UNIVERSITY OF CAMBRIDGE

Mobile Health Practical 1

Introduction to Signal Processing

Erika Bondareva
eb729@cam.ac.uk

Kayla-Jade Butkow
kjb85@cam.ac.uk

Goal of the Practical

- Intro to signal processing in Python
- Introduction to IMU data
- Python Notebook in Colab for data processing
  - Load and organise data
  - Data visualisation
  - Signal processing in Python
- Learn about the upcoming assignment

Intro to Signal Processing with Python

- You should be already familiar with the following concepts:
  - Analogue and digital signal
  - Nyquist theorem
  - Discrete Fourier transform and Fast Fourier transform
  - Spectrograms
  - Basics of filtering
- Most common tools for digital signal processing – Python, MATLAB, C++

Intro to Signal Processing with Python

Python tools necessary for this practical

Data loader	**Data organiser**	**Data visualiser**	**Actual signal processing**

Intro to Signal Processing with Python

Python tools necessary for this practical

Data loader | Data organiser | Data visualiser | Actual signal processing

- Need to load:
  - sensor data — could be a CSV, WAV, etc.
  - metadata — typically a CSV

```
import csv

with open('data.csv', 'r') as csvfile:
    data = csv.reader(csvfile)
    for row in data:
        print(row)
```

```
import pandas as pd

data = pd.read_csv('data.csv')
print(data)
```

Intro to Signal Processing with Python

Python tools necessary for this practical

Data loader | **Data organiser** | Data visualiser | Actual signal processing

- Display the data and transform it if needed

```python
import pandas as pd
import numpy as np
import re
```

- `import pandas as pd` → Library for data manipulation and analysis
- `import numpy as np` → Supports and allows complex operations on large, multi-dimensional matrices
- `import re` → Library for working with regular expressions

Intro to Signal Processing with Python

Python tools necessary for this practical

Data loader | Data organiser | **Data visualiser** | Actual signal processing

- Always a good idea not to work with the data blindly
- The most straightforward library — `matplotlib.pyplot`

```python
import matplotlib.pyplot as plt

x = [1, 2, 3, 4, 5]
y = [2, 4, 6, 8, 10]
plt.plot(x, y)
plt.show()
```

Intro to Signal Processing with Python

Python tools necessary for this practical

Data loader

Data organiser

Data visualiser

Actual signal processing

- Multiple libraries with predefined functions

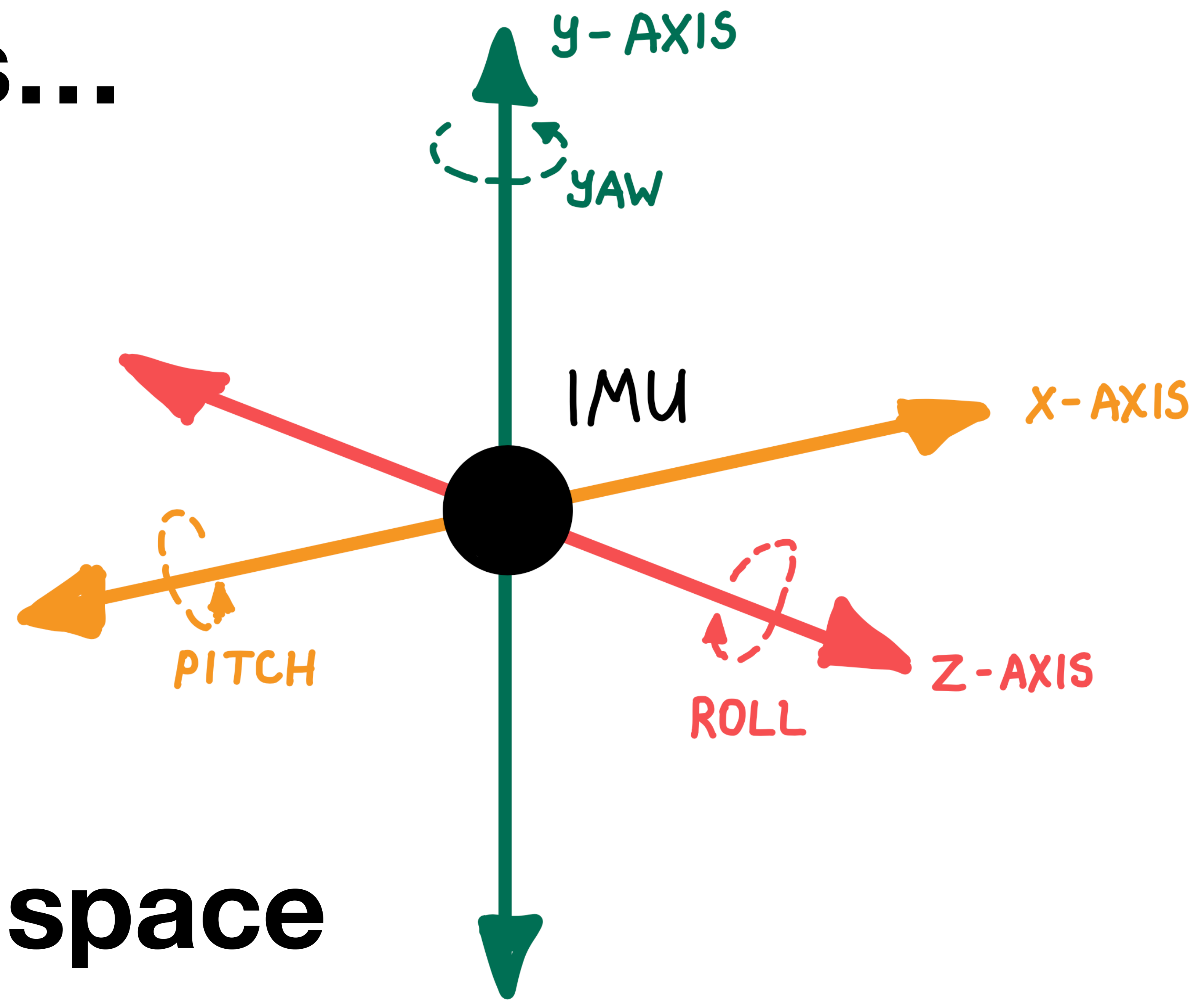
```
import numpy as np
from scipy import signal

# generate a 1kHz sine wave
fs = 10e3
f = 1e3
t = np.linspace(0, 1, fs, endpoint=False)
x = np.sin(2 * np.pi * f * t)
```

Inertial Measurement Unit (IMU)

Electronic device that measures and reports…

- specific force / gravity — accelerometer
- angular rate — gyroscope
- (sometimes) magnetic field — magnetometer

