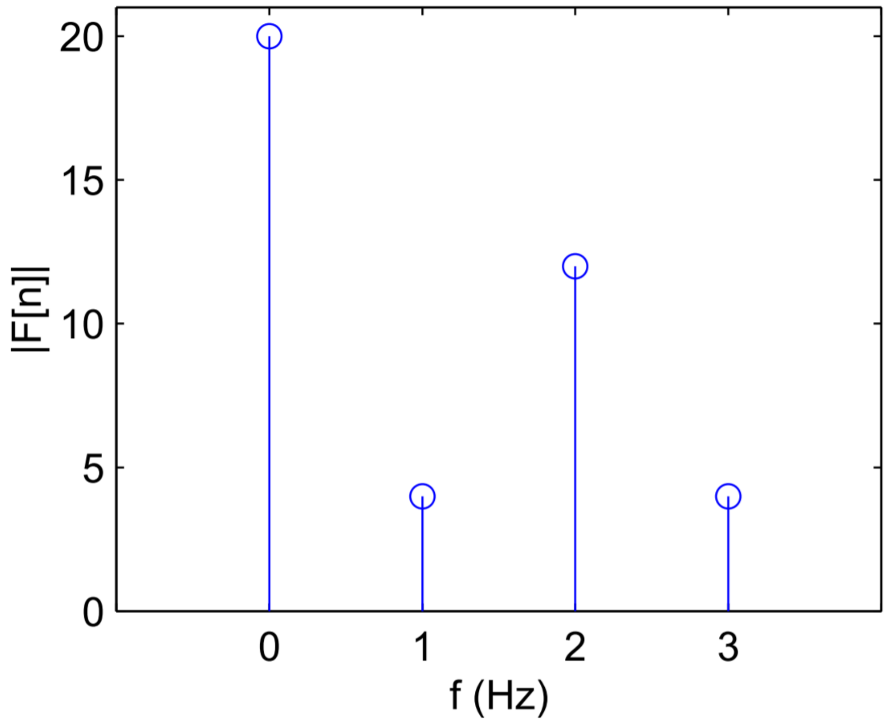
Discrete Fourier Transform

i.e. $f[0] = 8,\ f[1] = 4,\ f[2] = 8,\ f[3] = 0, \qquad (N = 4)$

$$\text{Therefore} \quad F[n] = \sum_{0}^{3} f[k] e^{-j\frac{\pi}{2}nk} = \sum_{k=0}^{3} f[k](-j)^{nk}$$

$$\begin{pmatrix} F[0] \\ F[1] \\ F[2] \\ F[3] \end{pmatrix} = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & -j & -1 & j \\ 1 & -1 & 1 & -1 \\ 1 & j & -1 & -j \end{pmatrix} \begin{pmatrix} f[0] \\ f[1] \\ f[2] \\ f[3] \end{pmatrix} = \begin{pmatrix} 20 \\ -j4 \\ 12 \\ j4 \end{pmatrix}$$

