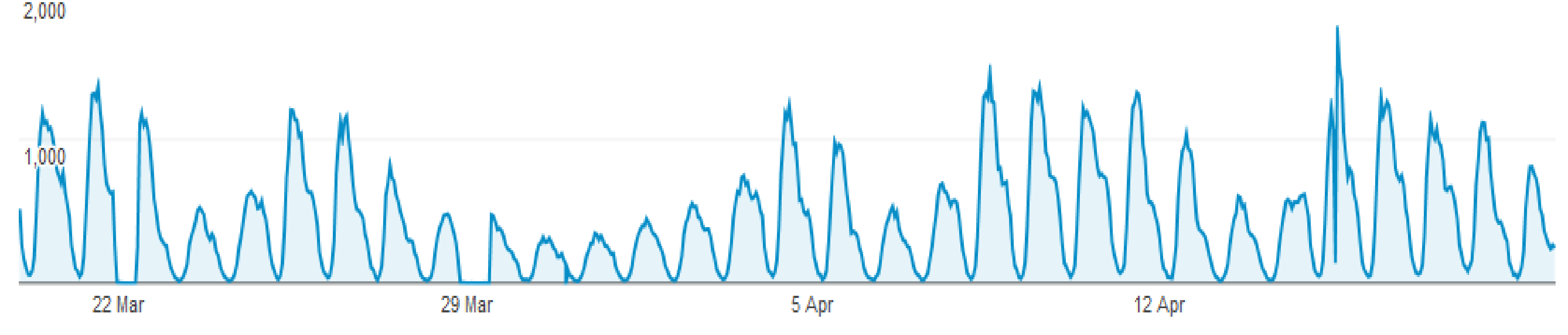
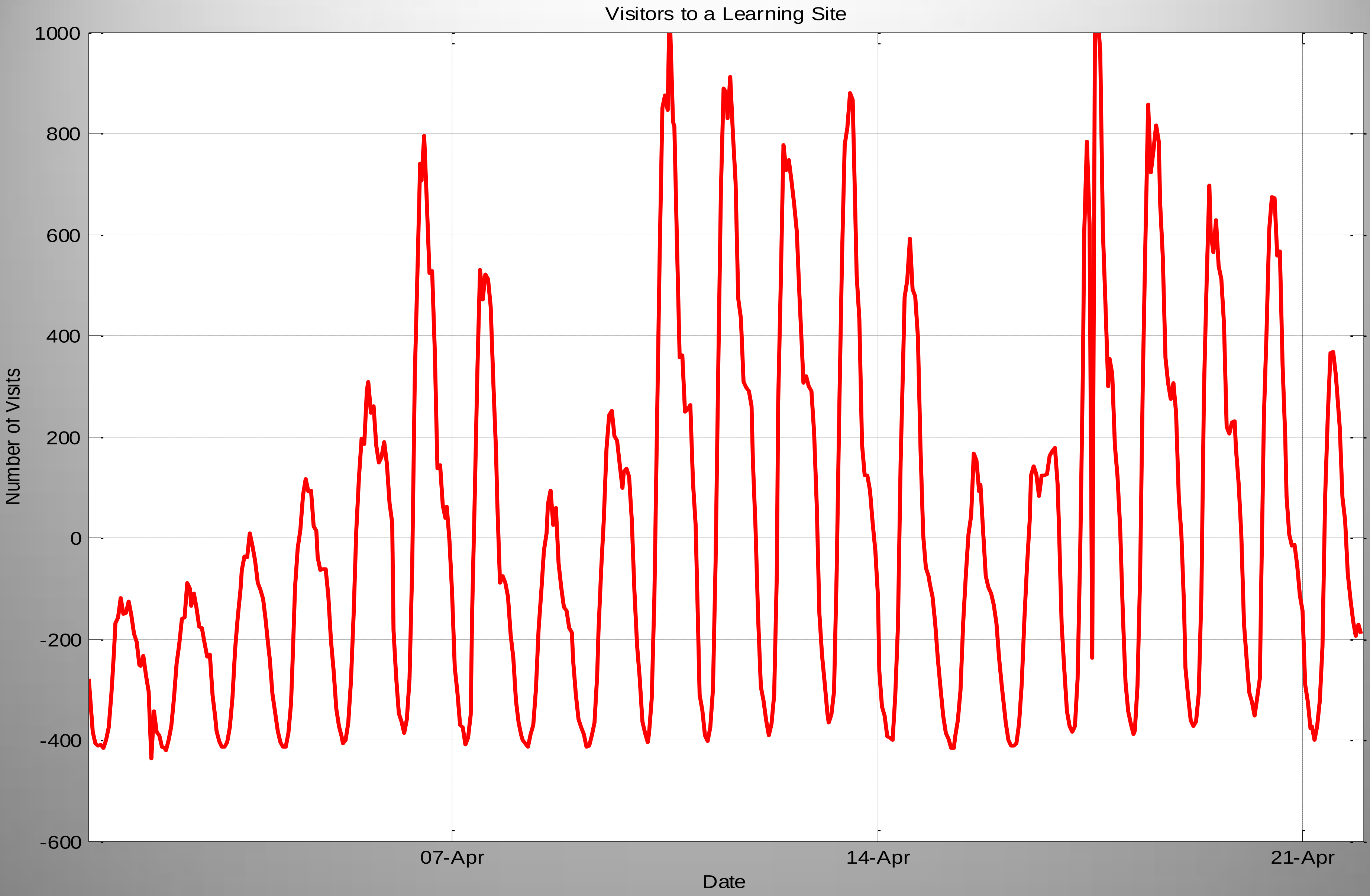
Time Series Data and Fourier Transforms