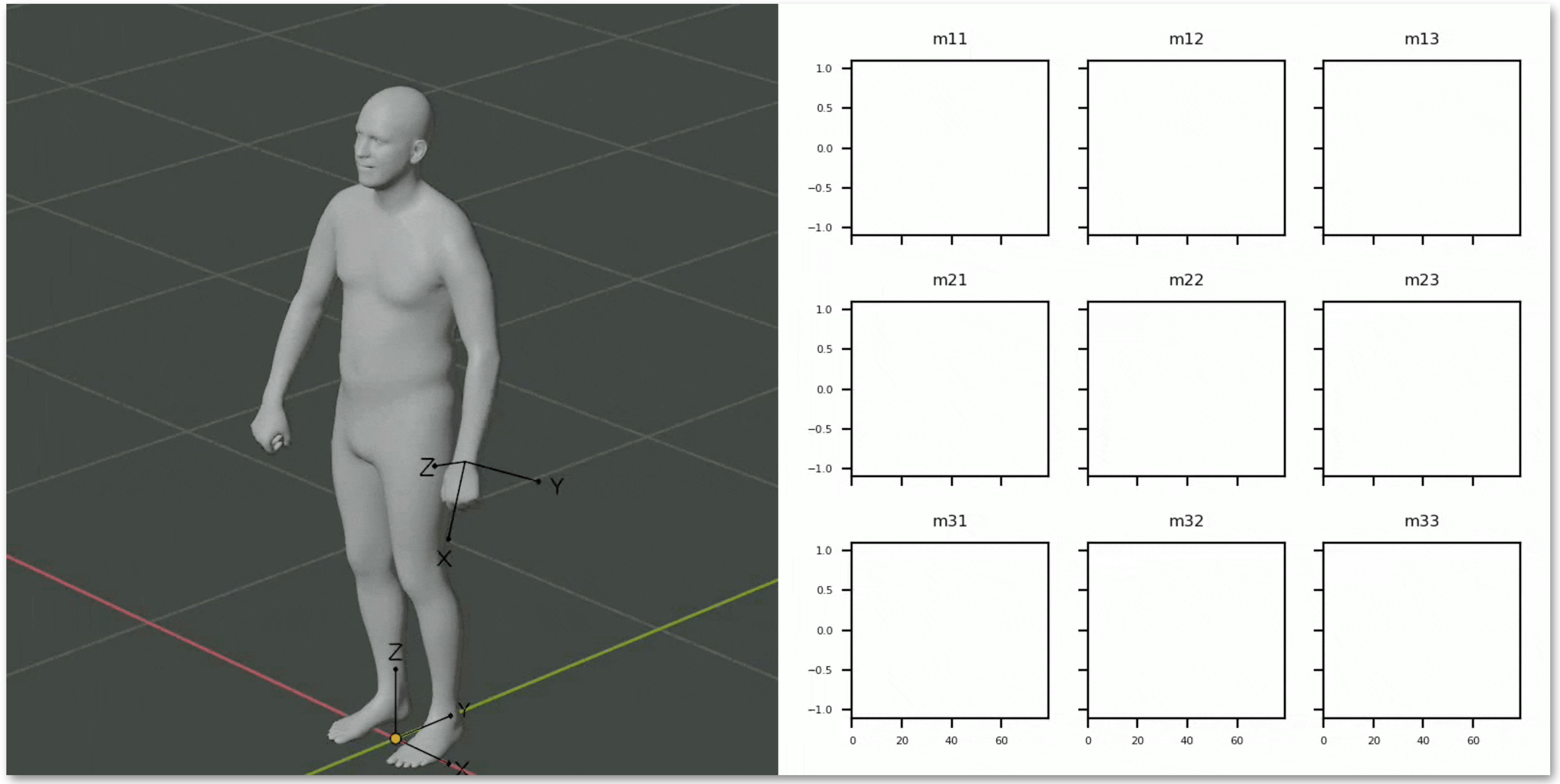


… to estimate orientation of an object in 3D space

Infinity AI IMU Fitness Dataset

Open Source Synthetic Dataset for Fitness Applications

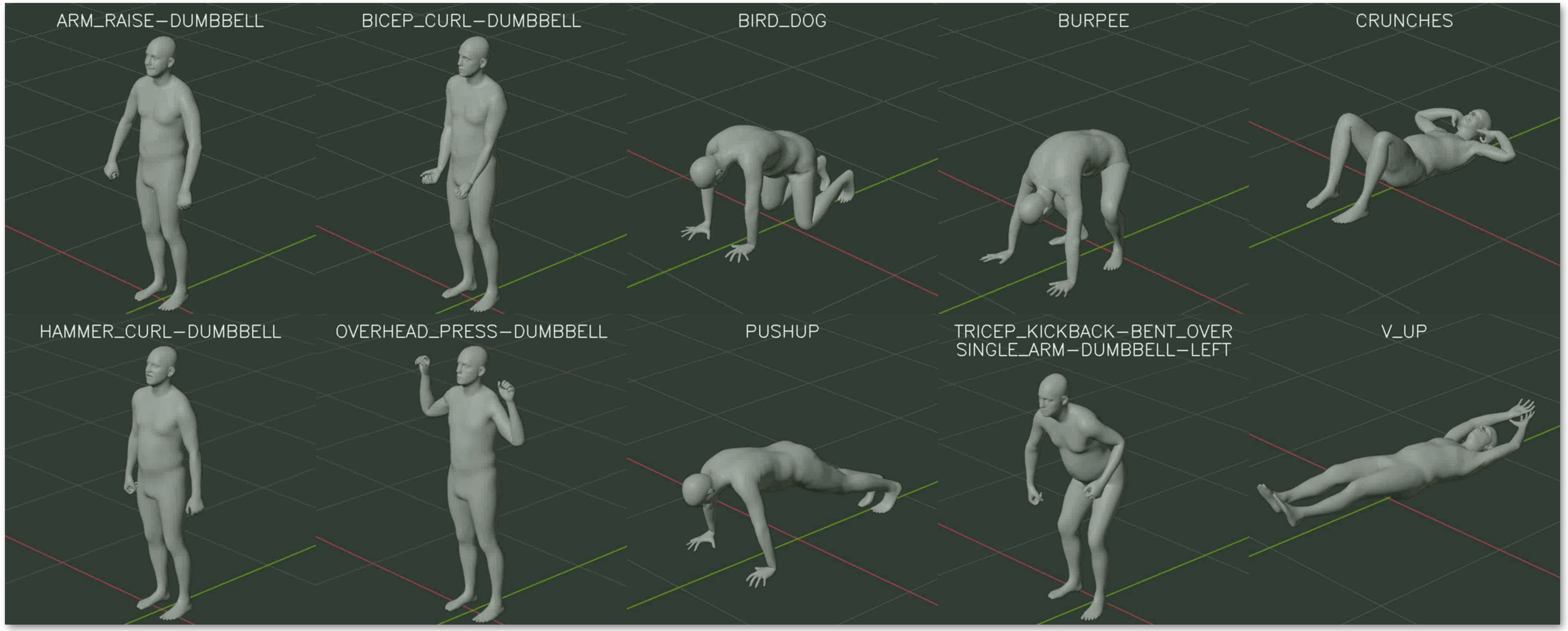
- Paired video and IMU data
  (9 DoF: angular positions)
- We've extracted a subset of the data
- 20Hz sampling frequency



[1] https://marketplace.infinity.ai/products/imu-fitness-basic-dataset

Infinity AI IMU Fitness Dataset

Open Source Synthetic Dataset for Fitness Applications



- 10 exercises
- 10 samples each
- 3 files per sample:
  - **CSV**
  - **JSON**
  - **MP4**

[1] https://marketplace.infinity.ai/products/imu-fitness-basic-dataset

Importing Libraries

`import requests` → Allows to send HTTP requests

`import zipfile` → Enables work with ZIP archives

`import os` → Provides a way of using operating system (OS) dependent functionality

`import numpy as np` → NUMerical PYthon

`import scipy` → SCIentific PYthon, provides functions for stats and signal processing

`import pandas as pd` → Used for working with datasets

`import matplotlib.pyplot as plt` → Plotting library

Loading data

	rep_count_from_intermediate	rep_count_from_start	ref_xy_rotation	time	rotation_matrix_m11	rotation_matrix_m12	rotation_matrix_m13	rotation_matrix_m21	rotation_matrix_m22	rotation_matrix_m23	rotation_matrix_m31	rotation_matrix_m32	rotation_matrix_m33
0	0.4736842105263160	0.0	4.8182613644283	0.0	0.31345245242118800	-0.21886013448238400	-0.9240388870239260	0.09758635610342030	0.9753504395484920	-0.1979101151227950	0.9445763826370240	-0.028138162568211600	0.3270837366580960
1	0.49027919940091000	0.016594988874594500	4.8182613644283	0.05	0.3190750181674960	-0.2161204069852830	-0.9227584600448610	0.09248825907707210	0.9761050343513490	-0.19663383066654200	0.9432057738304140	-0.022603364661335900	0.331439346075058
2	0.5068740668841270	0.033189856357810800	4.8182613644283	0.1	0.33161893486976600	-0.21445204317569700	-0.9187160134315490	0.08539441972970960	0.9766470193862920	-0.19715073704719500	0.9395406246185300	-0.013074328191578400	0.34218764305114700
3	0.5234688116608430	0.049784601134527000	4.8182613644283	0.150000	0.3504527509212490	-0.21328036487102500	-0.9119728207588200	0.07712976634502410	0.976990818977356	-0.19884651899337800	0.9333991408348080	-0.0006539252935908740	0.35883939266204800
4	0.5400634317583760	0.06637922123206050	4.8182613644283	0.2	0.3798833191394810	-0.2133406549692150	-0.9000969529151920	0.06752893328666690	0.976841151714325	-0.20303015410900100	0.9225663542747500	0.016345201060175900	0.38549232482910200
5	0.5566579245464840	0.08297371402016790	4.8182613644283	0.25	0.4221969544887540	-0.21402782201767000	-0.8808756470680240	0.05850962549448010	0.9761358499526980	-0.2091301530599590	0.9046139717102050	0.03675442561507230	0.42464426159858700
6	0.5732522867373600	0.09956807621104460	4.8182613644283	0.300000	0.47755521535873400	-0.21485623717308000	-0.851925790309906	0.05239385738968850	0.974876880645752	-0.21649464964866600	0.8770380020141600	0.05875243619084360	0.4768146872520450
7	0.5898465143856410	0.1161623038593260	4.8182613644283	0.350000	0.5444672703742980	-0.2160136103630070	-0.8104896545410160	0.050907816737890200	0.9729984998703000	-0.2251272201538090	0.8372358083724980	0.08131415396928790	0.5407626628875730

Data format

There are few problems with **Euler angles** (XYZ axes):

- Gimbal lock
- Cannot do smooth interpolation
- Computationally inefficient
- Unintuitive for rotations

