**Fast Fourier Transform**

Python: <https://pyforneuro.com/chapter7>

The FFT is an analysis method that turns EEG data from time-domain into frequency-domain.

**Electrical Activity Overview**

An EEG measures electrical activity from the brain.

Electrical activity tells us how neurons communicate with each other by detecting the electrical oscillations occurring in pyramidal cells (neurons in the cerebral cortex which are perpendicular and parallel to the scalp). Electrical oscillations come from electrical dipoles created by the activation of postsynaptic potentials, i.e., when there is a positively charged region and negatively charged region, separated by space. For instance, when excitatory input flows into the cell, that area becomes relatively more positively charged while the dendrites are more negatively charged. EEG electrodes pick up on the sum of these dipoles; because neural activity is synchronized, the signal is large enough that it isn’t cancelled out when summed. Ultimately, the sums of positive “sources” and negative “sinks” are what we see as the wave oscillations in EEG signal data.

Diagram of a tree with a diagram of a tree

AI-generated content may be incorrect. A diagram of a nervous system

AI-generated content may be incorrect.

A drawing of a human head

AI-generated content may be incorrect.

**Image Source**: Jackson, A. F., & Bolger, D. J. (2014). The neurophysiological bases of EEG and EEG measurement: A review for the rest of us. Psychophysiology, 51(11), 1061-1071. <https://doi.org/10.1111/psyp.12283>

**Fourier’s Theorem and EEG - Background**

Electrical activity is a measure of energy, and electrical oscillations are presented as waves. Fourier’s Theorem tells us that these waves are the sum of sine and cosine waves. Fourier Transform breaks down (decomposes) these waveforms into sine and cosine waves.

A graph of a function

AI-generated content may be incorrect.

<https://www.mathsisfun.com/algebra/trig-sin-cos-tan-graphs.html>

Fourier’s Theorem tells us each individual wave can be expressed in three unique terms:

Amplitude: the magnitude of the wave, in microvolts, tells us the strength of brain activity.

Amplitude can range from 15 uV – 150 uV; the average range is 30 uV – 80 uV while awake. Amplitude is used to calculate the strength or power of the electrical activity.

Frequency: the number of oscillations of the wave, in hertz, tells us the speed of brain activity.

Ranges of frequencies are bucketed into bands or bins:

|  |  |  |
| --- | --- | --- |
| **Name** | **Frequency Range** |  |
| Sub-Delta | 0 – 1 Hz | Slowest wave |
| Delta | 1 – 4 Hz |  |
| Theta | 4 – 8 Hz |  |
| Alpha | 8 – 12 Hz |  |
| Beta | 12 – 30 Hz |  |
| Low-Gamma | 30 – 80 Hz |  |
| High-Gamma | > 80 Hz | Fastest wave |

Phase: the position of the wave within its cycle at a certain time, in degrees or radians, tells us whether the brain activity is synced.

Phase data tells you whether the oscillations in different brain areas are occurring in sync, i.e., whether waves in different locations are in the same phase of their cycle at a certain point in time.

EEG research shows us that the strengths, speeds, and relative positions of electrical activity are associated with certain cognitive functions of the brain.

**Fourier Transform and EEG Data – Application**

EEG data can be processed in two domains, time and frequency.

* Time domain data gives you the amplitude over time

uV

A black line with a white background

AI-generated content may be incorrect. T

* Frequency domain gives you the amplitude over hertz or gives you amplitude over time, per hertz (Time-Frequency analysis)
  + One type of frequency (spectral) analysis is the Fast Fourier Transform

Frequency analysis:

uV

A black line with a white background

AI-generated content may be incorrect. Hz

Time-Frequency analysis:

For a certain frequency or frequency band,

uV

A black line with a white background

AI-generated content may be incorrect. T

Fourier Transform turns the time-domain activity, uV/T, into frequency-domain activity, uV/Hz *(the time-domain signal is multiplied by the cosine and imaginary sine of the EEG signal, which gives you the frequency signal).*

There are several variations of Fourier Transforms. The classic/original version considers time to be a continuous variable, i.e., time is infinite.

Because EEG data is collected within a certain timeframe, time has been defined as a finite variable, i.e., a discrete variable. The Fourier Transform that considers time a discrete variable is the Discrete Fourier Transform.

The Fast Fourier Transform (FFT) is a more efficient way of calculating the DFT and is generally used in EEG data analysis.

