

Neuromorphic BMI – Analysis tool guide

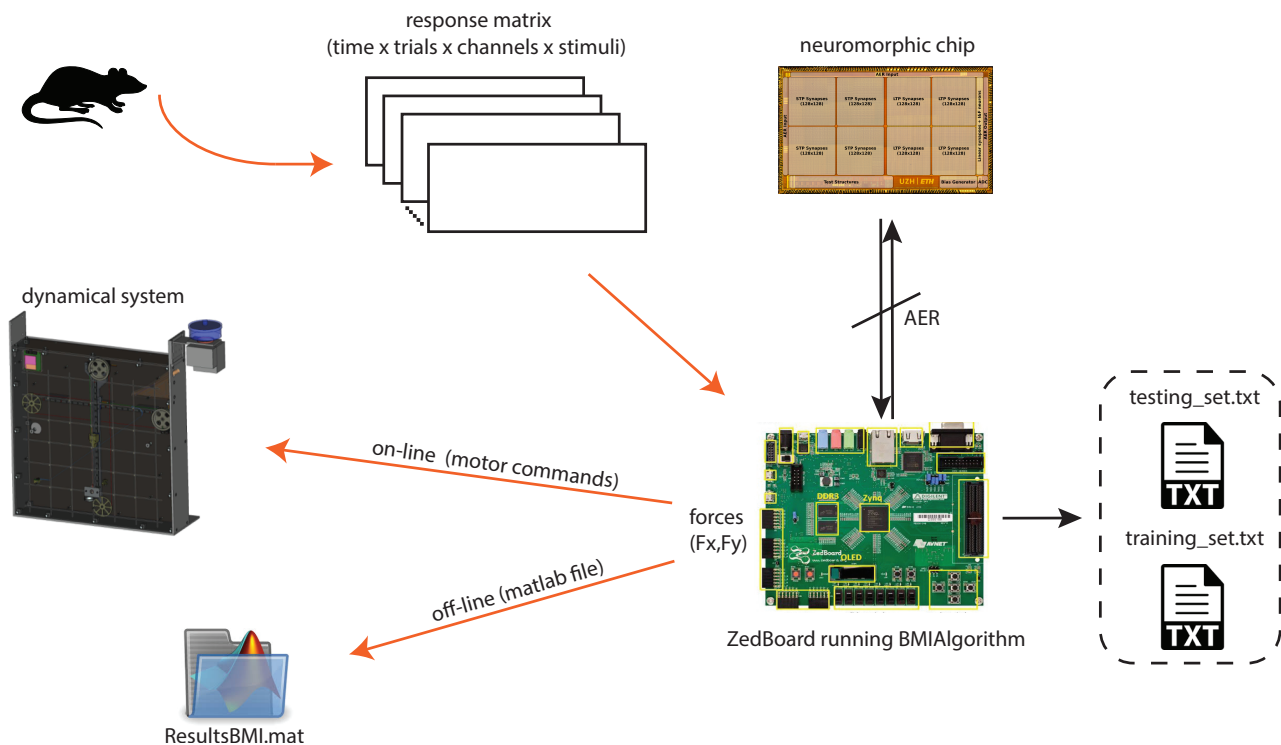


Figure 1 Data flow diagram

Neural data recorded from the brain are collected and packed into a four dimensional binary Matlab matrix (Response Matrix) with dimension $\#timeBin \times \#trials \times \#recChannel \times \#stimuli$ (in our case $256 \times 50 \times 15 \times 4$).

This structure is stored into the ZedBoard memory that splits it into two subsets: a training and a test set. The first one contains 80% of the trials belonging the original dataset; the remaining 20% are associated to the test set. Initially, through the AER connection, the ZedBoard communicates the training set to the neurochip that activates its learning protocols in order to accomplish its decoding task. Once the training is finalized ZedBoard sends the testing set that is decoded on the basis of the previous training. All the trials decoded from the neuromorphic chip are then collected from ZedBoard and saved into .txt files (one per each set: respectively training_set.txt and testing_set.txt).