Jason Bailey

Quick Summary

- Look Time Series Data
- See data in Time domain (time series) and Frequency domain (using Fourier Transform)
- Application: Filter data/Extract pattern with Fourier Transform
- FFT - Fast Fourier Transform

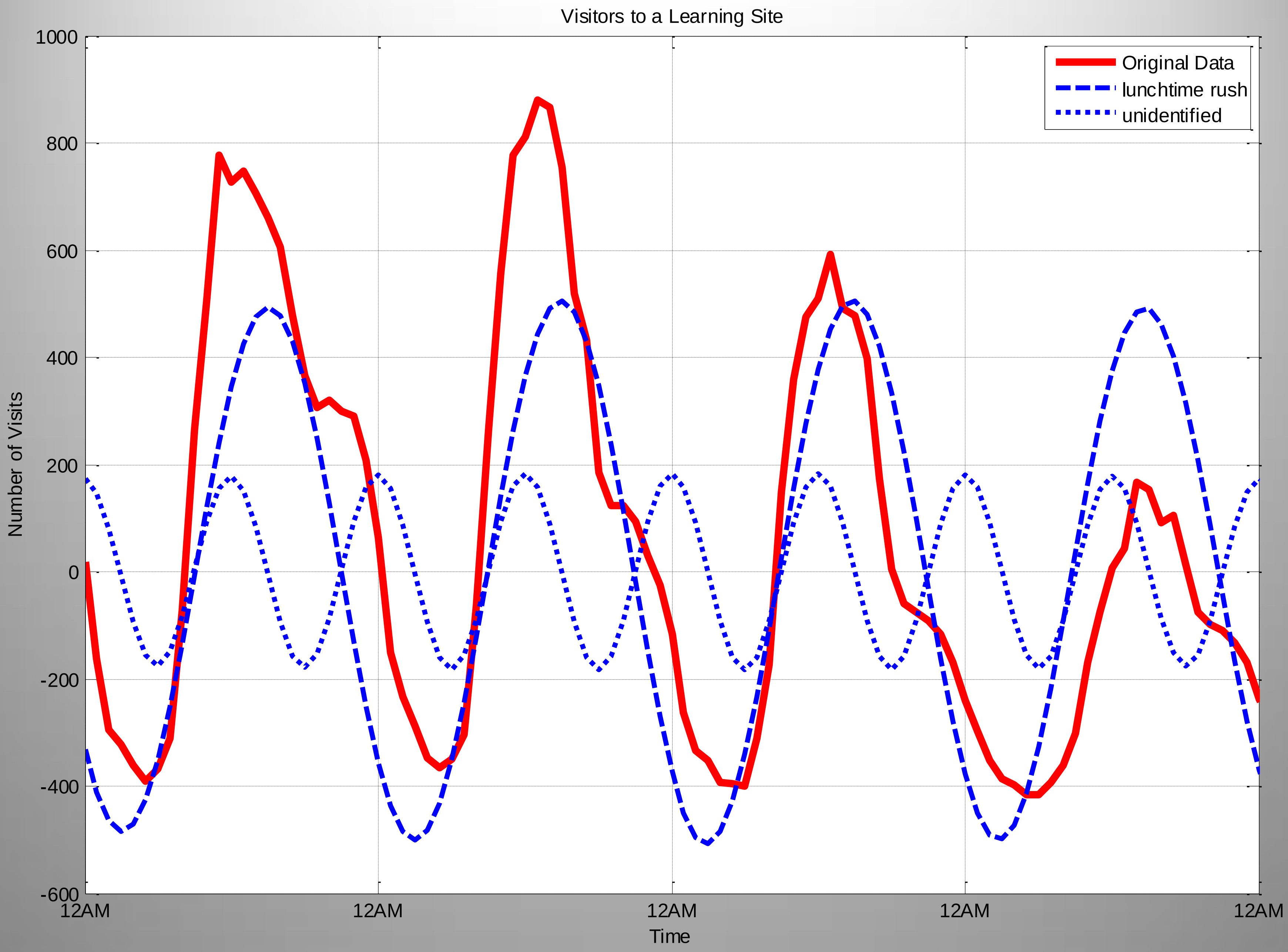

Visitors to a Learning Site
Number of Visits
1000
800
600
400
200
0
-200
-400
-600
07-Apr
14-Apr
21-Apr
Date

What is Time Series Data

- A sequence of data points
- Typically at successive points in time spaced at uniform time intervals
- Used:
- statistics, signal processing, pattern recognition, finance, weather forecasting, earthquake prediction, control engineering and communications engineering

http://en.wikipedia.org/wiki/Time_series

What if we want to extract
a pattern
from time series data?



