

Localisation d'activité intracérébrale à partir d'électrodes de surface

L1 MIEE option électronique - Séance 2/3

Laboratoire LTSI - UMR INSERM 642 - Université de Rennes 1

1 Introduction

Au cours du premier TP, nous avons défini le cadre général de l'étude : à partir d'enregistrements ElectroEncéphaloGraphiques (EEG) de surface, nous cherchons à localiser une zone épileptogène à la surface du cerveau. Nous avons également découvert le logiciel MATLAB, qui permet entre autres de manipuler facilement des vecteurs. Durant cette seconde séance, nous allons continuer à utiliser MATLAB et irons un peu plus loin dans son utilisation avec l'écriture d'un programme principal et de fonctions. Nous continuerons également à manipuler des vecteurs en étudiant la notion de produit scalaire normalisé, outil central de la méthode de localisation étudiée en cours.

2 Création et utilisation d'une fonction MATLAB

Le logiciel MATLAB permet, outre l'exécution directe de commandes dans la fenêtre de commande de MATLAB, d'écrire des programmes qui pourront être exécutés ultérieurement.

- Cliquer sur l'onglet "M-File" dans le menu "File/New/" de MATLAB. L'éditeur de programme de MATLAB va alors s'ouvrir.

Nous allons créer une fonction permettant de calculer la transposée Hermitienne d'un vecteur.

- Dans la fenêtre qui s'est ouverte, tapez les 2 lignes suivantes :
function [v1 v2] = TransposeeHermitienne(vecteur) % declaration de la fonction
% corps de la fonction
v1 = conj(vecteur. '); % codage de la fonction
v2 = vecteur';

Nous allons maintenant enregistrer cette fonction dans notre répertoire de travail. Il est important que le nom du fichier soit le même que le nom de la fonction associée. Vous devez donc appeler votre fichier "TransposeeHermitienne" (le format de fichier *.m est le format MATLAB).

- Cliquer sur l'onglet "Save As" dans le menu "File/" de l'éditeur de programme de MATLAB afin de sauvegarder le programme sous le nom de votre choix dans le répertoire de votre choix. N'hésitez pas à vous créer un répertoire de travail propre à ce module d'enseignement sur votre compte.

Maintenant, vous pouvez revenir dans le logiciel et vous placer dans votre répertoire de travail.

- Servez vous de la barre d'adresse "current directory" présente en haut de la fenêtre de commande de MATLAB.

Vous pouvez maintenant créer un programme principal qui aura pour but d'appeler la fonction précédemment créée.

- Ouvrez une nouvelle page d'écriture à l'aide de l'éditeur de MATLAB que vous sauvegarderez dans votre répertoire de travail sous le nom "tp2".

La fonction "TransposeeHermitienne" prend en argument un vecteur et renvoie la transposée Hermitienne de ce vecteur. Il vous faut donc commencer par créer un vecteur $\mathbf{v} = [2i, -1, 1]^T$:

- Compléter le programme "tp2" afin de tester la fonction "TransposeeHermitienne". Sauvegardez-le à nouveau, puis exécutez-le dans la fenêtre de commande MATLAB. Vérifiez que le résultat obtenu par MATLAB est bien la transposée hermitienne du vecteur \mathbf{v} défini ci-dessus.

3 Le produit scalaire normalisé

3.1 Définitions et propriétés

L'algorithme étudié en cours permet de localiser une source épileptique de profondeur à partir de données EEG de scalp. Le but final de ce TP est d'implémenter cet algorithme afin de localiser la source que vous aurez vous même positionnée sur le cortex. Nous reviendrons plus en détails sur cet algorithme lors de la séance prochaine. Ce que nous pouvons d'ores et déjà dire, c'est que la méthode de localisation repose sur l'optimisation de la fonction de coût suivante :

$$\varphi(\mathbf{a}(\boldsymbol{\theta}), \mathbf{e}) = \frac{\|\langle \mathbf{e}, \mathbf{a}(\boldsymbol{\theta}) \rangle\|^2}{\|\mathbf{e}\| \|\mathbf{a}(\boldsymbol{\theta})\|} = \frac{\|\mathbf{e}^H \mathbf{a}(\boldsymbol{\theta})\|^2}{\|\mathbf{e}\| \|\mathbf{a}(\boldsymbol{\theta})\|} = \frac{\mathbf{a}(\boldsymbol{\theta})^H \mathbf{e} \mathbf{e}^H \mathbf{a}(\boldsymbol{\theta})}{\|\mathbf{e}\| \|\mathbf{a}(\boldsymbol{\theta})\|} \quad (1)$$

par rapport à $\boldsymbol{\theta}$, où \mathbf{e} est un vecteur que nous définirons lors de la prochaine séance de TP. Rappelons que le vecteur $\mathbf{a}(\boldsymbol{\theta})$ désigne le vecteur de transfert vers les électrodes EEG de l'activité électrique du dipôle de courant situé à la position $\boldsymbol{\theta}$. La fonction Ψ (1) est connue sous le nom de produit scalaire normalisé des vecteurs $\mathbf{a}(\boldsymbol{\theta})$ et \mathbf{e} . Nous allons à présent étudier les propriétés d'une telle fonction et son implémentation sous MATLAB.