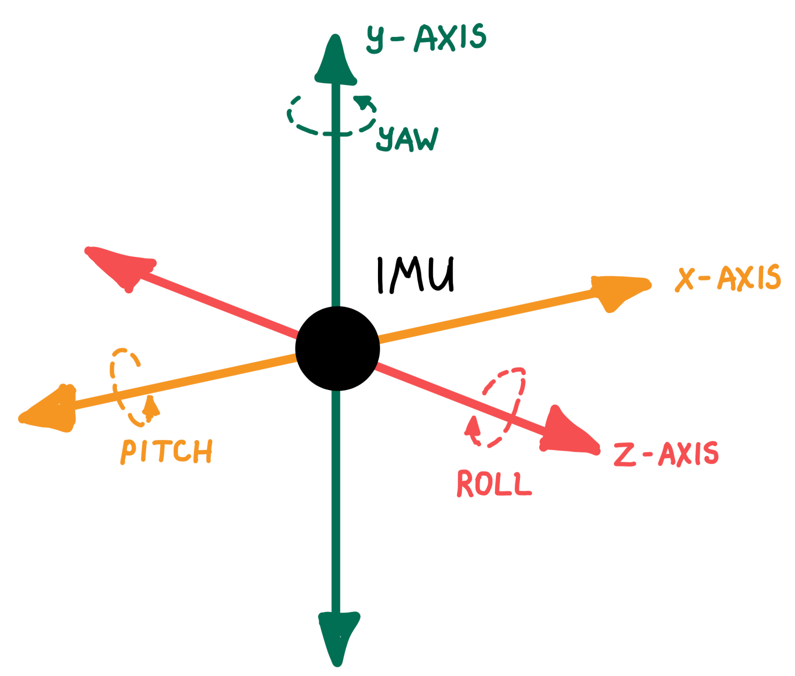


Image credit: Mark Hughes

So instead can use **quaternions** —
mathematical representation for 3D rotations.

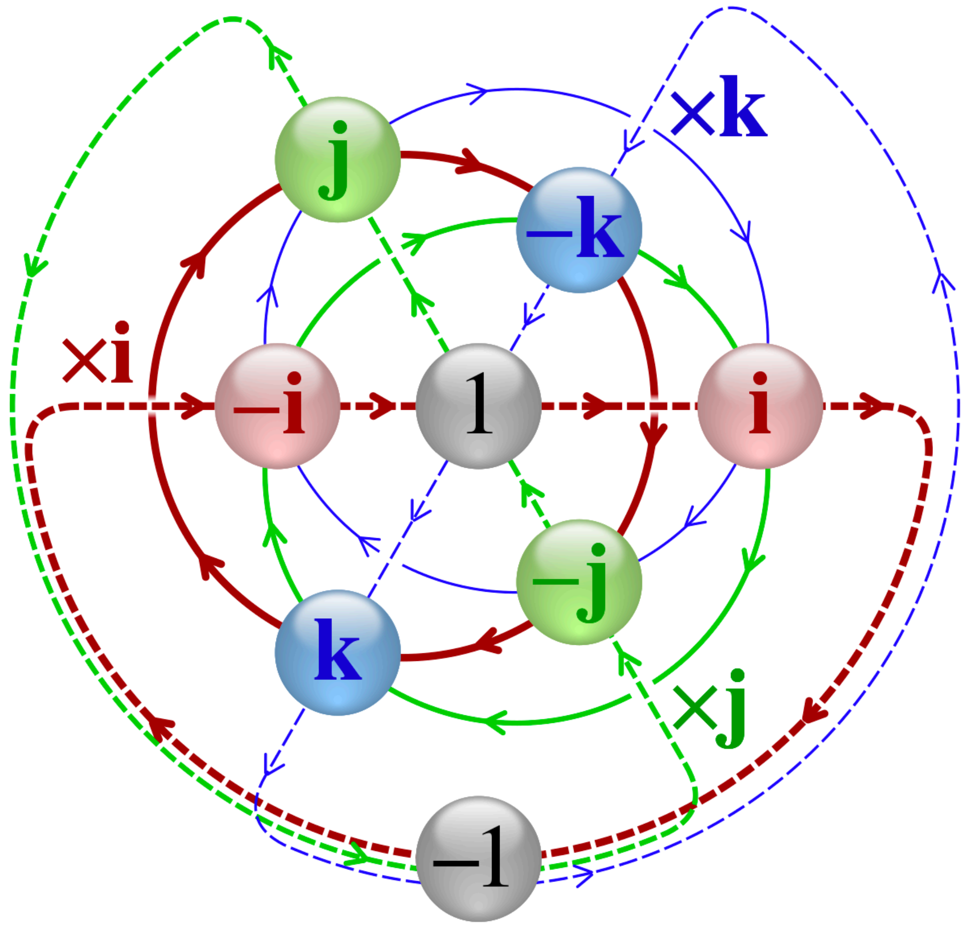
Quaternions

$$a + b\mathrm{i} + c\mathrm{j} + d\mathrm{k}$$

a, b, c, d — real numbers

$\mathrm{i}, \mathrm{j}, \mathrm{k}$ — basic quaternions

$$\mathrm{i}^2 = \mathrm{j}^2 = \mathrm{k}^2 = -1$$



Multiplication table

	1	i	j	k
1	1	i	j	k
i	i	-1	k	-j
j	j	-k	-1	i
k	k	j	-i	-1

Basic quaternions can be interpreted as unit-vectors pointing along X, Y and Z.

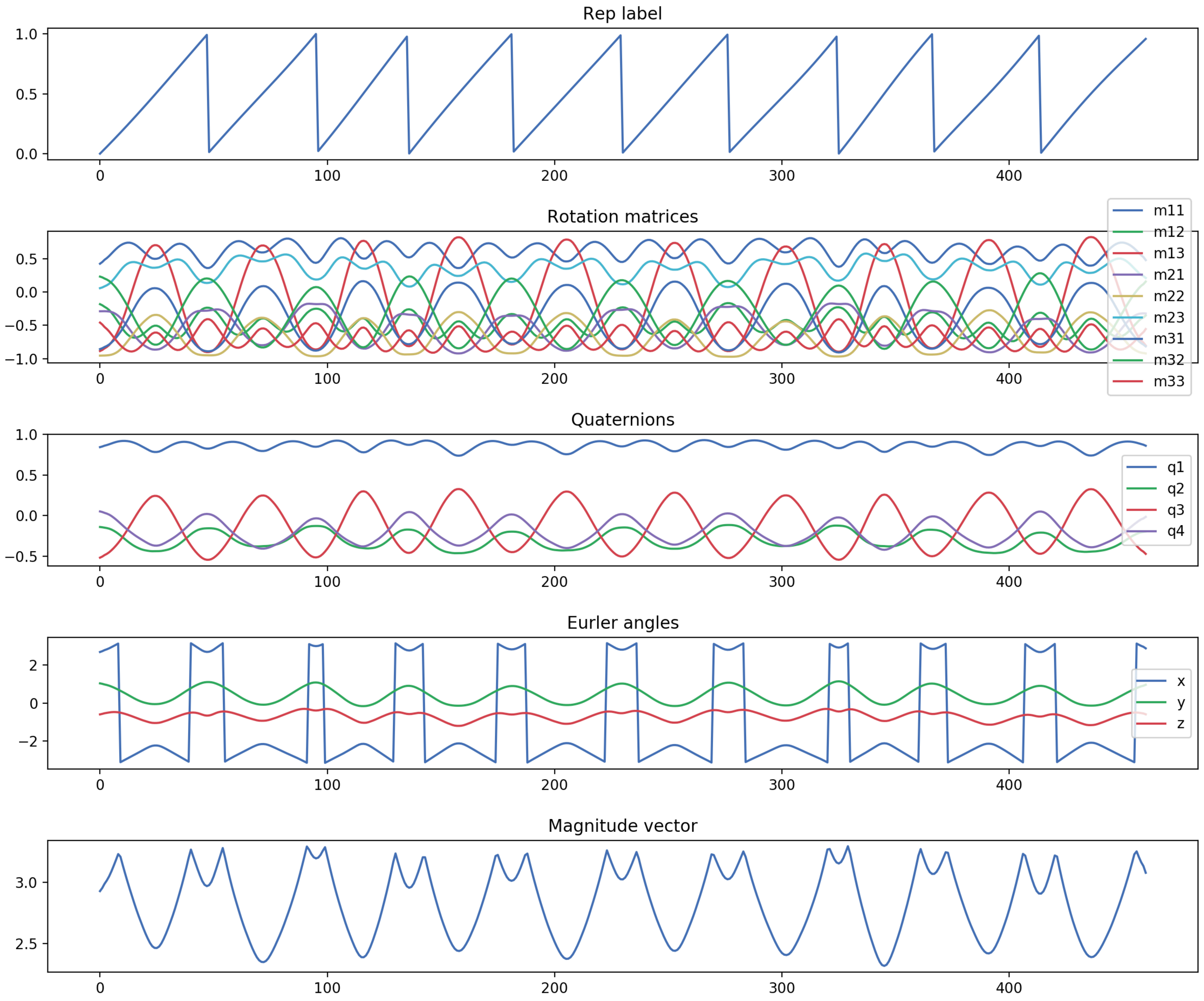
Can convert rotation matrices easily to quaternions or Euler angles.