https://gist.github.com/espeecat/5438953

A sine wave or sinusoid

$y(t) = A \sin(2\pi f t + \phi)$

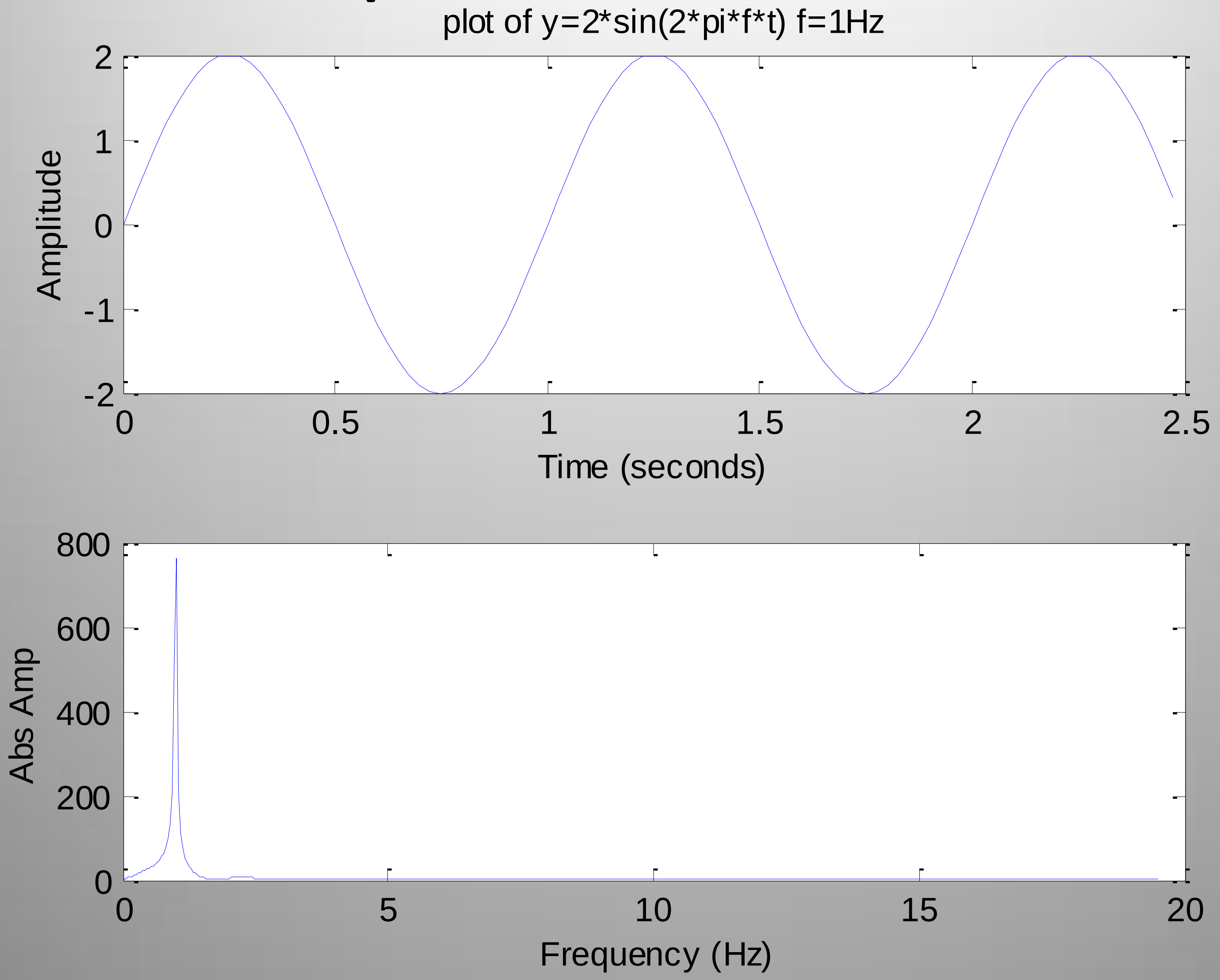
Sometimes $2\pi f$ written as ω

Cosine too

Much better to see it in a graph

- Use a tool like Matlab