The magnitude of the DFT coefficients is shown below



Discrete Fourier Transform of 4 point sequence

Inverse Discrete Fourier Transform

The inverse transform of

$$F[n] = \sum_{k=0}^{N-1} f[k] e^{-j\frac{2\pi}{N}nk}$$

Inverse Discrete Fourier Transform

is

$$f[k] = \frac{1}{N} \sum_{n=0}^{N-1} F[n] e^{+j \frac{2\pi}{N} nk}$$

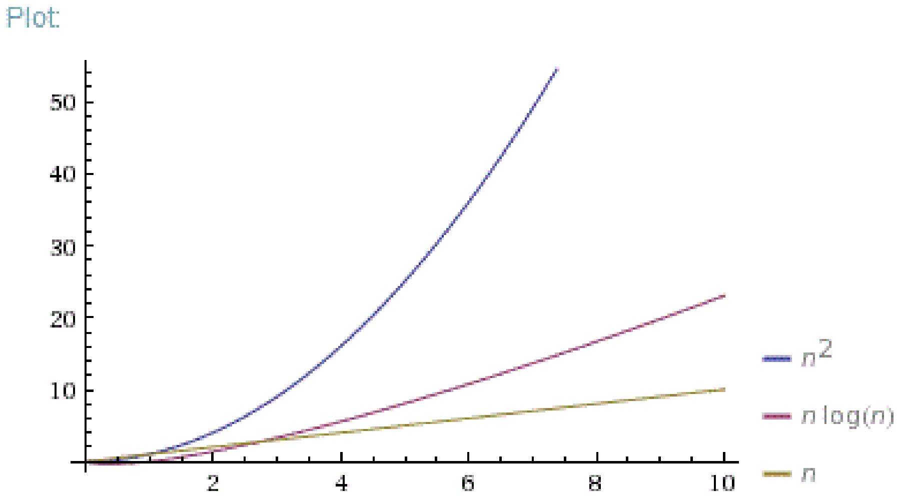
i.e. the inverse matrix is $\frac{1}{N}$ times the complex conjugate[6] of the original (symmetric) matrix.

Mind that the *F [n]* coefficients are complex. In the simplest case one can assume that the *f [k]* values are real as well.

[6]The **complex conjugate** of a complex number is the number with an equal real part and an imaginary part equal in magnitude but opposite in sign. For example, if *x* and *y* are real, then the complex conjugate of ***x + yi*** is ***x - yi .***

Fast Fourier Transform

- It is a fast and effective way of computing DFT
- Several people discovered fast FFT algorithms independently and many people have since contributed to their development
- but it was a 1965 paper by John Tukey of Princeton University and John Cooley of IBM Research that is generally credited as the starting point for the modern usage of the FFT
- Computational Complexities (Big O notation)
  – DFT – $O(n^2)$
  – FFT – $O(n\mathrm{Log}_2(n))$
- Applications
  – Audio and image compression
  – Denoising data
  – streaming video
  – satellite communications,



Fast Fourier Transform

- **DFT is a linear operator** (i.e., a matrix) → maps the data points in f to the frequency domain F:
  - {f_1, f_2, …, fn} $\overset{DFT}{\dashrightarrow}$ {F1, F2, …, Fn}
  - High computational complexity → N^2complex multiplications
- **FFT** reduces this complexity to $O(N \log N)$
  - For example, audio is generally sampled at 44.1 kHz
  - For 10 seconds of audio, the dimension of f, = 4.41×10^5 .
    - DFT → 2×10^{11} matrix multiplications.
    - FFT → 6×10^6 matrix multiplications
    - Speed-up factor of over 30, 000.
- For this reason, most devices and software have FFT libraries built in.

