Module Interface Specification for EEGSourceLocalization

Leila Mousapour

November 30, 2020

1 Revision History

Date	Version	Notes
27 Nov, 2020	1.0	First draft of the MIS

2 Symbols, Abbreviations and Acronyms

See SRS Documentation at https://github.com/LeilaMousapour/Brain-Computer-Interface-/blob/master/docs/SRS/SRS.pdf

Contents

3 Introduction

The following document details the Module Interface Specifications for EEGSourceLocalization. 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 how every module is interfacing with every other part of the program based on "module state machine" approach.

Complementary documents include the System Requirement Specifications and Module Guide. The full documentation and implementation can be found at https://github.com/LeilaMousapour/Brain-Computer-Interface-

4 Notation

The structure of the MIS for modules comes from Hoffman and Strooper (1995), with the addition that template modules have been adapted from Ghezzi et al. (2003). The mathematical notation comes from Chapter 3 of Hoffman and Strooper (1995). For instance, the symbol := is used for a multiple assignment statement and conditional rules follow the form $(c_1 \Rightarrow r_1 | c_2 \Rightarrow r_2 | ... | c_n \Rightarrow r_n)$.

The following table summarizes the primitive data types used by EEGSourceLocalization.

Data Type	**Notation**	**Description**
character	char	a single symbol or digit
integer	$\mathbb{Z}$	a number without a fractional component in (-∞, ∞)
natural number	$\mathbb{N}$	a number without a fractional component in [1, ∞)
real	$\mathbb{R}$	any number in (-∞, ∞)
matrix	mat	2D datatype containing double-precision floating-point values.
cell	cell	A cell array is a data type with indexed data containers called cells, where each cell can contain any type of data.
structure	structure	A structure array is a data type that groups related data using data containers called fields. Each field can contain any type of data.Access data in a structure using dot notation of the form structName.fieldName.

The specification of EEGSourceLocalization uses some derived data types: matrix which is datatype containing double-precision floating-point values, cell array that is a data type with indexed data containers called cells, where each cell can contain any type of data and structure array is a data type that groups related data using data containers called fields. EEGSourceLocalization uses functions, which are defined by the data types of their inputs and outputs. Local functions are described by giving their type signature followed by their specification.

5 Module Decomposition

The following table is taken directly from the Module Guide document for this project.

Level 1	**Level 2**
Hardware-Hiding	
Behaviour-Hiding	Input Parameters Output Format Covariance Calculator Head Model Lead Filed Source Localization Control Module Specification Parameters Module
Software Decision	Matrix/Cell/Structure built-in MATLAB Data Structures Various built-in MATLAB and Fieldtrip toolbox algorithms Plotting

Table 1: Module Hierarchy

6 MIS of Control Module

6.1 Module

main

6.2 Uses

- Source Localization Module (Section 12)
- Output Format Module (Section 14)
- Plot Module (Section 13)

6.3 Syntax

6.3.1 Exported Constants

None

6.3.2 Exported Access Programs

Name	In	Out	Exceptions
main	-	-	-

6.4 Semantics

6.4.1 State Variables

None

6.4.2 Access Routine Semantics

main():

- transition: Modify the state of Param module and the environment variables for the Plot and Output modules by following these steps

Get (filenameIn: string) and (filenameOut: string) from user

load_params(filenameIn)

#Find covariance of the input EEG data (EEG_cov
$EEG_cov :=$ solve(CovCalc(Param.EEG))

Generate a head model based of the MRI

$Headmodel := HeadModelgenerator(\text{MRI})$

#*Align the electrodes on the generated head model and calculate the forward model (lead field matrix)*

$LeadField := \text{solve}((leadfield(\text{headmodel,elecLoc}))$

#*Calculate the sources.*

LCMVbeamformer($headmodel$, $EEGcov$, $leadfiled$, $elecAligned$, $AlgCfg$)

$out := Var(\mathbf{q}_i) = tr[\mathbf{H}^T(q_i)C^{-1}(x)H(q_i)]^{-1}$

#*Output calculated source activities to a file and to a plot.*

plot(MRI, $Var(\mathbf{q}_i)$)

output(filenameOut, $Var(\mathbf{q}_i)$, EEG, $elecAligned$, $AlgCfg$)

7 MIS of Input Parameter Module

7.1 Module

Param

7.2 Uses

- Specification Parameter Module (Section 8)
- Hardware hiding module

7.3 Syntax

7.3.1 Exported Constants

None

7.3.2 Exported Access Programs

Name	In	Out	Exceptions
paramLoad	String (fileName.ext)	-	FileError
VrifyParam	-	-	FrequencyRangeError, EEGRankDeficientError
EEG	-	structure	
elec-location	-	structure	
MRI	-	structure	
AlgConfig	-	structure	

7.4 Semantics

7.4.1 State Variables

The input parameter module will be implemented as a class and it will remember the values of the input parameters for the lifelong of the program. Thus, all the inputs are state variables.

From R1
EEG: structure
elec-location: structure
MRI: structure
AlgConfig: structure

7.4.2 Environment Variables

Its purpose is to capture when the module has external interaction with the environment, such as for a device driver, screen interface, keyboard, file, etc. In the case of this module, the environment vars are the files we read the inputs from.