  - A programmable calculator with good graph/chart abilities
- Other tools are available and much cheaper

An example of a sinusoid and FFT



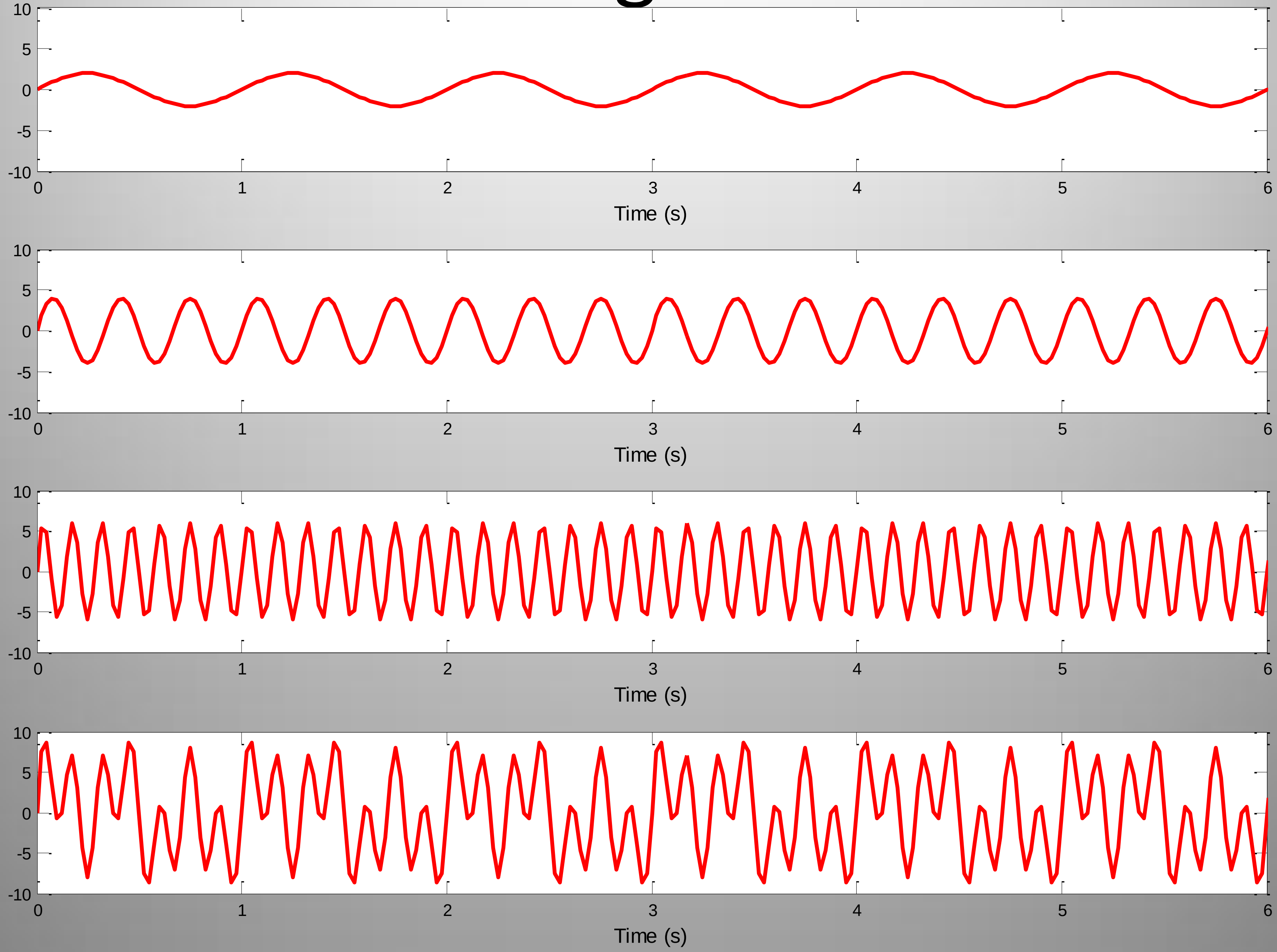
https://gist.github.com/espeecat/5439069