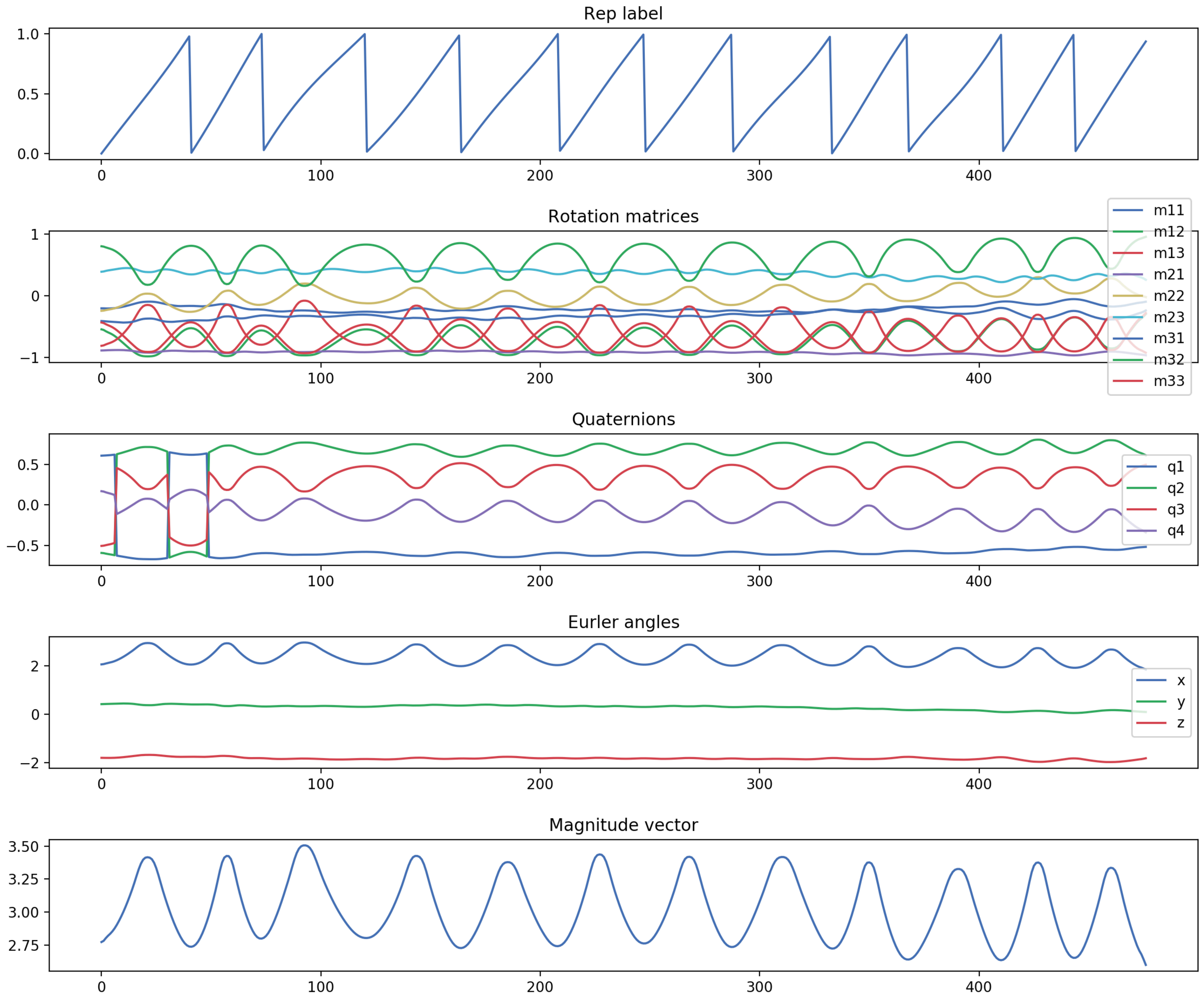
Converting quaternions to Euler angles

Can convert rotation matrices easily to quaternions or Euler angles.

We've loaded the data and converted it to a form we understand. How do we now start to analyse and process it?

