Detailed examples of concrete BCI applications with OpenViBE

Sébastien Rimbert
Brain-computer interfaces (BCIs) are communication systems that enable to send commands to computer by means of brain signals alone. These brain signals are usually measured using electroencephalography (EEG), and then processed by the BCI.

[adapted of Lotte., 2012]

[Wolpaw & Wolpaw., 2012]
Summary

• OpenViBE **to control** a robotic device

• OpenViBE for **stroke rehabilitation**

• OpenViBE to better monitoring **general anesthesia**
BCI based on cerebral motor activity
Robotic arm control based on motor imageries combination

[Lindig-Leon et al, IEEE EMBC, 2017]
Motor imageries combination

How to increase the number of mental orders using combined motor imageries?

[Lindig-Leon et al, IEEE UKRCON 2017]
OpenViBE scenario – JacoArm

Pre-processing

EEG signal acquisition

Classification algorithm

Feedback

Cues display

Visual feedback

[ Lindig-Leon et al, IEEE UKCRON 2017 ]
Robot control based on SSVEP-BCI

by just focusing on the desired SSVEP target.

[Si-Mohammed et al, IEEE TVCG 2018, under publication]
Robot control based on SSVEP-BCI

[Si-Mohammed et al, IEEE TVCG 2018, under publication]
OpenViBE for stroke rehabilitation

GRASP-IT with your Brain
Cerebrovascular accident (CVA)
Stroke rehabilitation

Kinesitherapy, Mirror therapy, Ergotherapy, Psychomotricity, Brain-computer Interface

Motor cortex stimulation
Increased synaptic plasticity

[Pichiorri et al., 2015]
OpenViBE for stroke rehabilitation

[Rimbert et al., 2017, GRASP-IT project]
OpenViBE scenario – Grasp’it

1st
First Scenario
Data Acquisition

2nd
Second Scenario
Classifier Trainer

3rd
Third Scenario
EEG Online Classification

EEG

Feedback

Unreal
VRPN
Python
OpenViBE scenario 1 – Data acquisition

EEG signal acquisition

Pre-processing (remove channel + filtering)

File recording

Cues display

CSP training

Acquisition client

EEG signal acquisition

Channel Selector

Temporal filter

CHECK settings of the temporal filter

LUA Simulostimulator

CHECK settings for the LUA protocol stimulator.

Identity

Signal display

Pre-processing

Remove channel + filtering

GDF file writer

Stimulation based epoching

Rest class

Stimulation based epoching

Right hand movement trials

CSP Spatial Filter Trainer

CHECK the settings of the box.

OVTK_StimulationId_Train is received at the end of the LUA stimulator protocol
OpenViBE scenario 2 – Classifier trainer

Pre-processing (remove channel + filtering)

File recorded during the 1st scenario

CSP file created during the 1st scenario
OpenViBE scenario 2 – Classifier trainer

- Epoching of 2-classes
- Features selection for 2-classes
- Training classifier
OpenViBE scenario 3 – Online classification

EEG signal acquisition
- Generic stream reader has to be modified in order to capture EEG.
- Channel Selector
  - EX 1; EX 2; EX 3; EX 4; EX 5; EX 6; EX 7; EX 8
  - Can assign a signal display box on EX1
- Temporal filter
  - CHECK settings of the temporal filter

Pre-processing
- Lua Stimulator
- Protocole Stimulator
- CHECK

File recording
- Generic stream writer
- GDF file writer
- Player Controller
- Time based epoching
  - The number of samples is indirectly set up here.

CHECK if the config file needed is the good one.
OpenViBE scenario 3- Online classification

Communication with Unreal Engine

Classification (CSP + LDA)
OpenViBE scenario – Unreal Engine
Visual + Haptic Neurofeedback

[ANR GRASP-IT, Bougrain, 2018]
Visual + Haptic feedback
Visual + Haptic feedback

[ANR GRASP-IT, Bougrain (submitted)]
Visual + Tactile feedback

[Jeunet et al., Interact 2015]
https://camillejeunet.wordpress.com/mini-tutorial-tactile-feedback/
“General anesthesia is a reversible medically process induced unconsciousness, amnesia, analgesia and immobility.”
Clinical protocol

3 motor task:
- Rel movement;
- Motor imagery;
- Median nerve stimulation
Experimental conditions
Interfacing

Stimulateur ENERGY

Pulse Analogique

Arduino

UART

Python

Trigger

OpenViBE

Script LUA

OpenViBE

Acquisition EEG

Sujet

Median nerve Micromed stimulator

EEG Biosemi amplificator
OpenViBE scenario – Median Nerve Stimulation

Change the study’s parameter in the Lua Stimulator (double click on it):
Time interval between two trials, the time between the first auditory cue and the MNS, ...