The Fourier Transform (FFT)

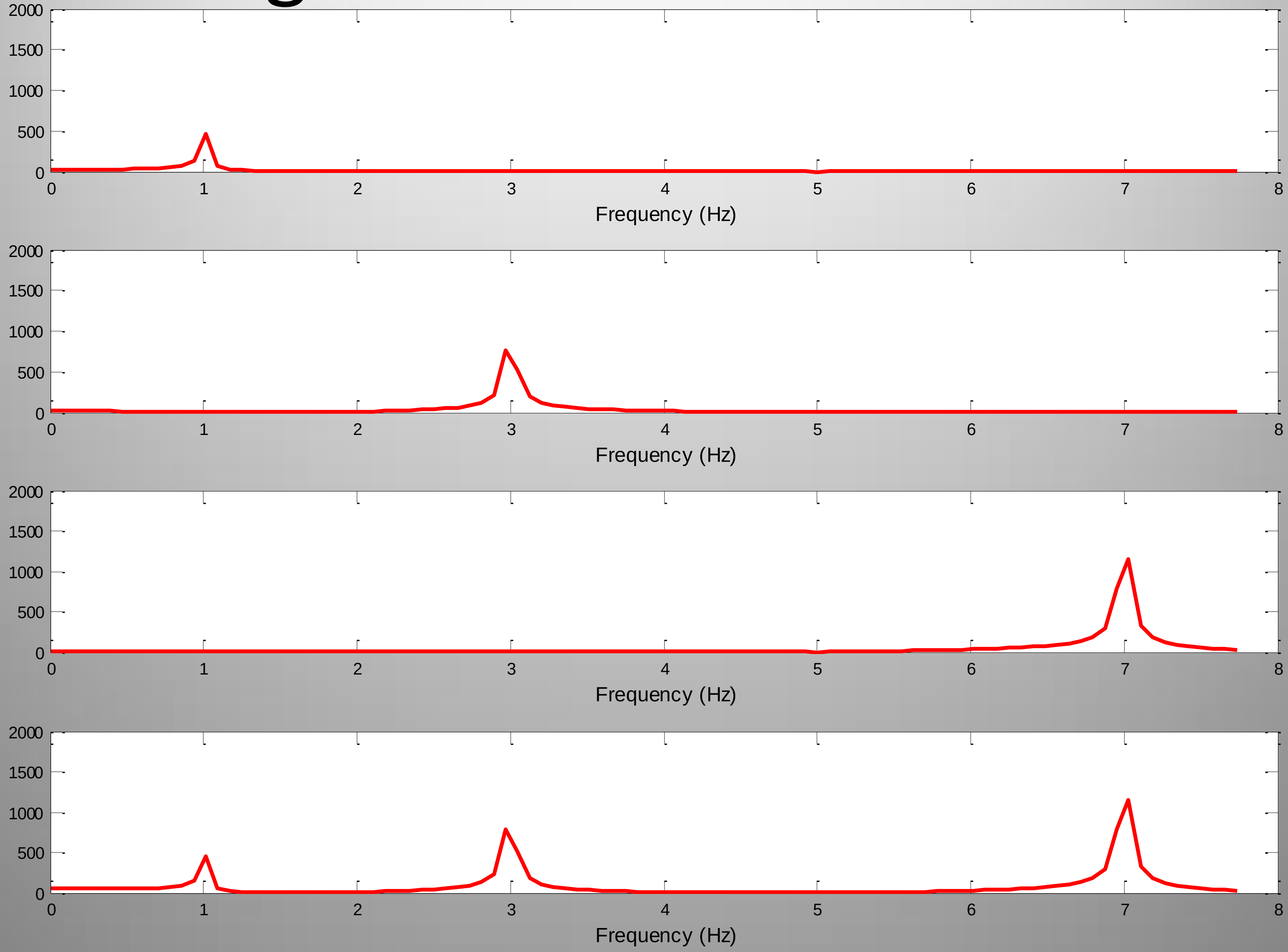
- Based on Fourier Series - represent periodic time series data as a sum of sinusoidal components (sine and cosine)
- (Fast) Fourier Transform [FFT] – represent time series in the frequency domain (frequency and power)
- The Inverse (Fast) Fourier Transform [IFFT] is the reverse of the FFT
- Like graphic equaliser on music player