Time series visualisation — bicep curls





Time series visualisation — magnitude vector

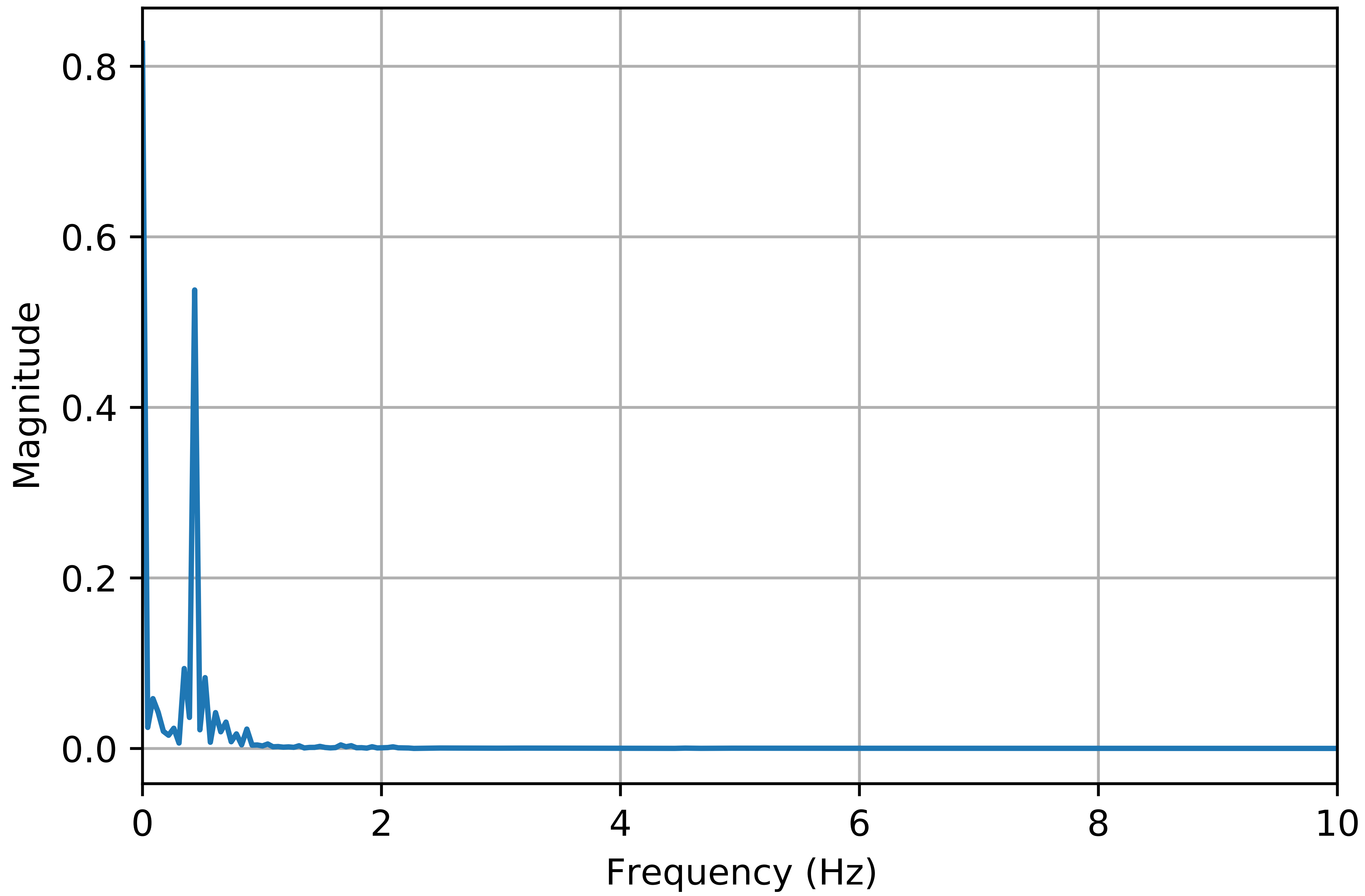
Time series visualisation

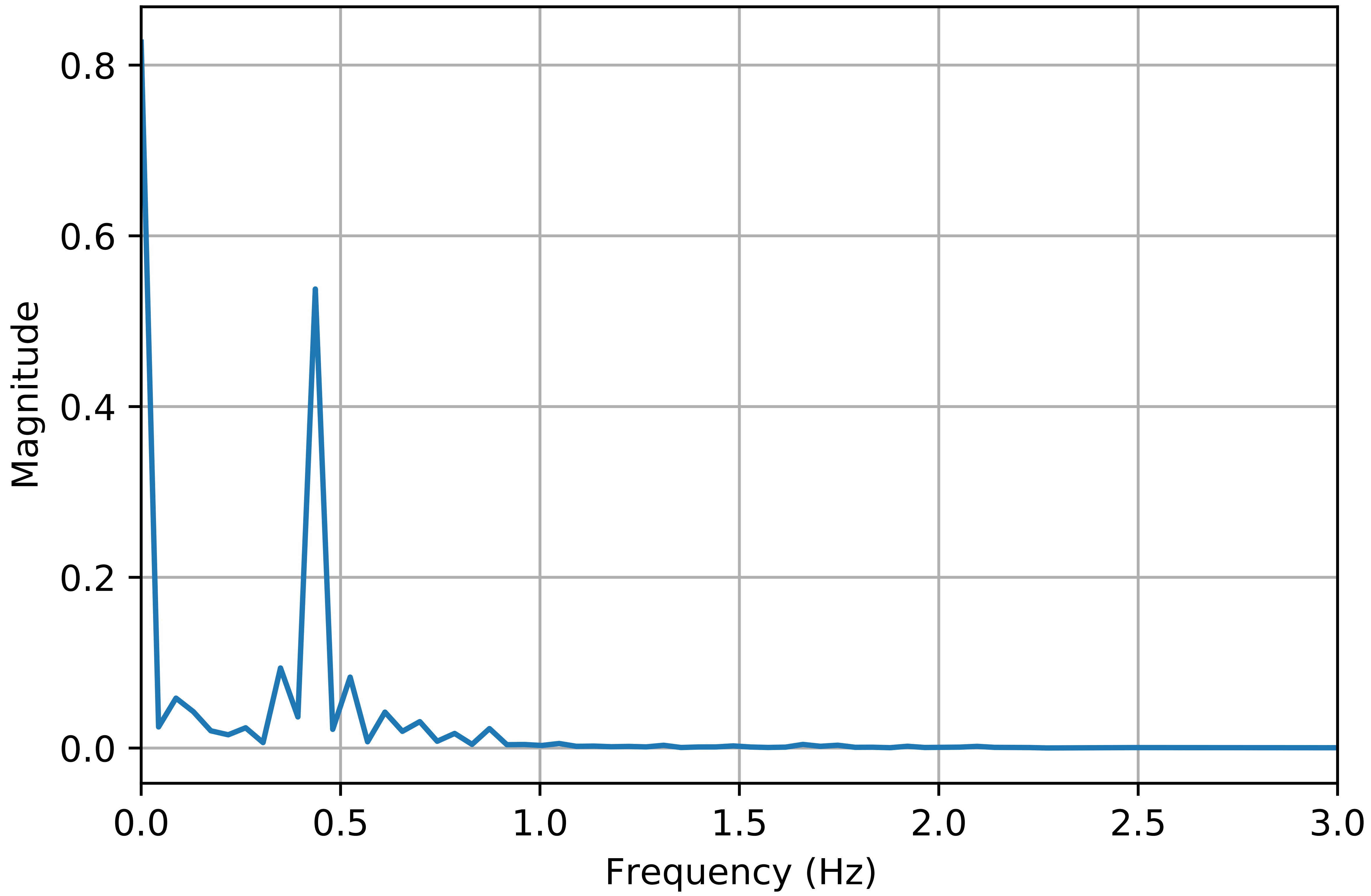
… continued on next slide

Time series visualisation

… continuing from previous slide

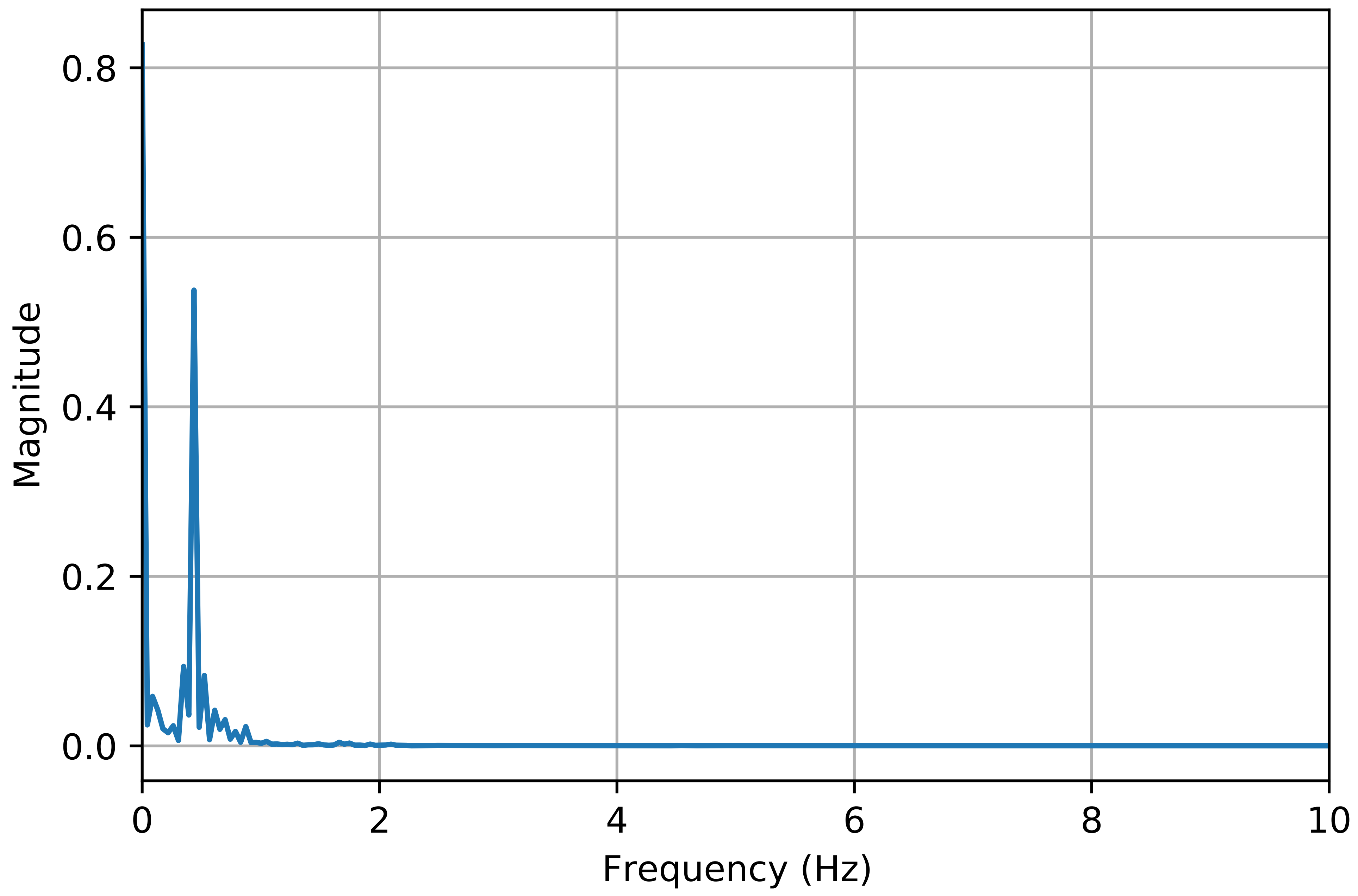
Frequency spectrum