EEGdataFile : a file contacting the filtered, cleaned EEG data which is segmented and formatted into Fieldtrip package acceptable format (containing the trials, label of the channels, time points and sampling frequency.)

ElectrodeLocationFile: a file containing the location of the EEG electrode channels, the system the EEG was recorded (Biosemi, 10-20 etc.) and the coordination system of the values.

MRIfile: a file containing the anatomycal MRI of the person or a template MRI file.

AlgConfig: a file containing the type of the SL algorithm to use and the configurations of the algorithm.

7.4.3 Assumptions

- ParamLoad will be called before the values of any state variables will be accessed.
- The EEGdataFile is arranged as Fieldtrip package acceptable format.
- ElectrodeLocationFile is in .loc or .sfp format.

7.4.4 Access Routine Semantics

Function to load, verify, and store input data (R1 and R2 from SRS).

1. Dot function to access the state variables

   Param.EEG:

   - output: $out := EGG$
   - exception: none

   Param.$elec - loc$:

   - output: $out := elec - loc$
   - exception: none

   Param.MRI:

- output: $out := MRI$
- exception: none

Param.$AlgCfg$:

- output: $out := AlgCfg$
- exception: none

2. load_params($EEGdataFile, ElectrodeLocationFile, MRIfile, AlgConfig$):

   - transition: The filenames are first associated with the file f. inputFile is used to modify the state variables using the following procedural specification:
     (a) Read data from inputFiles to populate all the state variables.
     (b) Calculate the Welch power spectrum of the EEG signal.
     Param.$EEGPSD$:= EEg Welch power spectral density.
     (c) Calculate the rank of the EEG signal matrix.
     Param.$EEGrank$:= rank of the EEG data matrix
     (d) verify_params()
   - exception: exc := a file name s cannot be found OR the format of inputFile is incorrect ⇒ FileError

3. verify_params():

   - out: out := none
   - exception: exc :=

$\neg(Param.\text{EEGPSD} < 0.01)$	⇒ EEGnotFiltered
$\neg(Param.\text{EEGPSD} > 100)$	⇒ EEGnotFiltered
$\neg(Param.\text{EEG} < -50)$	⇒ EEGAmpError
$\neg(Param.\text{EEG} > 50)$	⇒ EEGnoEEGAmpErrortFiltered
$\neg(Param.\text{EEGrank} < Param.\text{EEG}.channelNum)$	⇒ RankDeficient

8 MIS of Parameter Specification Module

The secrets of this module is the value of the specification parameters.

8.1 Module

SpecParam

8.2 Uses

N/A

8.3 Syntax

8.3.1 Exported Constants

From Table 2 in SRS update the values in the data object:
SpecParam.EEG_min := -50
SpecParam.EEG_max := +50
SpecParam.$F(EEG_min)$:= 0.05
SpecParam.$F(EEG_max)$:= 100

8.3.2 Exported Access Programs

Name	In	Out	Exceptions
SpecParam	-	-	-

8.4 Semantics

N/A

9 MIS of Covariance Calculation

This module calculates the covariance of the EEG data to use in the SL module later.

9.1 Module

CovCalc

9.2 Uses

- Input Parameter Module (Section 7)

9.3 Syntax

9.3.1 Exported Constants

None

9.3.2 Exported Access Programs

Name	**In**	**Out**	**Exceptions**
CovCalc	structure	structure	-

9.4 Semantics

9.4.1 Access Routine Semantics

CovCalc(Param.EEG):

- output: $out := \frac{\sum_{n=1}^{channno}(x_i - \bar{x})(y_i - \bar{y})}{channo - 1}$
- exception: none

10 MIS of Head model Module

The secrete of this module is that is creates the head model with BEM (Boundary Element Method) based on MRI and align the electrodes on it. The likely change here is that method of creating the head model and the parameters (conductivity of the brain tissues) might change.

10.1 Module

HeadModelgenerator

10.2 Uses

- Input Parameter Module (Section 7)

10.3 Syntax

10.3.1 Exported Constants

None

10.3.2 Exported Access Programs

Name	**In**	**Out**	**Exceptions**
getFiducial	matrix	matrix	-
volumeSegment	structure	structure	-
prepareMesh	structure	structure	-

10.4 Semantics

10.4.1 Assumptions

- we assume the tissue conductivity for each tissue type to be a certain constant

10.4.2 Access Routine Semantics

1. getFiducial(MRI):
   - output: $out := \begin{bmatrix} x_1 & y_1 & z_1 \\ x_2 & y_2 & z_2 \\ x_3 & y_3 & z_3 \end{bmatrix} = \begin{bmatrix} Nasion \\ RightEar \\ LeftEar \end{bmatrix}$
   - exception:
     exc := MRI structure does not include fiducial point's coordinations $\Rightarrow$ FidutialMissing

2. volumeSegment(MRI)

    - output: $out := volSeg =$ the coordination of all points of the 3 surfaces of the head (brain/skull/scalp)
    - exception: none

3. prepareMesh($volSeg$)

    - output: $out :=$ the segmentation of the 3 surfaces of the head (brain/skull/scalp) using Boundary Element Method.
    - exception: none

11 MIS of Lead field Module

The service of this module is that it first align the electrode locations on the generated headmodel and then, segments the brain volume into 3D grid of voxels and create the source model, solve the forward problem and creates the leadfield matrix.