Lua Stimulator

Keyboard stimulator
Create a trigger when we press the pointer’s button

Player Controller

Stimulation multiplexer

Signal display

GDF file writer

Stimulation Listener

First auditory cue
Sound Player

Second auditory cue
Sound Player

Python script used to communicate with the Arduino

Sound cues + median nerve stimulation

---

for i = 1, number_of_trials do
  var = math.random(-intervariab, intervariab) -- Récupération d'une nouvelle valeur aléatoire à chaque boucle
  var_s = var/1000 -- Conversion de ms en s
  t = t + intervalle + var_s -- Calcul du prochain instant de stimulation
  t_stim = t + stim_time --Calcul de l'instont de départ de la cue
  io.write(string.format("Prochaine stim a %f \n", t_stim))
  io.write(string.format("Prochaine cue a %f \n", t_cue))
  box:send_stimulation(1, OVTK_StimulationId_Label_00, t_stim)
  box:send_stimulation(2, OVTK_StimulationId_Beep, t)
  box:send_stimulation(3, OVTK_StimulationId_Beep, t+2) --Deuxième bip sonore
end

---

t = t + 20 --20 seconds after the last stimulation, used to stop the scenario
box:send_stimulation(4, OVTK_StimulationId_TrialStop, t)
Median Nerve Stimulation – Python script

```python
import serial
import serial.tools.list_ports

arduino_ports = [
    p[0]
    for p in serial.tools.list_ports.comports()
    if 'Arduino' in p[1]
]
if not arduino_ports:
    raise IOError("No Arduino found")
if len(arduino_ports) > 1:
    print ('Multiple Arduinos found')
ser = serial.Serial(arduino_ports[0])
print ("Arduino found, connected to port {}".format(p[0]))

class MyOVBox(OVBox):
    def __init__(self):
        OVBox.__init__(self)  # Initialisation de la box OpenViBE
        self.signalHeader = None

    def process(self):
        for chunkIndex in range( len(self.input[0]) ):
            if(type(self.input[0][chunkIndex]) == OVStimulationSet):
                stim = self.input[0].pop()  # Create a variable containing the stimulation
                if stim:
                    ser.write('z')  # Write on the serial port in case we have any stimulation

    def uninitialized(self):
        ser.close()  # Close the serial port when stopping the OpenViBE scenario
        return

box = MyOVBox()
```

**OpenViBE box initialization**

**MNS stimulation**
OpenViBE scenario – Median Nerve Stimulation

Change the study's parameter in the Lua Stimulator (double click on it):
Time interval between two trials, the time between the first auditory cue and the MNS, ...

Lua Stimulator

Keyboard stimulator
Create a trigger when we press the pointer’s button

Player Controller

Stimulation multiplexer

Acquisition client

Signal display

GDF file writer

Stimulation listener

Trigger movements recordings

Python script used to communicate with the Arduino

First auditory cue

Sound Player

Second auditory cue

Sound Player

Sound Player

for i = 1, number_of_trials do
var = math.random(-intervariable, intervriabl) -- Récupération d’une nouvelle valeur aléatoire à chaque boucle
var_s = var/1000 -- Conversion de ms en s
t = t + intervalle + var_s -- Calcul du prochain instant de stimulation
t_stim = t + stim_time -- Calcul de l’instant de départ de la cue
io.write(string.format("Prochaine stim à %i \n", t_stim))
--io.write(string.format("Prochaine cue à %f \n", t Cue))
box:send_stimulation(1, OVTK_StimulationId_Label_00, t_stim) -- Programmation de la stimulation au temps t précédemment calculé
box:send_stimulation(2, OVTK_StimulationId_Beep, t)
box:send_stimulation(3, OVTK_StimulationId_Beep, t+2) -- Deuxième bip sonore
end

t = t + 20 -- 20 seconds after the last stimulation, used to stop the scenario
box:send_stimulation(4, OVTK_StimulationId_TrialStop, t)
Thank you for your attention

ありがとうございます
Detailed examples of concrete BCI applications with OpenViBE

Sébastien Rimbert