Project Title: System Verification and Validation Plan for EEGSourceLocalizer Software

Leila Mousapour

December 21, 2020

1 Revision History

Date	Version	Notes
Nov 2, 2020	1.0	First version of system VnV plan
Dec 18, 2020	2.0	Revised version based on feedback and issues

Contents

List of Tables

2 Symbols, Abbreviations and Acronyms

symbol	description
SRS	Software Requirements Specification
VnV	Verification and Validation
VnVP	Verification and Validation Plan
VnVR	Verification and Validation Report
MG	Module Guide
MIS	Module Interface Specification
EEGSourceLocalizer	Electroencephalogram Source Localization Software
EEG	Electroencephalogram
BCI	Brain-Computer Interface
SL	Source Localization

For completion, please see SRS Documentation at https://github.com/LeilaMousapour/Brain-Computer-Interface-/blob/master/docs/SRS/SRS.pdf

This document provides the roadmap of the Verification and Validation (VnV) plan for EEGSourceLocalizer. The purpose of the VnV Plan is to identify the activities that will establish compliance with the requirements (verification) and to establish that the system will meet the design purpose (validation). The general information is introduced in section 3. Verification plans and test description are in section 4 and section 5, respectively.

3 General Information

3.1 Summary

EEGSourceLocalizer software is going to be tested by this VnV plan. This software is designed estimate the activity of every voxel of the brain in time based on the electric potentials recorded from the scalp in the form of EEG signals by means of source localization (SL) algorithms. This document specifies verification methods that the system requirements have been fulfilled (whether the system was built right) as well as validation plan that the developed system effectively achieves its intended purpose (was the right system built).

3.2 Objectives

The purpose of the validation plan is to define how system validation will be performed at the end of the project—the strategy that will be used to assess whether the developed system accomplishes what it was designed to do. This essentially implies assessing whether the system meets the goals, objectives, and user needs stated at the beginning of the project which is in case of scientific computing softwares it is usually done based on real-world data.

Also, the verification plan includes test strategies, definitions of what will be tested, and a test matrix with detailed mapping connecting the testing performed to the system requirements. This verification plan ensures that all requirements specified in the System Requirements (SRS) document have been met and reviewed. Specific goals of this document are to build confidence in the software correctness and demonstrate adequate usability.

The 2 criteria that differ among SL techniques are the "accuracy" and

"complexity". We generally want to improve accuracy while the complexity level of the solution does not increase significantly and this creates a a trade-off problem. In this software, by controlling for the complexity factor by choosing the algorithms and considering several assumptions and the most important objective is to acquire exact activity of the brain sources. Thus, the verification and validation plan is mostly build around this idea.

3.3 Relevant Documentation

All the documents for the EEGSourceLocalizer software are provided in following list and can be accessed in the Github repository of the software

- SRS for EEGSourceLocalizer
- MG and MIS design documents for EEGSourceLocalizer
- VnVR for EEGSourceLocalizer

4 Plan

4.1 Verification and Validation Team

- Leila Mousapour (Developer and Domain Expert)
- Dr. Spencer Smith (Domain Expert)
- Gabriela Sánchez Díaz (Domain Expert)
- Shayan Mousavi Masouleh (Secondary Reviewer)
- Naveen Ganesh Muralidharan (Secondary Reviewer)
- Siddharth Shinde (Secondary Reviewer)

4.2 SRS Verification Plan

The SRS document will be verified through static technique of document inspection. Therefore, reviewers including Dr. Smith, the "Scientific Software Development" course instructor and other classmates from the class will review and inspect the document. Reviewers can give feedbacks and revision

suggestions to the author by creating issues on GitHub. It is author's responsibility to check the submitted issues regularly and make necessary revisions. The reviewers will assess this document based on the SRS check list, designed by Dr. Smith.

4.3 Design Verification Plan

The design documents, including the module guide (MG) and module interface specification (MIS) will be verified through static technique of document inspection. Therefore, reviewers including Dr. Smith, the "Scientific Software Development" instructor and other classmates from the class will review and inspect the document. Reviewers can give feedbacks and revision suggestions to the author by creating issues on GitHub. It is author's responsibility to check the submitted issues regularly and make necessary revisions. The reviewers will assess this document based on the MG check list and the MIS check list designed by Dr. Smith.