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

Combining Sinusoids



Looking at the Fourier Transforms



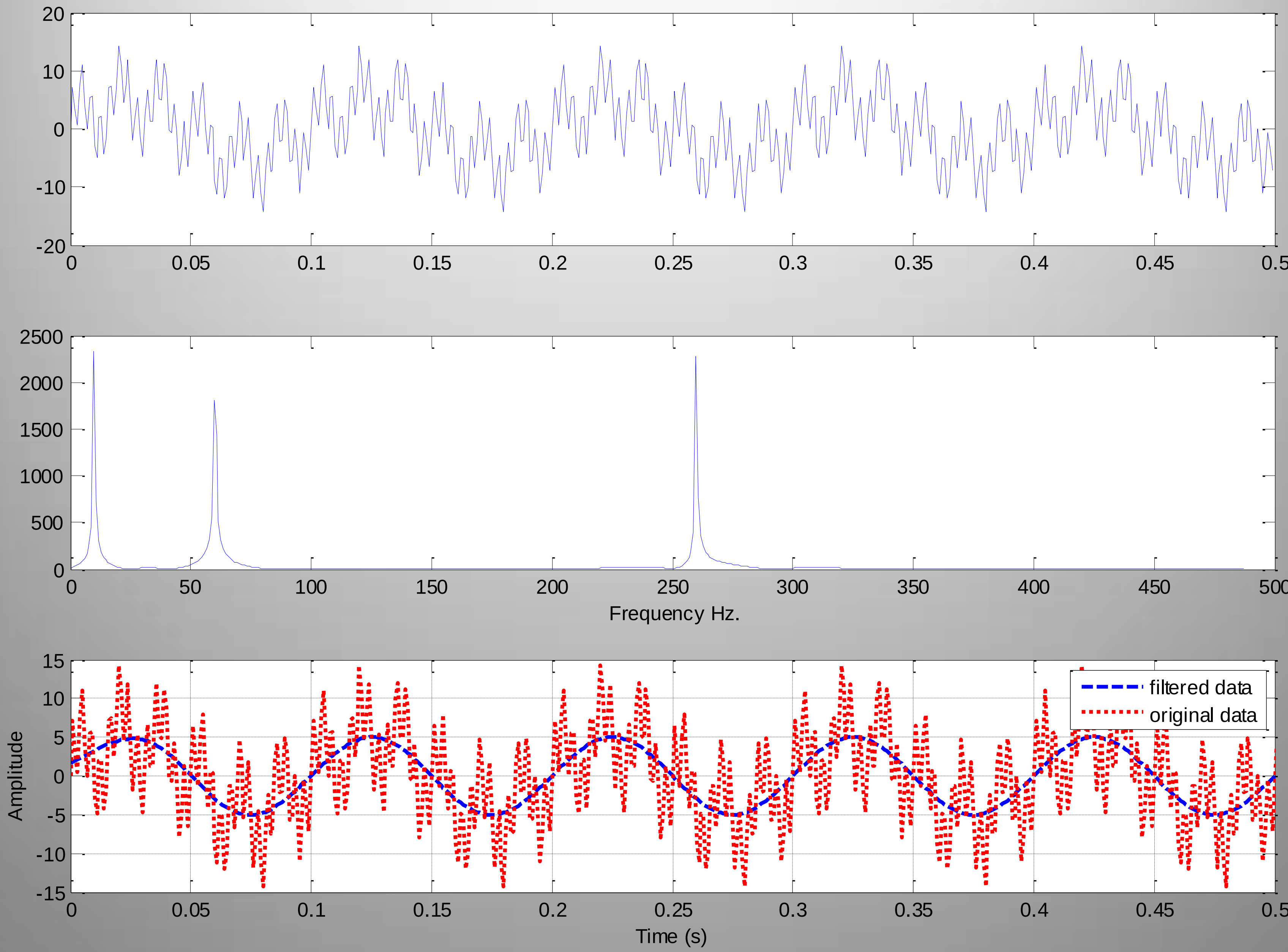
Applications of Fourier Transform

- Shazam – “finger printing” using Fourier Transforms
- Images – The Gabor Transform for facial recognition?
- Filtering data/ Extracting patterns
- Sound processing – discarding sound
- System Identification

