

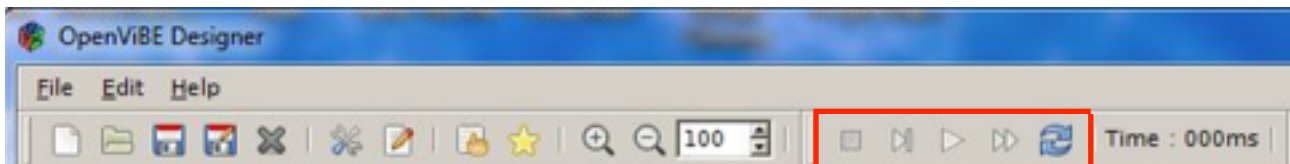
Guide d'utilisation d'OpenVIBE

I) Généralités

OpenVIBE est découpé en 2 parties : le designer et le serveur d'acquisition :

- Le designer permet de réaliser des interfaces cerveau-ordinateur via un système de schéma-bloc appelé scénario. Chaque scénario correspond à une chaîne de traitement à effectuer, et est constitué de différents blocs reliés les uns les autres.
- Le serveur d'acquisition permet de récupérer les données EEG issues d'un casque neuronal (comme le casque EPOC d'EMOTIV). Celles-ci peuvent alors être récupérées grâce au bloc « acquisition client » du designer.

II) Description de l'interface du designer



Un scénario fonctionne un peu comme un flux vidéo. Il existe une palette d'outils pour les contrôler (encadrée en rouge dans la capture ci-dessus). Ainsi, il est possible de lancer un scénario en appuyant sur le bouton représentant un triangle, de le stopper en appuyant sur le bouton représentant un carré, etc. Il est possible d'en lancer plusieurs en même temps. Il est également possible d'enregistrer un scénario dans le format xml.

III) Présentation des différents blocs utiles

OpenVIBE utilise un système de blocs pour parvenir à faire des interfaces cerveau ordinateur. Voici la liste des différents blocs utiles, classés par catégorie (source : http://openvibe.inria.fr/documentation/unstable/Doc_BoxAlgorithms.html) :

Acquisition et Réseau

- Acquisition Client : Ouvre une socket pour lire les informations envoyées par le serveur d'acquisition (i.e. les données du casque en temps réel). Il faut fournir le port utilisé par le serveur (1024 par défaut).
- Analog VRPN Server : Permet d'envoyer des données à une application client via le protocole VRPN.

Classification

- Classifier Processor : Permet de classer les vecteurs de caractéristiques entrants en utilisant un classifieur entraîné précédemment.
- Classifier Trainer : Effectue l'entraînement du classifieur selon l'algorithme utilisé (LDA, ...).

Extraction de caractéristiques

- Feature aggregator : Permet de rassembler les différentes entrées dans un vecteur caractéristique.

Lecture et écriture de fichier

- Generic Stream Writer : Permet d'écrire des données dans un fichier binaire.
- Generic Stream Reader : Permet de lire un fichier enregistré grâce à un bloc Generic File Writer.

Traitement de signal

- Temporal Filter : Applique un filtre temporel sur un signal. On peut choisir la méthode de filtrage, le type et l'ordre de filtre et les différentes fréquences de coupure.
- Spatial Filter : Permet de rediriger M entrées vers N sorties en multipliant chaque vecteur d'entrée avec une matrice.
- CSP Spatial Filter Trainer : Permet de calculer les coefficients d'un filtre spatial selon l'algorithme CSP.
- Identity : Duplique l'entrée de ce bloc vers sa sortie.
- Time Based Epoching : Permet de découper un signal en plusieurs morceaux ayant une durée et un intervalle spécifié.
- Simple DSP : Applique des formules mathématiques à une matrice.
- Signal Average : Permet de calculer la moyenne de chaque échantillon en entrée et délivre le résultat en sortie.
- Spectral Analysis (FFT) : Calcule l'analyse spectrale en utilisant la FFT.
- Spectrum Average : Calcule la moyenne de toutes les puissances de bande de fréquence pour un spectre.

Stimulation

- Player Controller : Permet de contrôler l'arrêt et la mise en pause d'un scénario à l'aide des stimulations reçues.

Visualisation

- Signal Display : Permet d'afficher un signal en temps réel. On peut par exemple utiliser plusieurs blocs de ce type par scénario pour comparer un signal après différents filtrages.
- Power Spectrum Display : Permet d'afficher l'amplitude d'un signal dans un ensemble de bandes de fréquences.
- 2D Topographic Map : Permet d'afficher l'activité cérébrale en 2 dimensions.
- 3D Topographic Map : Permet d'afficher l'activité cérébrale en 3 dimensions.