Fast Fourier Transform

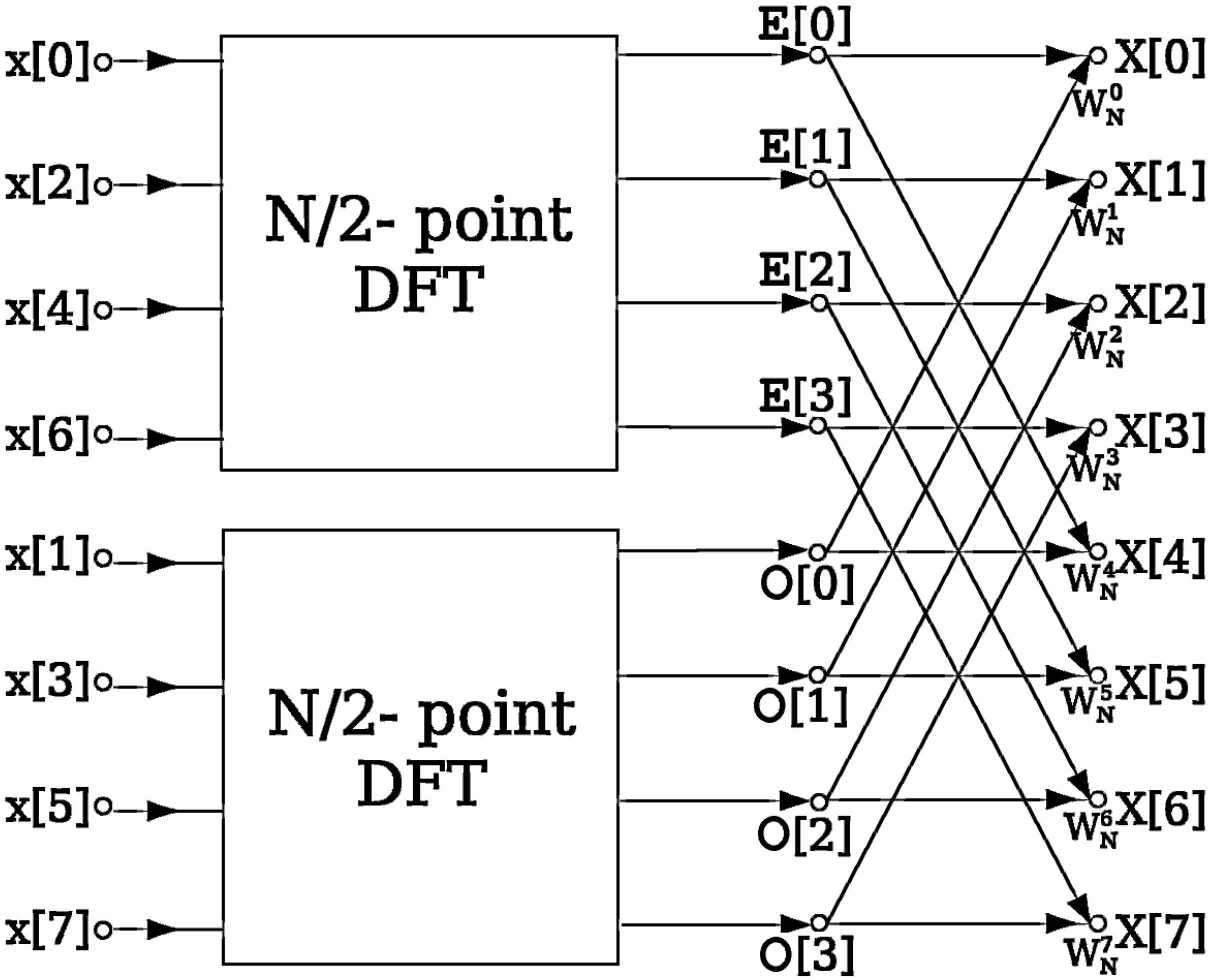
- FFT uses **divide-and-conquer**:
  - Breaks DFT of size N into smaller DFTs of size $N/2$.
  - Uses **symmetry and periodicity properties** of the complex exponential $W_N = e^{-j2\pi/N}$.
  - This drastically reduces computation.
- The most famous algorithm is the **Cooley–Tukey FFT algorithm**.

Fast Fourier Transform

- DFT matrix:
  - $X = W_N\, x$
- FFT doesn't explicitly form W_N ;instead:
  - It factorizes W_N into **sparse submatrices**.
  - Applies the transformation recursively.