Once ZedBoard stored all the decoded responses it starts the BMI Algorithm routine that, as described in the paper “**A bidirectional brain-machine interface featuring a neuromorphic hardware decoder**”, initialize the dynamical system in one of the eight predefined initial positions and runs it with the aim to reach a central target region by means of the just decoded neural signals. Every time ZedBoard sends a 2D Force that drives the external device it also sends a copy of it via Ethernet to a connected PC where a Matlab scripts collects both the just applied force and the position of the mobile cart. All these info are then collapsed in a unique structure named *ResultBMI.mat*.

In the following section we’re going to introduce how to analyze these data with our Matlab scripts in order to reproduce the results presented in “**A bidirectional brain-machine interface featuring a neuromorphic hardware decoder**”.

ResultsBMI.mat

This .mat structure contains the following field:

- *trjON*: cell array with dimension [#trial per starting position x #starting positions]. Each cell contains a vector with size [4 x #number of position collected during trajectory] representing a single trajectory. First two rows indicate the (x,y) coordinates of the mobile cart time by time, while the latter two rows report the velocity (vx,vy) of the system in the related point. These trajectories have been obtained with the Encoder ON.
- *trjOFF*: same as *trjON* but with the encoder turned OFF.
- *forceON*: cell array with dimension [#trial per starting position x #starting positions]. Each cell contains a vector with size [#of steps x 2] representing the forces applied to the system during a single trial. First column represents the x-component of the force applied to the mobile cart step by step, while the second column report the y-component. These forces have been obtained with the Encoder ON.
- *forceOFF*: same as *forceON* but with the encoder turned OFF.
- *testForcesBMI*: 3D matrix with size [#test trial per stimulus x #stimuli x 2] representing the whole testing set after the decoding stage performed by the neuromorphic chip.
- *recForce*: bidimensional matrix with size [#stimuli x 2] used to translate the chip output (reported in *testing_set.txt*) into a 2D force applicable to the dynamical system.
- *opts*: this struct contains a series of parameter used by the BMIAlgorithm running on the ZedBoard useful for the analysis tools.

Matlab Functions

- *frontiersMain*: main script useful to set the dataset name that has to be analyzed. This is the script that has to be run in order to perform all the analysis.
- *frontiersAnalysis*: function that lists and manages all the analysis that has to be performed on the selected dataset.
- *prepareDatasetAnalysis*: this function retrieves the data collected in the dataset returned by the ZedBoard and manages them in order to be used by all the analysis tools. It also performs some basic statistics about the convergence rate and the number of steps.
- *computeCovarianceAndMean*: this function computes all the main statistic about the trajectories (i.e. covariance between trajectories starting from the same initial position and their mean). It also plots the related results into two figures (this function produces figures SA and SB, from these 2 figures we also built up figure 8A).
- *rosePlots*: this function returns the rose plots associated to all the testing forces.
- *computeCP*: this is the most complex analysis that computes the *within-trajectory variance* (wtv) and the mean number of steps of all the generated trajectories. It also produces figures 8C and 8D of the manuscript.
- *computeFD*: this function splits each force produced by the BMI into its directive towards the target component and its perpendicular one. It produces figure 8E.
- all the other functions are just graphical tools useful for producing the figures introduced above.

How to use the code

- unzip the file downloaded from <http://www.sicode.eu/results/software.html> in section “A bidirectional brain-machine interface featuring a neuromorphic hardware decoder”
- Open Matlab and browse to the folder *MatlabAnalysisScript*
- Open *frontiersMain.m* and run the script

IMPORTANT: the estimated computational time is about one hour, but it could require more time depending on the pc hardware. If you want to reduce significantly the required time please comment line 48 in the *frontiersAnalysis.m* (*computeCP* function). Pay attention that in this way figures 8C and 8D will not be produced.