4.4 Implementation Verification Plan

EEGSourceLocalizer will be verified using several types of dynamic and static testing as follow:

- Dynamic testing for system verification: the following approaches are used:

    1. Test cases using EEG data which the expected output of SL is known: for example using simultaneous EEG/fMRI datasets which the expected sources for the EEG source localization are known based on the fMRI images.
    2. Methods of manufactured solution: this is the simplest and the first test for verification of the implementation which we synthesis EEG data from 1 or multiple dipoles using the head model (forward model) and then we feed the synthesized EEG from the known sources to the code. We know what the answer is as we manufactured it and we can confirm if the software is working.
    3. Parallel testing: "Brainstorm" software which is an open source MATLAB software can perform EEG source localization with dif-

ferent techniques. Thus, the result of obtained from EEGSource-Localizer can be compared again the Brainstorm pseudo-oracle.

- Static testing for system verification:
  - Code Walkthrough: The developer would walk an audience through the code slowly and explain every part of the code so that if there is a mistake, especially a logical error due to the steps taken for SL implementation will be discovered. The candidate audience for the code walkthrough would be the undergraduate student who is helping on this project.
    A meeting will be set with the undergraduate students. Before the walkthrough meeting, they will be asked to read through the code and write down questions about the parts that are not clear for them. During the meeting, the developer will note down all the defects and issues revealed and finally a review report with a list of findings will be documented to be addressed. This review report will be included in the VnVR as well.
- Dynamic testing for unit verification:
  All modules are to be unit tested to ensure correctness and dynamic unit test will be carried out. The MATLAB® unit testing framework supports three test authoring schemes:
  - Script-based unit tests: Write each unit test as a separate section of a test script file. We can perform basic qualifications, access the diagnostics that the framework records on test results, refine the test suite by selecting the tests we want to run, and customize the test run by creating and configuring a TestRunner object.
  - Function-based unit tests: Write each unit test as a local function within a test function file. Function-based tests subscribe to the xUnit testing philosophy. In addition to supporting the functionality provided by script-based tests, function-based tests give we access to a rich set of test authoring features.
  - Class-based unit tests: Write each unit test as a Test method within a class definition file. In addition to supporting the functionality provided by script-based and function-based tests, class-based tests provide you with several advanced test authoring features and give we access to the full framework functionality.

Script-based unit testing will be used for unit verification of this software. More details on unit testing can be found in the Unit Testing Plan document.

4.5 Software Validation Plan

In scientific software development, validation is referred to testing the software with a real-world experiment data and investigate if the mathematical modelling, assumptions etc. are valid for this real-world problem. In the beginning of 2020, a dataset was published in Nature journal under the name of "Simultaneous human intracerebral stimulation and HD-EEG, ground-truth for source localization methods" that comprises EEG recorded electrical activity originating from precisely known locations inside the brain of living humans Mikulan et al. (2020). High-density EEG was recorded as single-pulse biphasic currents were delivered at intensities ranging from 0.1 to 5 mA through stereotactically implanted electrodes in diverse brain regions during pre-surgical evaluation of patients with drug-resistant epilepsy. Thus, this dataset can be use as a test for validation as well as verification for the software.

5 System Test Description

5.1 Tests for Functional Requirements

The functional requirements stated in the SRS document can be categorized in two main sections: FRs related to inputs (R2 and R3) and FRs related to outputs of the EEGSourceLocalizer software. Thus, the FR test is explained in two areas below.

5.1.1 Area of Input Testing

As described in the section 4.2.6 in SRS, EEGSourceLocalizer shall verify whether the inputs meet the data constraints (R2). Frequency component of the EEG data should be checked to see if the data is band-pass filtered. Also, EEGSourceLocalizer should ask the user to confirm if the artifact removal has been done. Additionally, EEGSourceLocalizer shall verify that the covariance matrix of the input data is not rank-deficient (R3). An error message shall

be displayed if input data is invalid.
These 2 requirements will be tested by test case provided below.

Input validation check

1. test-input

   Control: Automatic

   Input: Any raw EEG signal which was sampled at 512 Hz.
   Output: The expected output for this data is an error message as the Welch power spectrum would be calculated and the power would be checked to see if it contains frequencies more than 100Hz and lower than 0.05 Hz

   Test Case Derivation: The frequency spectrum of the raw data depends on the sampling frequency and when it is sampled at 512 Hz, it contains over-range frequencies when it is not band-pass filtered yet. Thus it contains frequencies more than 100Hz and lower than 0.05 Hz.

   How test will be performed: Write a script to automatically feed the raw EEG data in the code and test it.