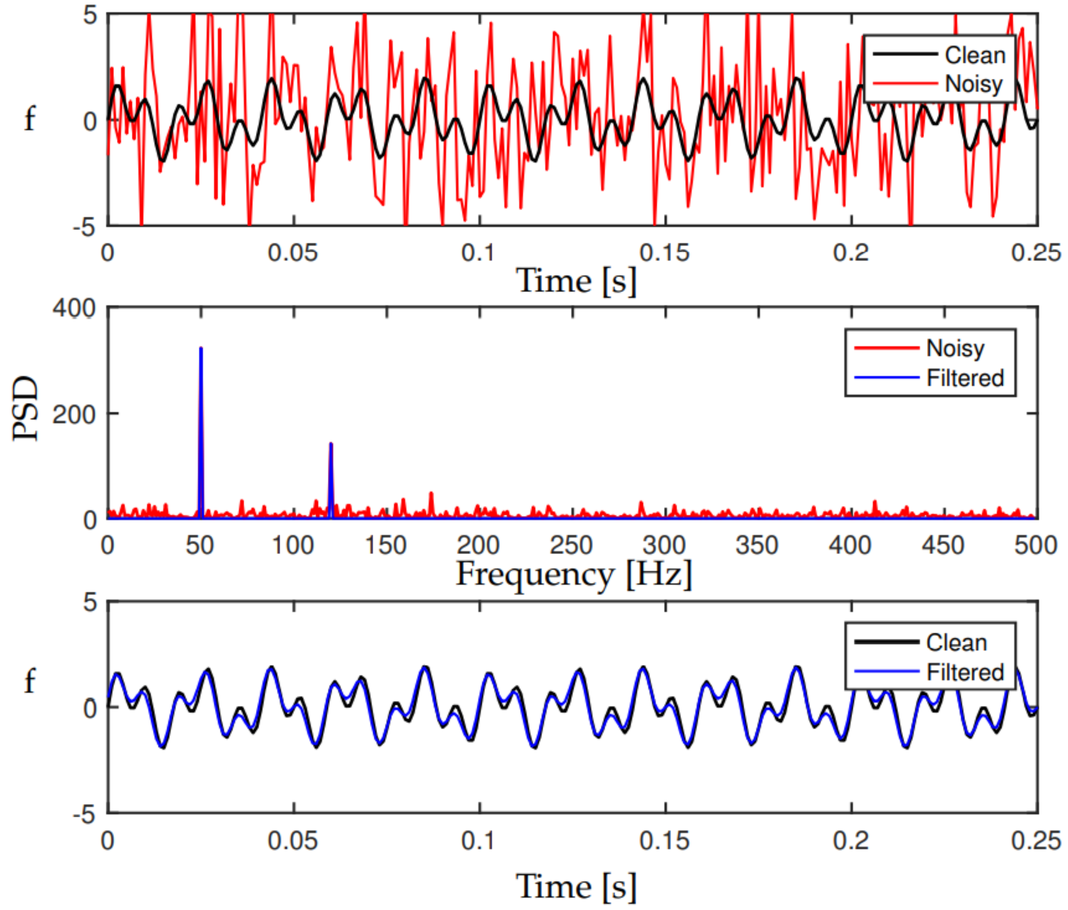
Diagram Representation

- $N = 8 = 2^3$



Application – Noise Filtering

- Consider the function with frequencies $f_1 = 50$ and $f_2 = 120$. (above, black)
  - $f(t) = \sin(2\pi f_1 t) + \sin(2\pi f_2 t)$
- We then add a large amount of Gaussian white noise to this signal (above, red)
- PSD is the normalised squared magnitude (middle plot).
- Zero out components that have power below a threshold to remove noise from the signal (filtered signal – below, blue).



To Conclude….

- Sinusoids – and periodic/non-periodic signals
- Signal spectra, amplitude, period and phase
- Trigonometric identities
- Euler to work on either ***e*** or ***cos/sin***
- Fourier series, to represent a periodic signal in terms of cosine and sine waves
- Discrete Fourier Transform to go from the time-domain to the frequency-domain
- FFT to reduce computational complexities of DFT

In addition to the module reading list have a look at……..

- https://www.le.ac.uk/users/dsgp1/LODZLECT/Lodz3.pdf

- http://www.ee.ic.ac.uk/hp/staff/dmb/courses/E1Fourier/00300_ComplexFourier.pdf

- https://users.dimi.uniud.it/~antonio.dangelo/MMS/materials/Guide_to_Digital_Signal_Process.pdf (Chapter 8)

Appendix

Fourier Series

Example

Let the function $f(x)$ be 2π-periodic and suppose that it is presented by the Fourier series:

$$f(x) = \frac{a_0}{2} + \sum_{n=1}^{\infty} \{a_n \cos nx + b_n \sin nx\}$$

Calculate the coefficients a_0, a_n, and b_n.