**FFT Prerequisites/Assumptions**

FFT can ONLY be used if 2 assumptions are satisfied:

1. Only frequency-domain information is needed
   1. Time-domain information is not needed at all
2. The EEG data is stationary/quasi-stationary
   1. The EEG data does not change over time; it is rhythmic and repetitive
   2. For example, resting state data (eyes open vs eyes closed) or SCROL data (upward vs downward comparison)

**How to run FFT in BVA**

The below examples are from BrainProduct’s webinar, *Spectral Analysis Using FFT.* In the webinar, they analyzed resting state alpha band activity, hypothesizing that there would be a significant difference between Eyes Open and Eyes Closed conditions.

1. Pre-Processing Raw Data

Apply history template for filters, ocular correction, interpolation, and remove artifacts.

* + Key pre-requisite parameters:
    - Sampling Rate (SR, Hz)
      * How many data points (samples) per second are collected from the EEG
    - Segment Length (T, seconds)
      * Duration that is processed as a single unit (epoch, window)
    - Data points (N)
      * The number of data points in a segment depends on the Sampling Rate (SR) and Segment Length (SL)
        + N = SR \* SL
      * Number of data points (N) must be a power of 2
        + This will avoid spectral leakage (explained below, see 5. Data Window)).
    - Frequency Resolution (FR, Hz)
      * The smallest frequency difference at which you would see a change in EEG data
        + 1/SL = SR/N

1. Segmentation for Stationarity/Quasi-Stationarity

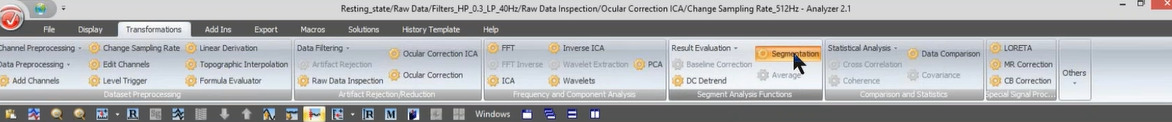
FFT can analyze stationary data, but EEG data is non-stationary by nature. We can make EEG data **quasi-stationary** by segmenting the data, i.e., by extracting portions of the signals over time (within those extracted portions, the signal is stationary).

→ Transformation tab

→ Segment Analysis Function

→ Manually set according to the desired markers in the data

E.G.: Resting State markers would entail Eyes Open (“EO”)/Eyes Closed (“EC”) markers

****

A screenshot of a computer screen

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

1. Segmentation for Frequency Resolution

Frequency Resolution tells the smallest difference in frequency with which you can detect a change in the EEG data.

Frequency Resolution = 1/T = SR/N

→ Transformation tab

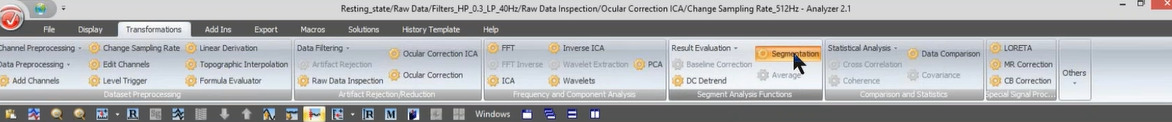
→ Segment Analysis Function

→ Divide data set into equal sized segments

E.G.: If SR = 512 Hz, then N = 1024, and T = 2 seconds

FR = 0.5 Hz

You’ll see the Hz change in 0.5 Hz increments

****

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect. Note: automatically updates N = 1024

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

You would now be able to analyze the frequency data for every 2 seconds within the Eyes Open condition.

1. FFT

→ Transformation tab

→ Frequency & Component Analysis

→ FFT

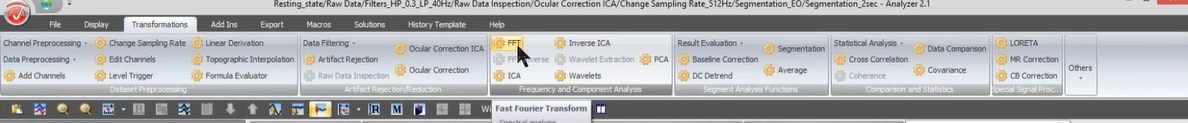
→ Set Resolution: “Other resolution”

* + - * BVA would automatically input Maximum Resolution per the SR/N calculation, but BVA gives you the option to enter the FR manually
        + Best to only decrease the resolution if manually changing this is necessary; otherwise, best to use Maximum Resolution

If the FR is increased, the data points will need to be interpolated and the spectral data will not be accurate.