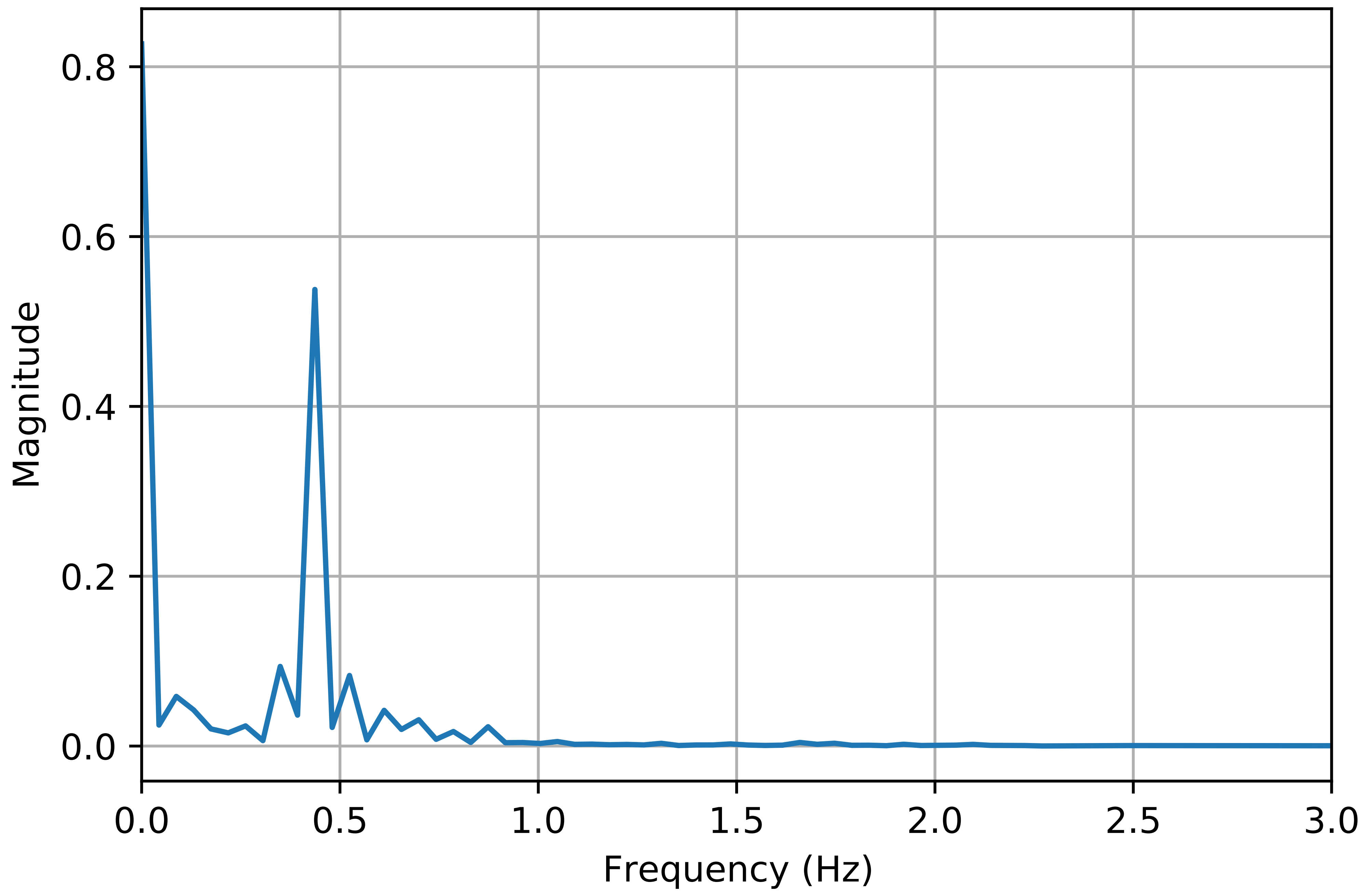


Frequency visualisation (via FFT)

Frequency visualisation (via FFT)



Frequency visualisation (via FFT)



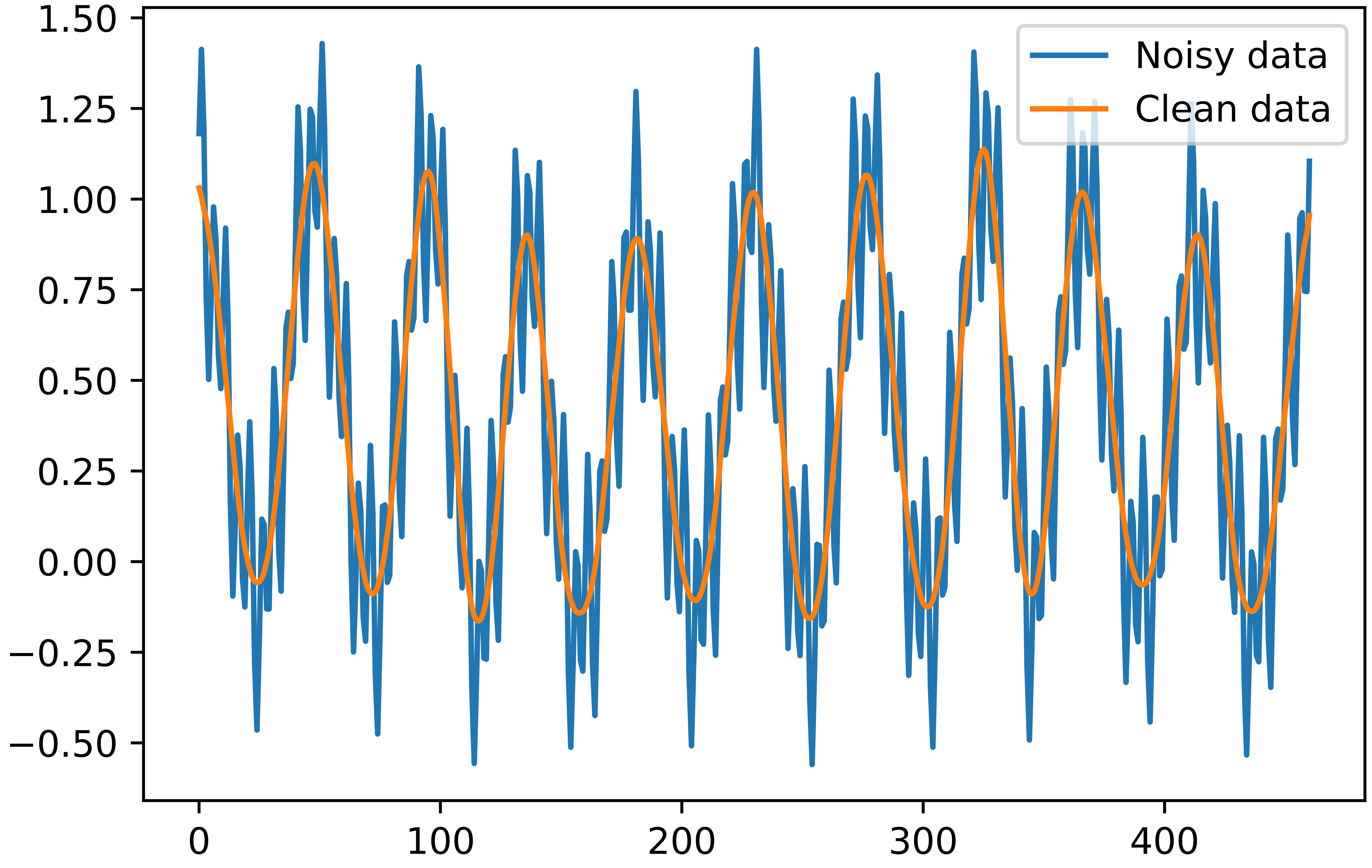
But this data doesn't mimic that collected in the real world!

Lets make it noisier

- We're going to add some artificial noise to our very clean synthetic dataset
- Add a 2Hz wave and a 4Hz wave