11.1 Module

LeadFeild

11.2 Uses

- Input Parameter Module (Section 7)

11.3 Syntax

11.3.1 Exported Constants

None

11.3.2 Exported Access Programs

Name	In	Out	Exceptions
alignElec	structure	structure	-
prepSourceModel	structure	structure	-
LFcompute	structure	structure	-

11.4 Semantics

11.4.1 Access Routine Semantics

1. alignElec($elecLoc$):

   - output: is a structure including the new coordination for each electrode which is the result of a n linear transfer that maps the fiducial points of the input electrode file with the input MRI file that we created the headmodel based on.
   - exception: none

2. prepSourceModel($alignedElecLoc$, $headmodel$):

   - output: is a structure that contains the segmented headmodel volume into voxels which each of them will be considered as a source point. Thus, it creates out source model (3D location of the centre of each voxel).
   - exception: none

3. LFcompute(*alignedElecLoc*, *sourceModel*):

   - output: $out := K = \begin{bmatrix} K_{1,1} & K_{1,2} & \dots & K_{1,N_v} \\ \vdots & \ddots & & \\ K_{N_e,1} & \dots & & K_{N_e,N_v} \end{bmatrix}$ Where $K_{i,l} = (k^x_{i,l}, k^y_{i,l}, k^z_{i,l})$ is the scalp electric potential at the ith electrode, due to a unit strength X-oriented dipole at the lth voxel/source

   - exception: none

12 MIS of Source Localization Module

Compute the sources' activity in time based on the type of algorithm the user chose

12.1 Module

FindSources

12.2 Uses

12.3 Uses

- Covariance Calculator Module (Section 9)
- Head Model Module (Section 10)
- Lead Feild Module (Section 11)

12.4 Syntax

12.4.1 Exported Constants

None

12.4.2 Exported Access Programs

Name	In	Out	Exceptions
LCMVbeamformer	structure	structure	-
sLORETA	structure	structure	-

12.5 Semantics

12.5.1 Assumptions

- For simplicity of the calculation, it is assumed the orientation of the dipole (current source) is perpendicular to the scalp as opposed to calculation of the activity of the source in the 3 orientations x, y, x in the space.

12.5.2 Access Routine Semantics

1. LCMVbeamformer($headmodel$, $EEGcov$, $leadfiled$, $elecAligned$):
   - output: $out := Var(\mathbf{q}_i) = tr[\mathbf{H}^T(q_i)C^{-1}(x)H(q_i)]^{-1}$
   - exception: non
2. sLORETA($headmodel$, $EEGcov$, $leadfiled$, $elecAligned$):

- output: $out := Var(\mathbf{q}_i) = tr[\mathbf{H}^T(q_i)C^{-1}(x)H(q_i)]^{-1}$
- exception: non

13 MIS of Plotting Module

13.1 Module

plot

13.2 Uses

N/A

13.3 Syntax

13.3.1 Exported Access Programs

Name	In	Out	Exceptions
plot	structure	-	-

13.4 Semantics

13.4.1 State Variables

None

13.4.2 Environment Variables

win: 2D sequence of pixels displayed on the screen

13.4.3 Assumptions

None

13.4.4 Access Routine Semantics

plot(MRI, $Var(\mathbf{q}_i)$):

- transition: Modify win to display a plot where the vertical axis is time and one horizontal axis is temperature and the other horizontal axis is energy. The time should run from 0 to t_{final}
- exception: none

14 MIS of Output Module

14.1 Module

output

14.2 Uses

- Input Parameter Module (Section 7)

14.3 Syntax

14.3.1 Exported Constants

max_width: integer

14.3.2 Exported Access Program

Name	In	Out	Exceptions
output	fname: string, structure	-	-

14.4 Semantics

14.4.1 State Variables

None

14.4.2 Environment Variables

file: A MATLAB .mat file

14.4.3 Access Routine Semantics

output(fname, $Var(\mathbf{q}_i)$, EEG, $elecLoc$, $AlgCfg$):

- transition: Write to environment variable named fname the following: the input parameters from Param, and the calculated source activities $Var(\mathbf{q}_i)$. The functions will be output as sequences in this file. Also, the EEG data, electrode locations and the configuration for the SL algorithm from the input module are save with the output of the source localization so that it is recorded that how the sources are obtained.

- exception: none

References

Carlo Ghezzi, Mehdi Jazayeri, and Dino Mandrioli. *Fundamentals of Software Engineering*. Prentice Hall, Upper Saddle River, NJ, USA, 2nd edition, 2003.

Daniel M. Hoffman and Paul A. Strooper. *Software Design, Automated Testing, and Maintenance: A Practical Approach*. International Thomson Computer Press, New York, NY, USA, 1995. URL `http://citeseer.ist.psu.edu/428727.html`.

[Extra information if required —SS]