2. test-rank

   Control: Automatic

   Input: Matrix

$$\begin{bmatrix} 1 & 0 & 1 \\ -2 & -3 & 1 \\ 3 & 3 & 0 \end{bmatrix}$$

   would be fed as EEG signal

   Output: The expected output for this data is an error message

   Test Case Derivation: The matrix above has the rank 2 while the full rank is 3 so it is rank deficients and can be used as test case.

   How test will be performed: Write a script to automatically feed the rank deficiend raw EEG data in the code and test it.

5.1.2 Area of Output Testing

As described in the section 4.2.6 in SRS, EEGSourceLocalizer shall perform the correct calculation and plot the result (R4 and R5). In other words, this software should calculated the correct sources. These 2 requirements will be tested by test case provided below.

1. test-SimulatedEEG

   Control: Mixed manual and automatic

   Input: EEG data simulated from one and three dipoles in known locations (we set coordination of the source in the head model) with known activity and known added noise. (Please find the test case generator script in the software Github repo under src/test)

   Dipole locations, frequency and amplitude(strength):

$$\begin{bmatrix} -50 & -10 & 30 & 70\ \sin(10\pi \mathrm{t}\ ) \\ 40 & -10 & 20 & 100\ \cos(20\pi \mathrm{t}\ ) \\ -10 & 20 & -10 & 90\ \cos(30\pi \mathrm{t}\ ) \end{bmatrix}$$

   Output: The expected output is the EEGSourceLocalizer generates a matrix of source time series and plot of the brain activity map in the same locations as the locations we set for the simulating the EEG data.

   How test will be performed: A test script would simulate the EEG data for the location and amplitude of the current dipoles (sources) and then it will feed this data to the code and obtain the coordination of the most active sources based on the averaged power over time. Finally, the norm of the difference between the known coordination and software output will be calculated and reported. It will be judged against the resolution of the source map (the brain voxel size).
   Also, this script will plot the dipoles from which we simulate the EEG data and also we plot the output of the software for the simulated EEG data and it can be manually judged by eyes.

2. test-Brainstorm-PseudoOracle

Control: Automatic

Input: EEG data for BCI IV competition dataset dat to software and to the Brainstorm software as pseudo-oracle.

Output: The expected output is the EEGSourceLocalizer generates a matrix of source time series and a plot of the brain activity map in the same locations as the Brainstorm software would.

How test will be performed: We will manually obtain the source time series from the Brainstorm software and then, a test script would compare the norm of the difference between this software's output and the Brainstorm software output. Finally, a relative error would be calculated and reported.
Also, we will plot the source activity maps in Brainstorm and also we plot the output of the software for the BCI iv competition EEG data and the correctness can be manually judged by eyes.

5.2 Tests for Nonfunctional Requirements

The EEGSourceLocalizer software has a nonfunctional requirements regarding accuracy (NFR1). Generally, due to the nature of the problem, it is very hard to obtain a measure of accuracy or error for an arbitrary EEG signal. Based on the developer's knowledge to this date, the only dataset which the exact location of the source is know is the ground truth dataset (mentioned earlier). Thus, the accuracy for the computed solution should meet the level of accuracy achieved by other commercial softwares on the ground truth dataset publish in Mikulan et al. (2020). Also, the test introduced in section 5.1.2, test-Brainstorm-PseudoOracle would be a good measure of comparison.

5.3 Traceability Between Test Cases and Requirements

References

URL http://www.bbci.de/competition/iv/desc_1.html.

Ezequiel Mikulan, Simone Russo, Sara Parmigiani, Simone Sarasso, Flavia Maria Zauli, Annalisa Rubino, Pietro Avanzini, Anna Cattani, Alberto Sorrentino, Steve Gibbs, Francesco Cardinale, Ivana Sartori,

	R1	R2	R3	R4	R5	NR1
test-input	X					
test-rank		X				
test-SimulatedEEG			X	X		
test-BrainstormPseudoOracle				X	X	X
test-GroundTruthData				X	X	X

Table 1: Traceability Matrix Showing the Connections Between Tests and Functional and Nonfunctional System Requirements

Lino Nobili, Marcello Massimini, and Andrea Pigorini. Simultaneous human intracerebral stimulation and HD-EEG, ground-truth for source localization methods. *Scientific Data*, 7(1):127, 2020. ISSN 2052-4463. doi: 10.1038/s41597-020-0467-x. URL https://doi.org/10.1038/s41597-020-0467-x.