Artificial noisy signal

More indicative of real world data

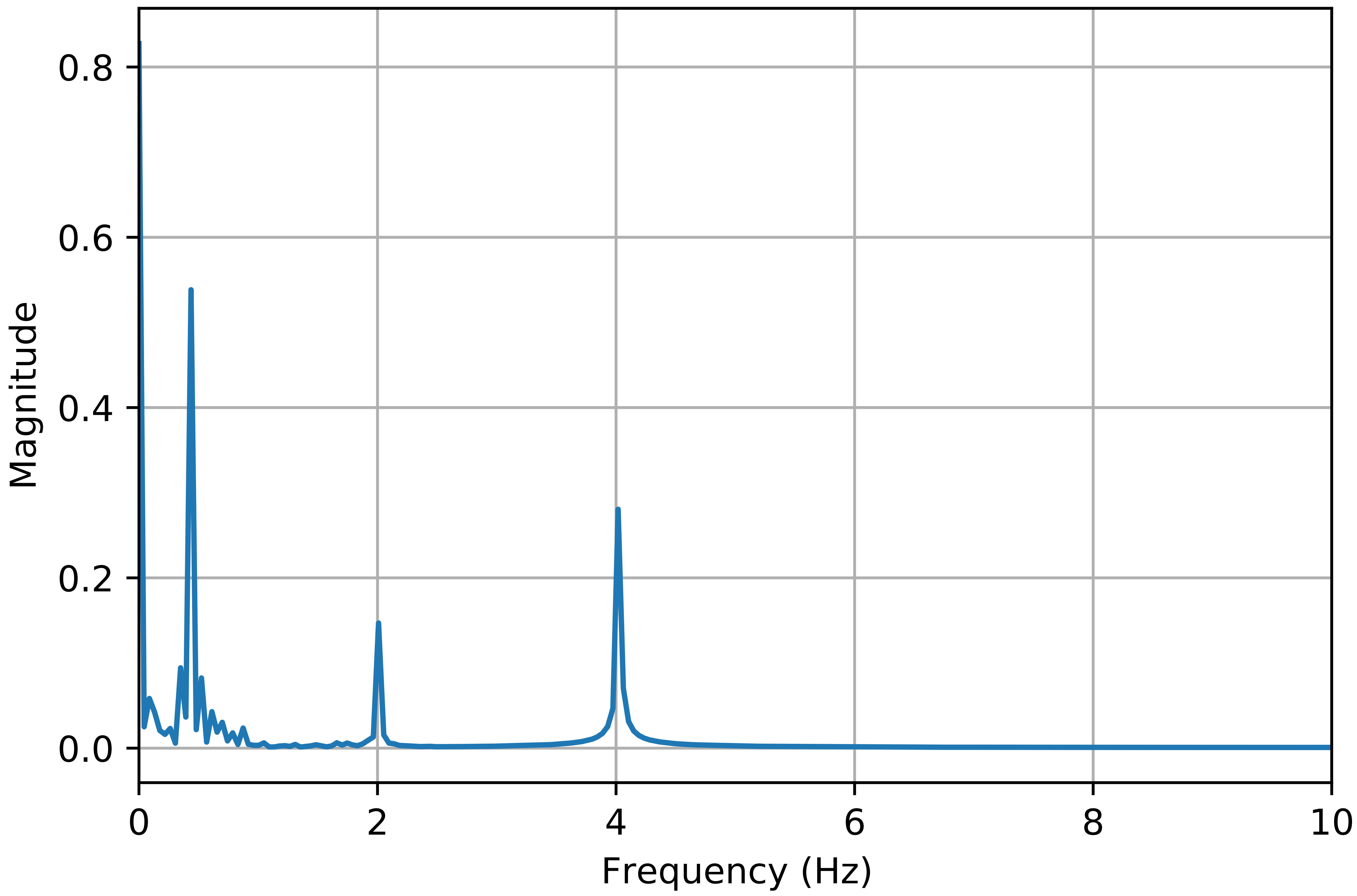


Observations

- Noisy peaks overlaid with the signal of interest
- DC offset in the signal (centred around ~0.5)

FFT (again)

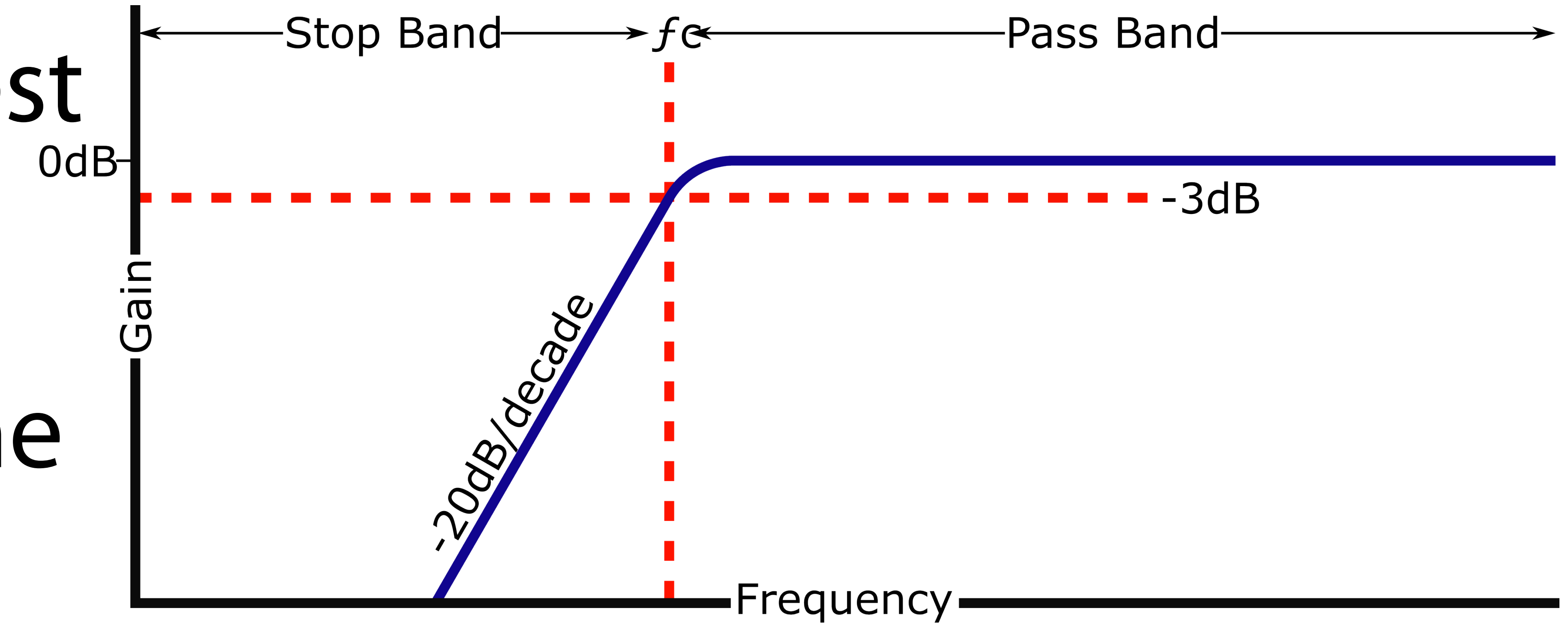
- Peaks at 2Hz and 4Hz interfering with the signal
- How can we remove the interference and the DC offset?



We’ve visualised the data.
Now lets preprocess it!

Filtering

- Remove the **DC offset** first
- **High pass** filter
- From the FFT, we can see that we have **peaks** of interest from ~**0.25Hz**
  - We need to apply the filter at a **lower cutoff** than the signals of interest
  - Try some frequencies and look at the resulting FFTs

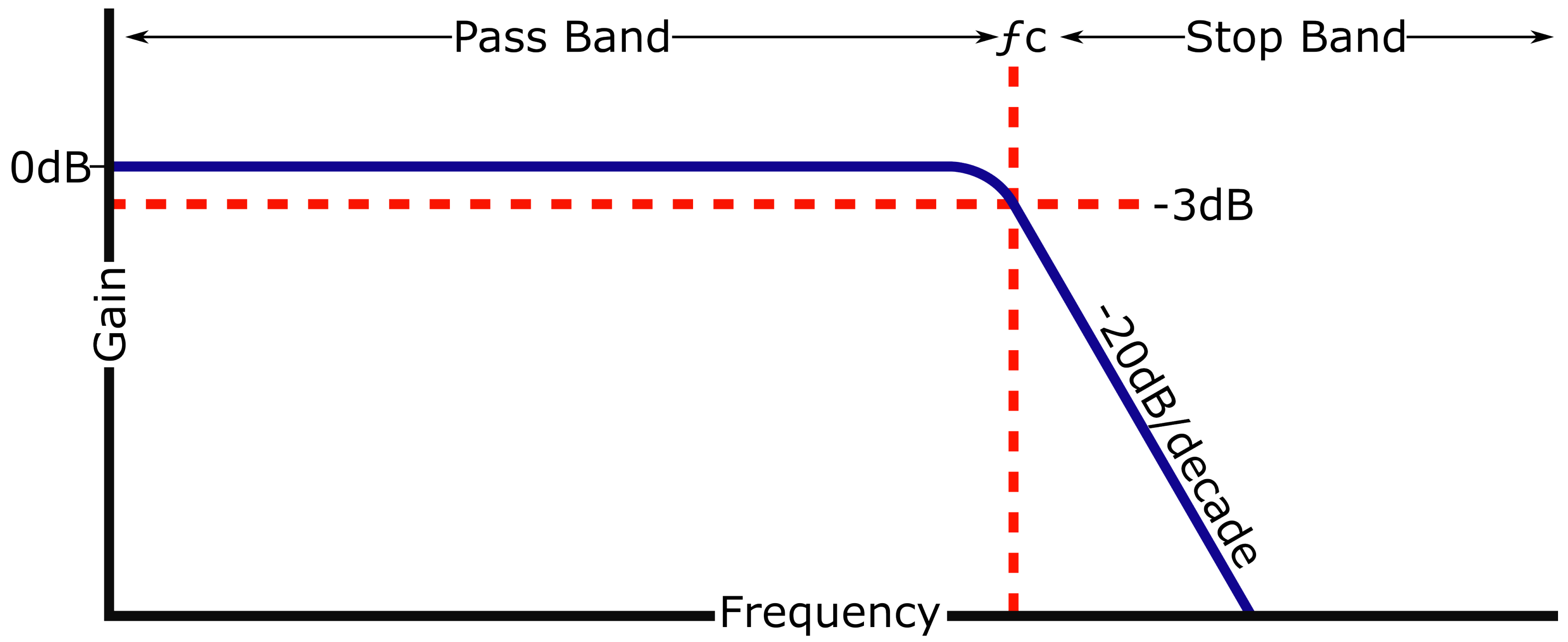


High Pass Filter

Apply High Pass Filter

Filtering

- Now lets remove the **2Hz** and **4Hz** noise
- **Low pass** filter
- From the FFT, we can see that our **maximum** peak of interest is at ~**1Hz**
    - We need to apply the filter at a **higher cutoff** than the signals of interest
    - Try some frequencies and look at the resulting FFTs