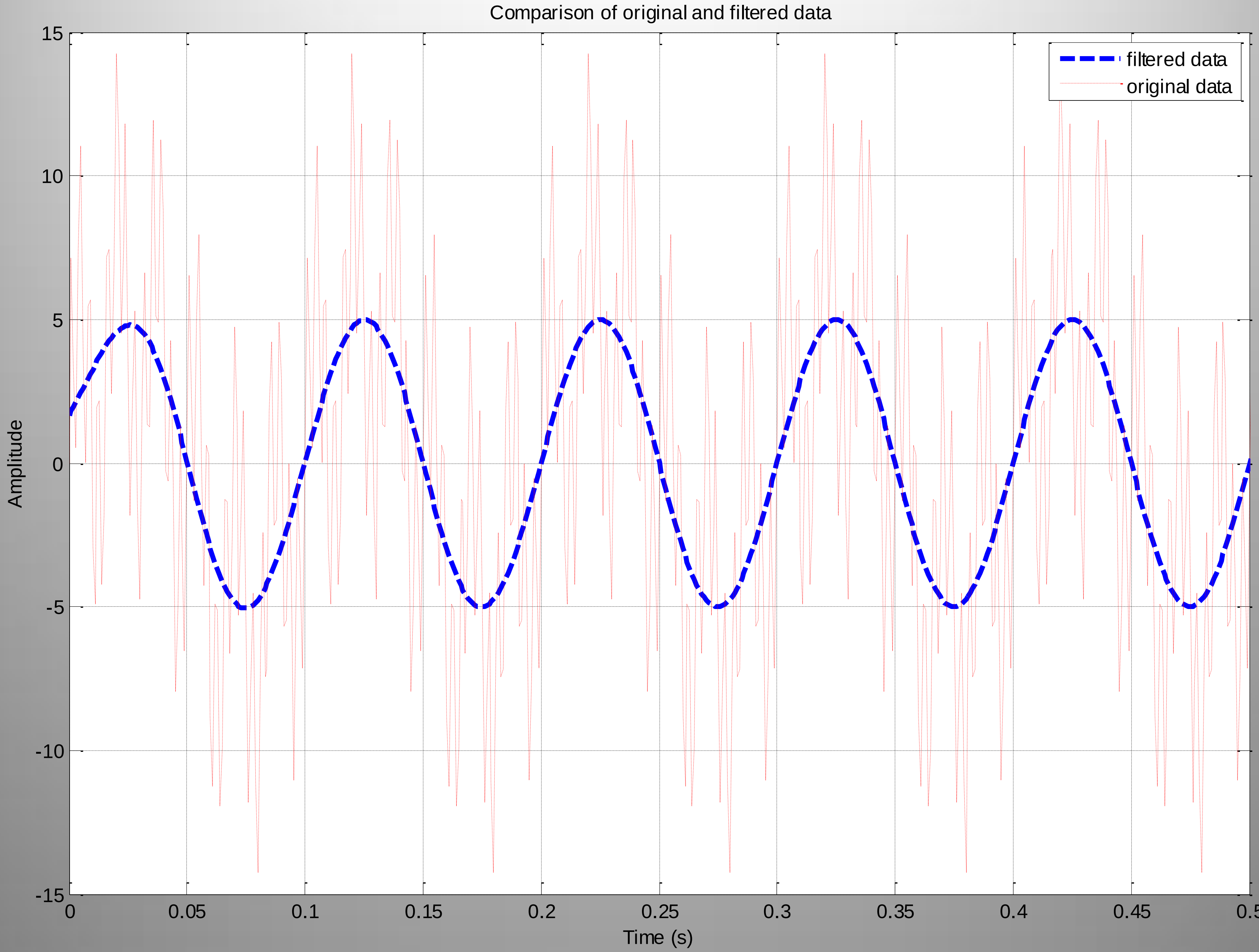
The (Fast) Fourier Transform

- Discrete-time Fourier Transform –assumes sampled data and limited length
- Implementations available in lots of programming languages e.g. http://www.fftw.org/
- Python numpy.fft

