MI BMI | Logarithmic band power

Lab 04

DIPARTIMENTO
DI INGEGNERIA
DELL'INFORMAZIONE

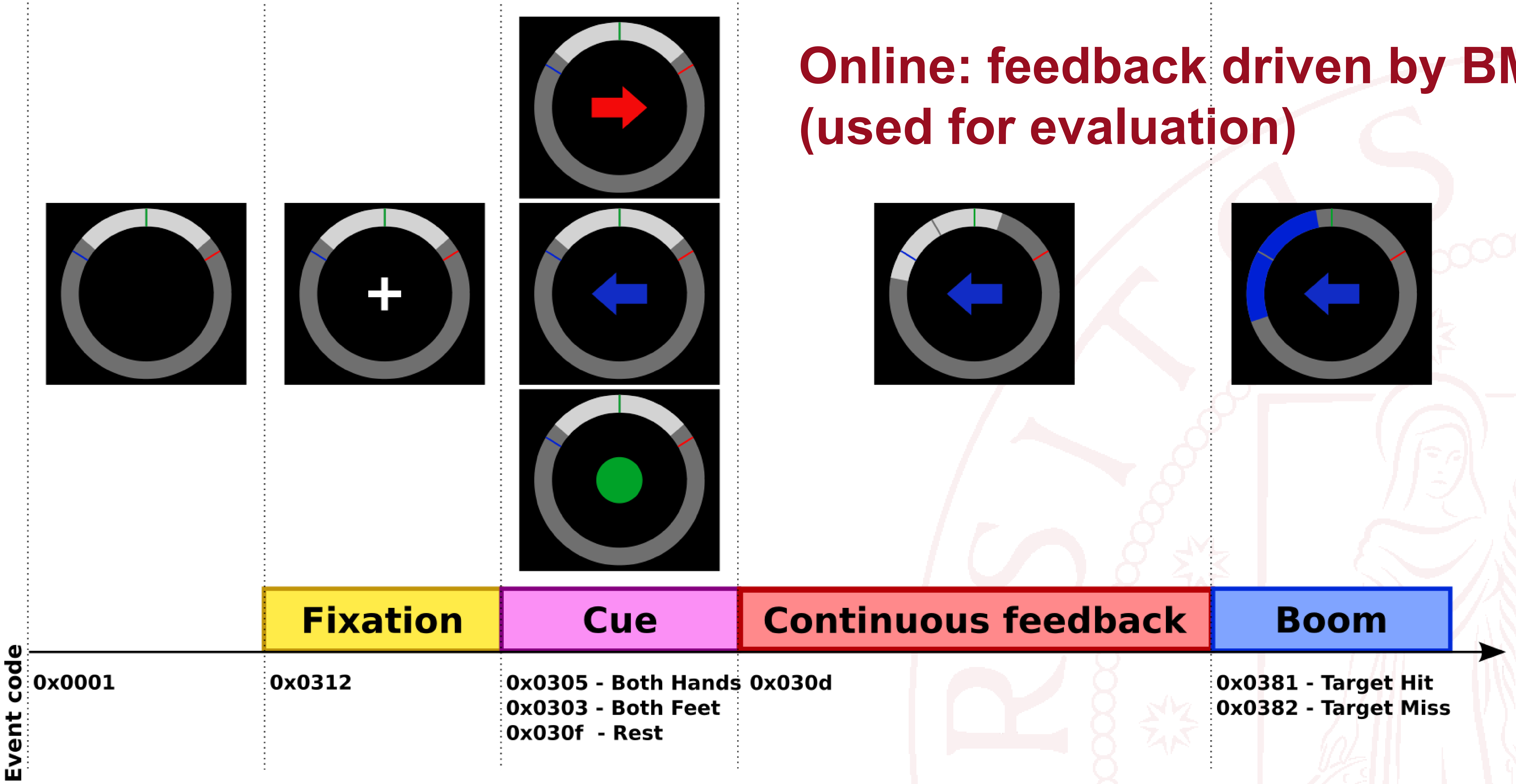
MI protocol of the provided data

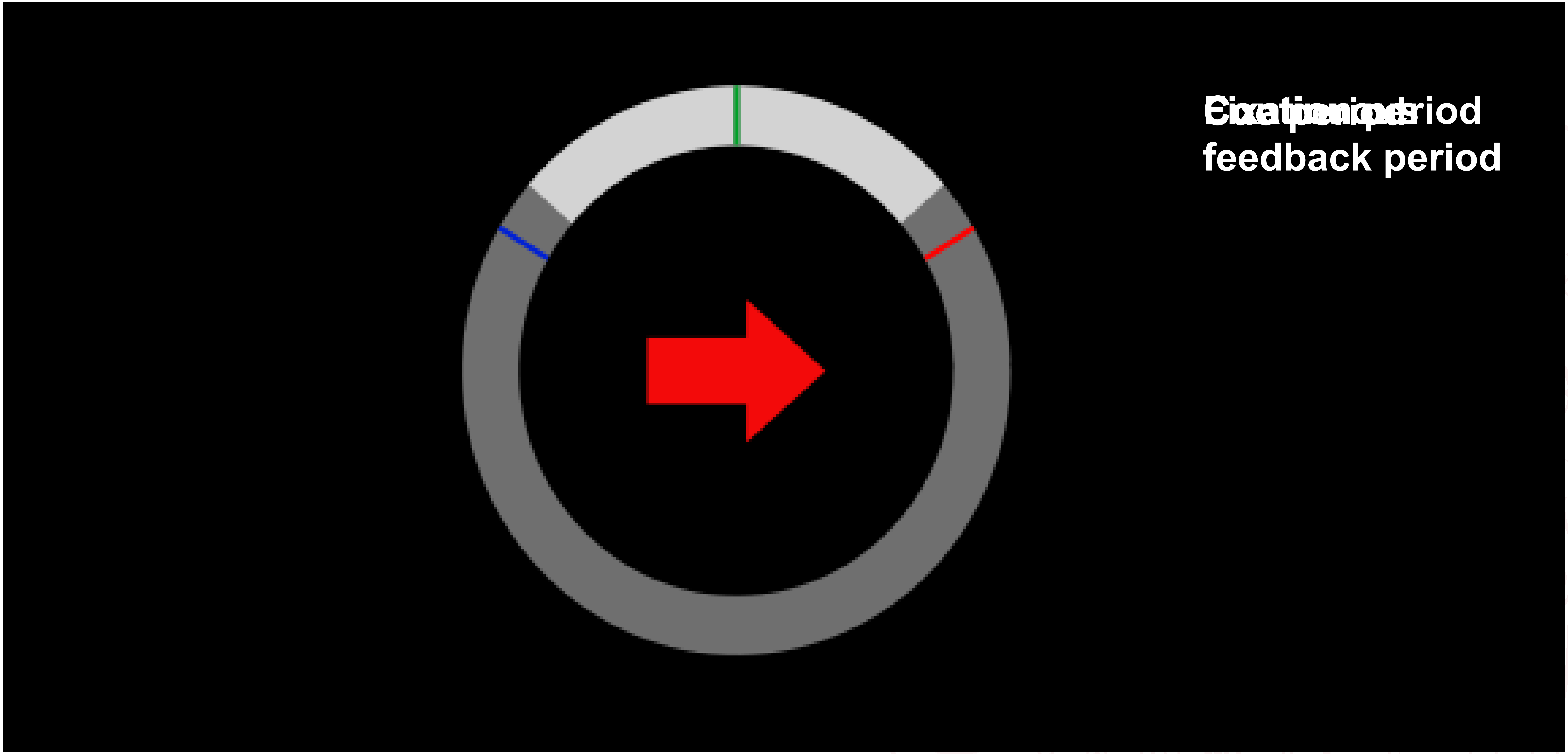
DIPARTIMENTO
DI INGEGNERIA
DELL'INFORMAZIONE

MI BMI | Data protocol

Offline: positive feedback towards the target (used for initial calibration)

Online: feedback driven by BMI (used for evaluation)




feedback period

MI BMI | GDF Event codes

Code	Description	Decimal value
0x0001	Trial start	1
0x0312	Fixation cross	786
0x0305	Both Hand	773
0x0303	Both Feet	771
0x030F	Rest	783
0x030D	Continuous feedback	781
0x0381	Target hit	897
0x0382	Target miss	898
0x8000	Event OFF	/