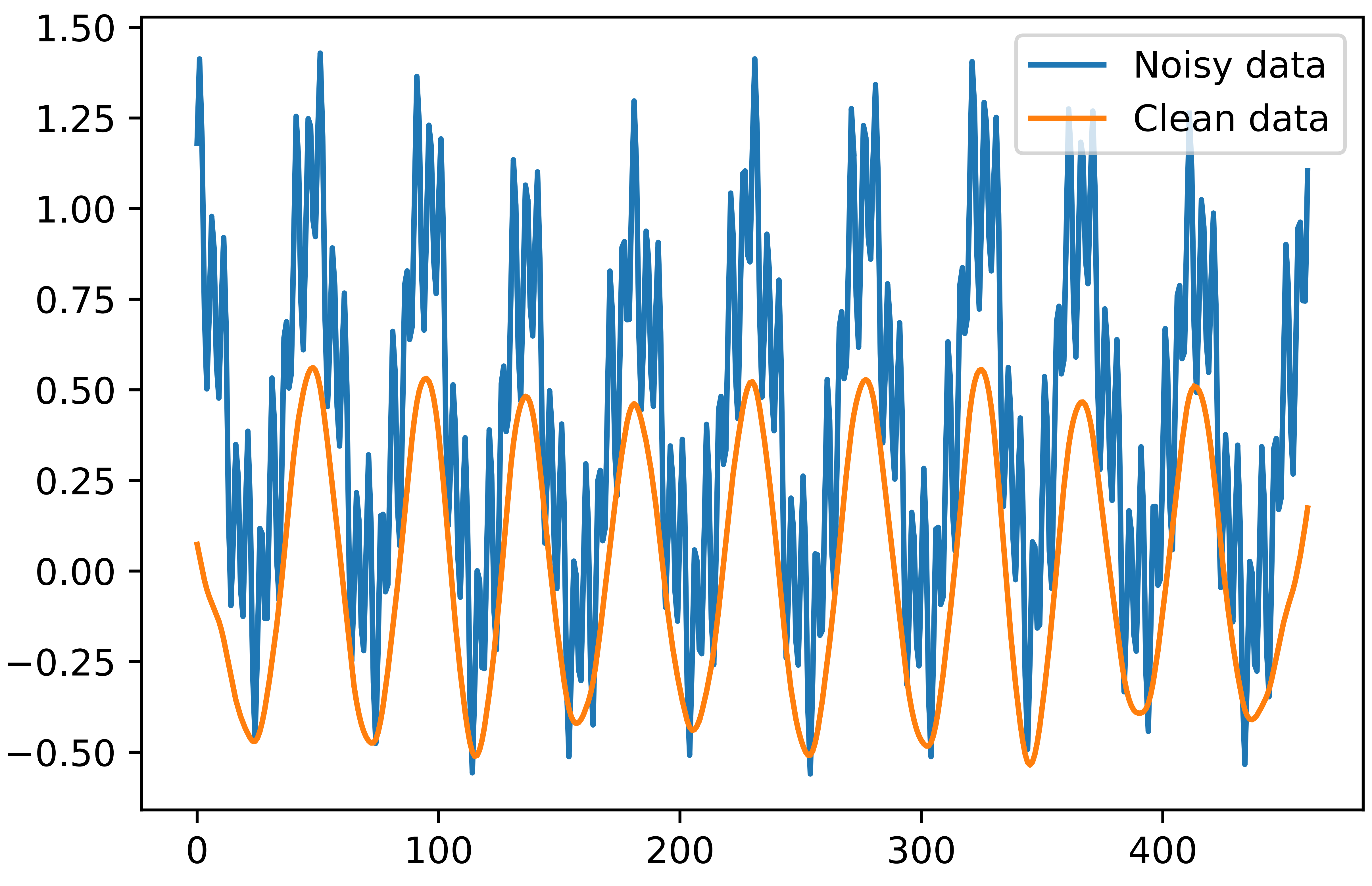


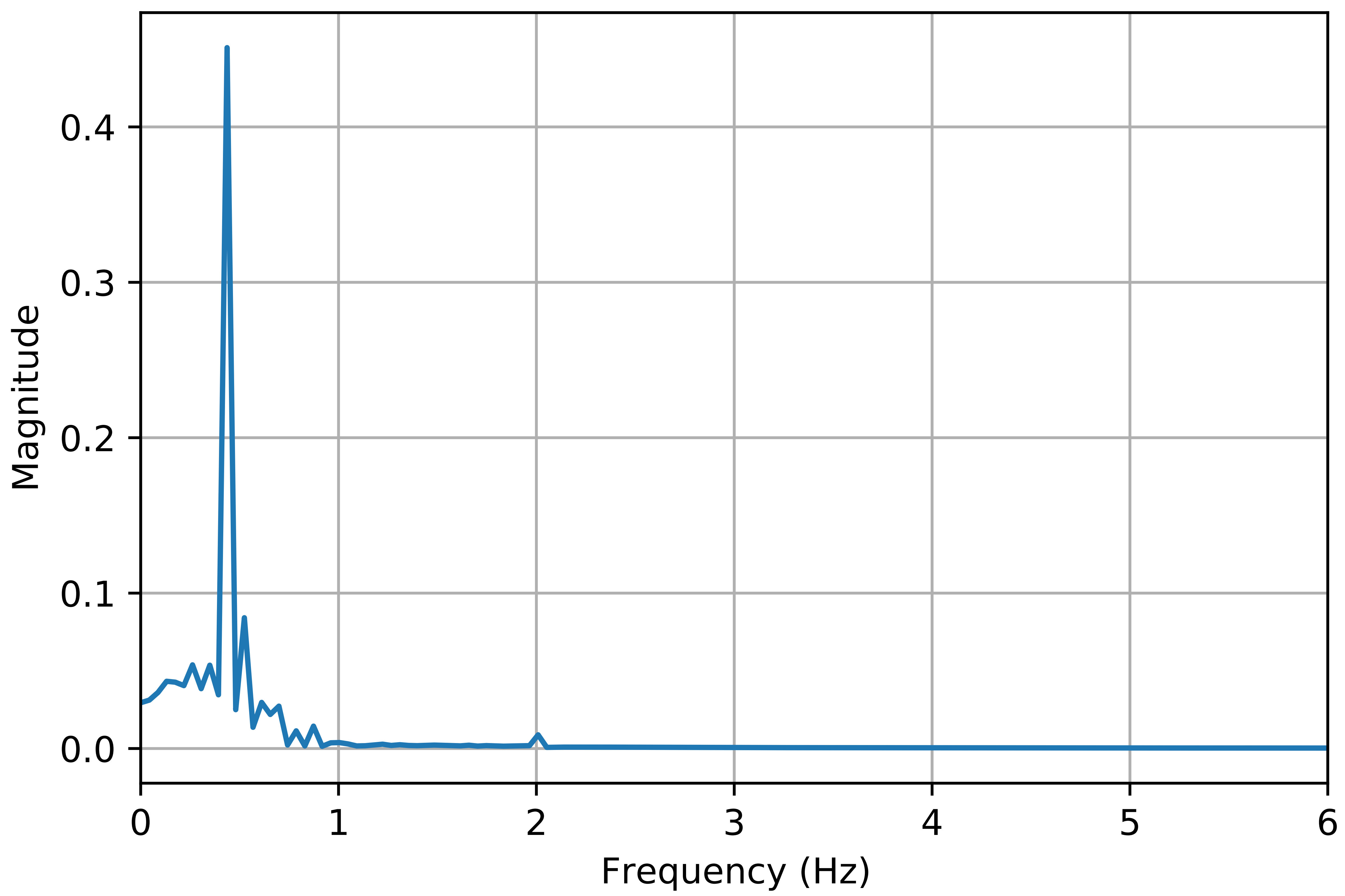
Low Pass Filter

Apply Low Pass Filter

Signal after filtering

What is this peak?





Dataset analysis — how much variability across samples?

Assignment

- Uploaded on Moodle
- Due **20 February**
- Audio dataset of **heart sounds**
- Preprocessing techniques for IMU can also be used for audio!
- Feature extraction will differ between IMU and audio