Filtering Time Series Data



Original data and filtered data

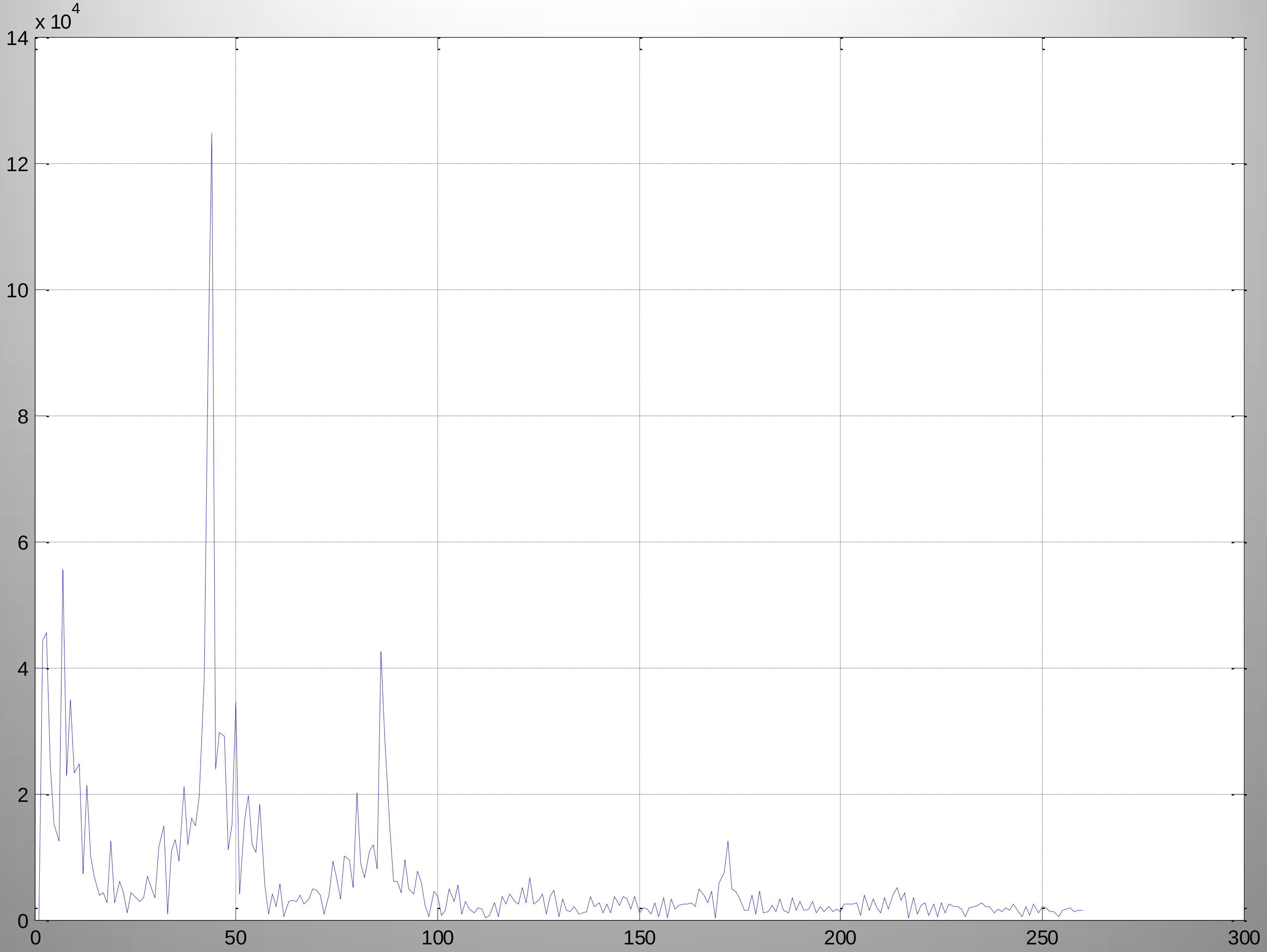


Thank you

Alternative to Matlab

- SciLab – https://www.scilab.org/
- Octave - http://www.gnu.org/software/octave/
- R - http://www.r-project.org/
- Programming language & graph library

Twitter @espeecat www.jasonbailey.net


x 10^4
14
12
10
8
6
4
2
0
0
50
100
150
200
250
300

Discrete-Time Fourier Transform

- ω = 2πf –angular frequency
- Euler Formula used but this represents

$$X_{2\pi}(\omega) = \sum_{n=-\infty}^{\infty} x[n]\, e^{-i\omega n}.$$

$$e^{i2\pi f t} = \cos(2\pi f t) + i \sin(2\pi f t)$$

$$X_k = \sum_{n=0}^{N-1} x_n \cdot e^{-i2\pi kn/N}$$

$$x_n = \frac{1}{N} \sum_{k=0}^{N-1} X_k \cdot e^{i2\pi kn/N}$$