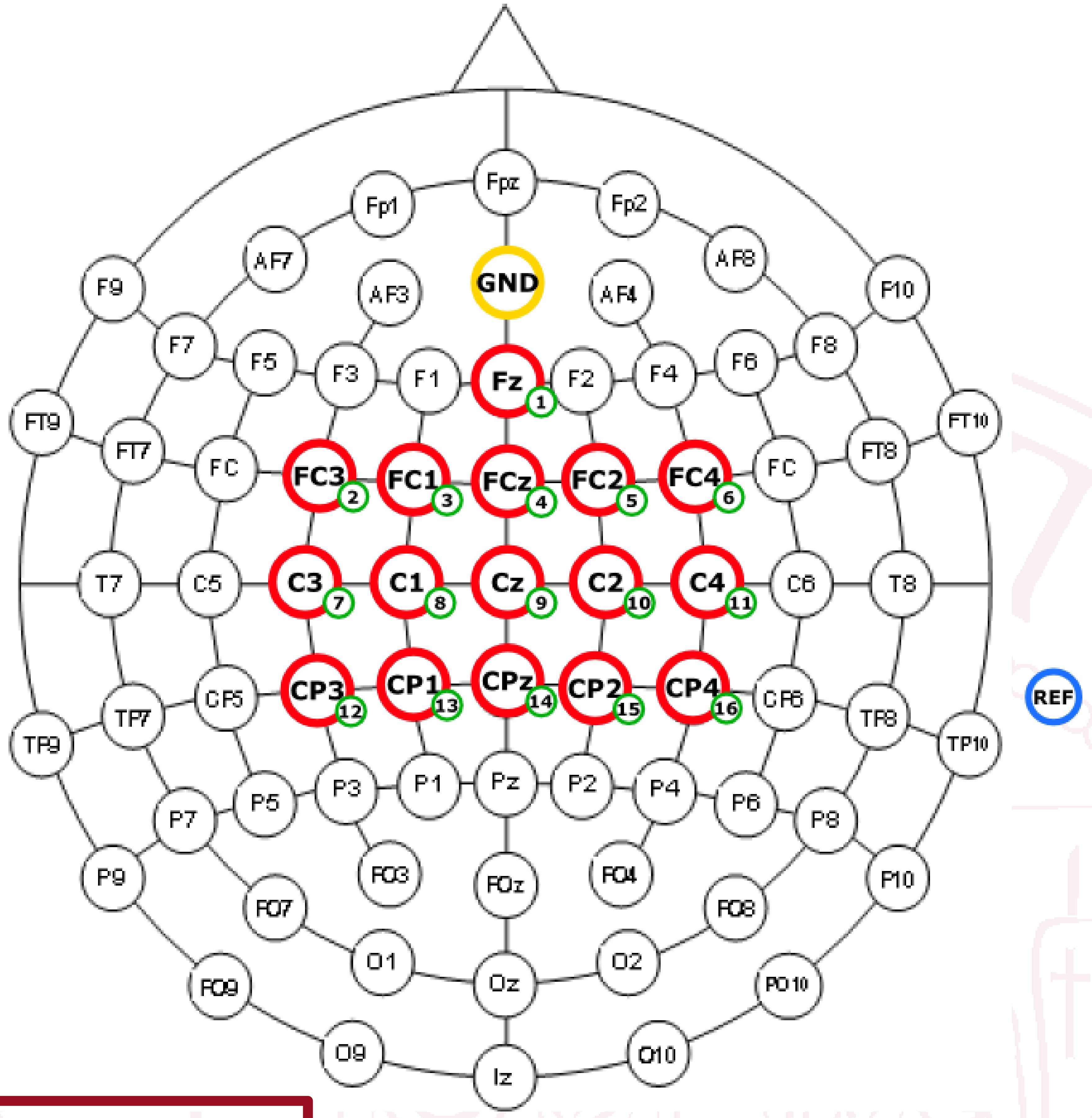
Each event is ‘’closed’’ by adding to its value **EVENT_OFF** (e.g., 0x0312 → 0x8312)
This is used for computing the **duration** of the event (automatically by `sload`)

MI BMI | Experimental design

- **16 EEG channels** over motor cortex
- Sample rate: 512 Hz

- **1 subject**
- **1 session**
- **3 offline runs**
- **4 online runs**

- **2 classes for offline** (Both hands, Both feet)
- **3 classes for online** (Both hands, Both feet, rest)



Terminology:
- session: [mounting cap ←→ removing cap] (usually = day)
- run: subject performing a group of trials (usually = file)
- trial: period of time when the subject is performing a specific task
- offline modality: the feedback is moving automatically towards the target
- online modality: the feedback is moved by the output of the BMI