→ Output Options:

* + - * Voltage = uV
        + If you want the amplitude at each frequency bin
      * Voltage Density = uV/Hz
        + If you want to compare amplitudes of difference signals
      * Power = uV2
        + If you want power at each frequency bin
      * Power Density = uV2/Hz
        + If you want to compare power across different signals
      * Produce Complex Output = *imaginary* *+ real output* (ie, *Fourier Coefficient*)
        + If you want to conduct further analysis to find phase data
      * Full Spectrum = obtains a one-sided spectrum / half-spectrum folded over



A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

1. Data Window

When segmenting data, the data gets cut off at points such that there are abrupt changes between segments (discontinuities), which could misrepresent the signal content. This is called spectral leakage. Applying a window function to the data segments limits discontinuities by tapering the data at the edges of the segment to 0. Windowing solves the spectral leakage problem.

No spectral leakage:

This depicts the stationarity assumption of FFT

A graph of a function

AI-generated content may be incorrect.

A graph of a function

AI-generated content may be incorrect.

Spectral leakage:

A graph of a function

AI-generated content may be incorrect.

A graph of a function

AI-generated content may be incorrect.

→ Transformation tab

→ FFT

→ Window:

→ Data Window: **ER LAB = Hanning Window**

* + - * Hanning: edges of the wave touch 0
      * Hamming: edges of the wave touch near 0

→ Window Length: **ER LAB = Input 10%**

* + - * 0% = no window, data on the edges of the wave aren’t suppressed to 0

A screenshot of a computer

AI-generated content may be incorrect.

* + - * 100% = data is super-suppressed, and emphasis is on middle of window

A screen shot of a graph

AI-generated content may be incorrect.

* + - * What is %age determined by ? …

A screenshot of a computer

AI-generated content may be incorrect.

1. Select the Electrodes of Interest (EOIs)

Double click to view the EOIs only

Scale up Amplitude to adjust visuals

A screenshot of a computer

AI-generated content may be incorrect.

1. Average the Data

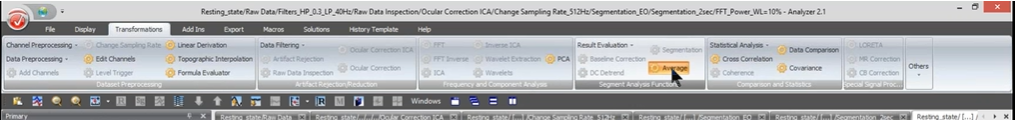
→ Transformation tab

→ Segment Analysis Function

→ Average

→ Use Full Range

* + - * This gives you average across all segments



A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

**NOTE**: In this example, we would need to run FFT on the Eyes Closed (“EC”) as well. (47 min)

Right click on Segmentation\_EO

→ Edit Parameters / Copy

→ Use the same Segmentation options as previously used

→ Start and End markers are now EC\_start and EC\_stop

Rename nodes appropriately

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer screen

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer screen

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

To compare conditions of interest, overlay the outputs

E.G., drag the node for Average\_Power\_EC onto the open window, Average\_Power\_EO

A screenshot of a computer

AI-generated content may be incorrect.

1. Data Extraction
   1. Generic Data Export

This gives you output for each frequency bin

→ Export

→ Node Export

→ Generic Data

* 1. Area Information Export

This gives you output for specific ranges or peaks

→ Export

→ Multiple Export

→ Area Information

1. Post-Processing (details TBD)
   1. Phase

This will let you compare phase data

→ FFT Settings:

* + - Voltage (uV)
    - Produce Complex Output
    - No Window

→ Exported Data:

* + - Provides complex output for each bin of the Frequency Resolution

A screenshot of a table

AI-generated content may be incorrect.

→ Computing Phase Values = Matlab Transformation

* + - atan2 = radians
    - atan2d = degrees
  1. Normalization

This gives you normalized power relative to:

* + A specific frequency band
  + The total spectrum (all frequency bands)

→ FFT Settings:

→ Power (uV2)

→ Use Full Spectrum

→ Segment Normalization: Normalize Segments

→ Interval Start/End: set manually (Hz)