Fourier Series

$$f(x) = \frac{a_0}{2} + \sum_{n=1}^{\infty} \{a_n \cos nx + b_n \sin nx\}$$

To define a_0, we integrate the Fourier series on the interval $[-\pi, \pi]$:

$$\int_{-\pi}^{\pi} f(x)\, dx = \pi a_0 + \sum_{n=1}^{\infty} \left[a_n \int_{-\pi}^{\pi} \cos nx dx + b_n \int_{-\pi}^{\pi} \sin nx dx \right]$$

For all $n > 0$,

$$\int_{-\pi}^{\pi} \cos nx dx = \left(\frac{\sin nx}{n} \right)\Bigg|_{-\pi}^{\pi} = 0 \quad \text{and} \quad \int_{-\pi}^{\pi} \sin nx dx = \left(-\frac{\cos nx}{n} \right)\Bigg|_{-\pi}^{\pi} = 0.$$

Therefore, all the terms on the right of the summation sign are zero, so we obtain

$$\int_{-\pi}^{\pi} f(x)\, dx = \pi a_0 \quad \text{or} \quad a_0 = \frac{1}{\pi} \int_{-\pi}^{\pi} f(x)\, dx.$$

Fourier Series

In order to find the coefficients a_n, we multiply both sides of the Fourier series by $\cos mx$ and integrate term by term:

$$\int_{-\pi}^{\pi} f(x)\cos mx dx = \frac{a_0}{2}\int_{-\pi}^{\pi}\cos mx dx + \sum_{n=1}^{\infty}\left[a_n\int_{-\pi}^{\pi}\cos nx\cos mx dx + b_n\int_{-\pi}^{\pi}\sin nx\cos mx dx\right].$$

The first term on the right side is zero. Then, using the well-known trigonometric identities, we have

$$\int_{-\pi}^{\pi}\sin nx\cos mx dx = \frac{1}{2}\int_{-\pi}^{\pi}[\sin(n+m)x + \sin(n-m)x]\,dx = 0,$$

$$\int_{-\pi}^{\pi}\cos nx\cos mx dx = \frac{1}{2}\int_{-\pi}^{\pi}[\cos(n+m)x + \cos(n-m)x]\,dx = 0,$$

Fourier Series

if $m \neq n$.

In case when $m = n$, we can write:

$$\int_{-\pi}^{\pi} \sin nx \cos mx dx = \frac{1}{2} \int_{-\pi}^{\pi} [\sin 2mx + \sin 0]\, dx, \quad \Rightarrow \int_{-\pi}^{\pi} \sin^2 mx dx$$

$$= \frac{1}{2} \left[\left(-\frac{\cos 2mx}{2m} \right) \Bigg|_{-\pi}^{\pi} \right] = \frac{1}{4m} \left[-\cancel{\cos(2m\pi)} \; + \; \cancel{\cos(2m(-\pi))} \right] = 0;$$

$$\int_{-\pi}^{\pi} \cos nx \cos mx dx = \frac{1}{2} \int_{-\pi}^{\pi} [\cos 2mx + \cos 0]\, dx, \quad \Rightarrow \int_{-\pi}^{\pi} \cos^2 mx dx$$

$$= \frac{1}{2} \left[\left(\frac{\sin 2mx}{2m} \right) \Bigg|_{-\pi}^{\pi} + 2\pi \right] = \frac{1}{4m} [\sin(2m\pi) \; - \; \sin(2m\,(-\pi))] + \pi = \pi.$$

Thus,

$$\int_{-\pi}^{\pi} f(x) \cos mx dx = a_m \pi, \quad \Rightarrow a_m = \frac{1}{\pi} \int_{-\pi}^{\pi} f(x) \cos mx dx, \;\; m = 1, 2, 3, \ldots$$

Fourier Series

Similarly, multiplying the Fourier series by $\sin mx$ and integrating term by term, we obtain the expression for b_m :

$$b_m = \frac{1}{\pi} \int_{-\pi}^{\pi} f(x) \sin mx dx, \quad m = 1, 2, 3, \ldots$$

Rewriting the formulas for a_n, b_n, we can write the final expressions for the Fourier coefficients:

$$a_n = \frac{1}{\pi} \int_{-\pi}^{\pi} f(x) \cos nx dx, \quad b_n = \frac{1}{\pi} \int_{-\pi}^{\pi} f(x) \sin nx dx.$$