MI BMI analysis: band power

How to compute signal power? There are several methods:

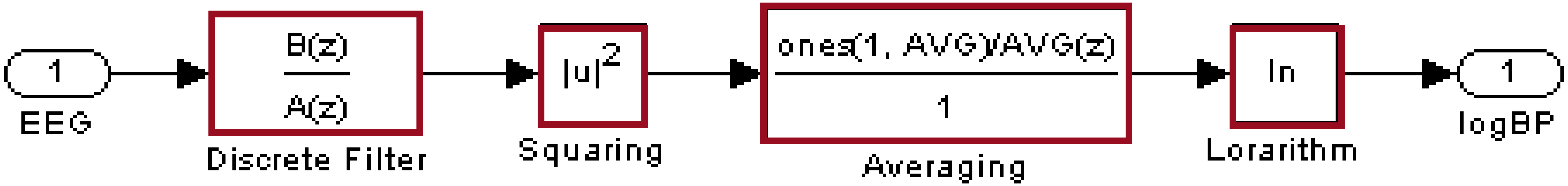
- Logarithmic band power
- Power spectral density (PSD)
- Spectrogram
- Wavelets
- Autoregressive model (AR)

Processing for logarithmic band power

After importing, concatenating and labeling the data, you have:

- data matrix `s: [samples x channels]`
- header struct h, with all the events correctly concatenated

- The following process can be applied **on all your data**



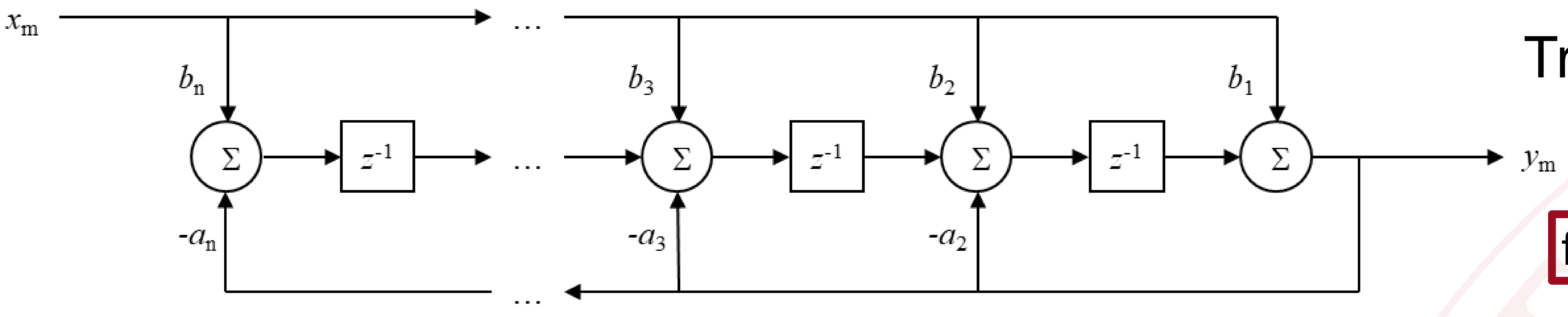
1. Filter the signal (band pass)
   - in the μ band (~10-12 Hz)
   - In the β band (~18-24 Hz)
   - Butterworth filter (order 5)
2. Squaring
3. Calculate the average over 1 second
4. Logarithm transform

Then, you can extract, trials, channels, periods of interest

Caveats on filters

In MATLAB, a filter is designed as:

$$Y(z) = \frac{b(1) + b(2)z^{-1} + \ldots + b(n_b+1)z^{-n_b}}{1 + a(2)z^{-1} + \ldots + a(n_a+1)z^{-n_a}} X(z),$$



Transfer function in Z domain

filter order

normalized frequencies [0 1]

```
>> [b, a] = butter(N, [W1 W2],…);
>> help butter
```

In hertz:
`2*[W1 W2]/SampleRate`

```
>> sfilt(:, 1) = filtfilt(b, a, s(:, 1));

>> sfilt(:, 1) = filter(b, a, s(:, 1));
```

Two steps for creating a filter in MATLAB:

- Find b and a coefficients
- Apply the filter

How to check if the filter is stable?

```
>> fvtool(b, a, 'Fs’, SampleRate)
```