Soient $\mathbf{a} = [a_1, a_2]^T$ et $\mathbf{b} = [b_1, b_2]^T$ deux vecteurs de \mathbb{C}^2 . Le produit scalaire de ces deux vecteurs est défini par :

$$\langle \mathbf{a}, \mathbf{b} \rangle = a_1 b_1^* + a_2 b_2^* \quad (2)$$

ou encore :

$$\langle \mathbf{a}, \mathbf{b} \rangle = \|\mathbf{a}\| \|\mathbf{b}\| \cos(\widehat{\mathbf{a}, \mathbf{b}}) \leq \|\mathbf{a}\| \|\mathbf{b}\| \quad (3)$$

où la norme d'un vecteur est définie par :

$$\|\mathbf{b}\| = \sqrt{\langle \mathbf{b}, \mathbf{b} \rangle} \quad (4)$$

On en déduit alors facilement l'inégalité suivante :

$$|\langle \mathbf{a}, \mathbf{b} \rangle| \leq \|\mathbf{a}\| \|\mathbf{b}\| \quad (5)$$

l'inégalité étant atteinte lorsque $\cos(\widehat{\mathbf{a}, \mathbf{b}}) = 1$, c'est-à-dire lorsque les deux vecteurs sont colinéaires. L'égalité (2) se généralise au cas de vecteurs de \mathbb{C}^N de la manière suivante :

$$\langle \mathbf{a}, \mathbf{b} \rangle = a_1 b_1^* + \dots + a_N b_N^* = \sum_{n=1}^N a_n b_n^* = \mathbf{b}^H \mathbf{a} \quad (6)$$

L'inégalité (5) reste alors valable, connue sous le nom d'inégalité de Cauchy-Schwarz. L'égalité est à nouveau atteinte si et seulement si \mathbf{a} et \mathbf{b} sont colinéaires ($\Leftrightarrow \mathbf{a} = \alpha * \mathbf{b}$ où $\alpha \in \mathbb{C}$). Pour deux vecteurs \mathbf{a} et \mathbf{b} non nuls, l'inégalité de Cauchy-Schwarz (5) peut se réécrire sous la forme :

$$\frac{|\langle \mathbf{a}, \mathbf{b} \rangle|}{\|\mathbf{a}\| \|\mathbf{b}\|} \leq 1 \quad (7)$$

On appelle alors produit scalaire normalisé la fonction φ définie par :

$$\varphi(\mathbf{a}, \mathbf{b}) = \frac{|\langle \mathbf{a}, \mathbf{b} \rangle|}{\|\mathbf{a}\| \|\mathbf{b}\|} \quad (8)$$

3.2 Manipulations du produit scalaire

Nous allons maintenant vérifier l'inégalité de Cauchy-Schwarz sur des exemples simples.

- Créez 3 vecteurs $\mathbf{a} = [2, 5, 6, 8, 6, 7]^T$, $\mathbf{b} = [2, 8, 6, 7, 4, 6]^T$ et $\mathbf{c} = [2, 8, 6, 7, 4, 6]^T$ et calculez les produits scalaires normalisés de ces vecteurs. Dans un premier temps, vous utiliserez les fonctions "dot" et "norm". Que remarquez-vous? Dans un second temps, vous chercherez à ne pas utiliser ces deux fonctions : vous pourrez utiliser la fonction "TransposeeHermitienne" implémentée précédemment. Vous créerez une fonction "ProdScalNorm" prenant en entrée deux vecteurs colonnes et renvoyant en sortie deux scalaires, correspondant aux deux manières précédentes d'implémenter le produit scalaire normalisé des deux vecteurs passés en entrée.

A présent, nous allons appliquer le produit scalaire normalisé à des signaux épileptiques, téléchargeables sur la page web de Laurent Albera¹ par le biais du fichier "pointes.mat".

- Ecrivez une ligne de commande permettant de charger le fichier "pointes.mat" en mémoire.
- Exécutez votre programme principal. Exécutez la commande "whos" dans la fenêtre de commande de MATLAB. A quoi sert cette fonction? Quelle variable a été créée lors du chargement en mémoire précédent? Quelles sont les dimensions de cette nouvelle variable? Ecrivez une procédure d'affichage graphique des signaux stockés dans cette variable : on choisira de représenter un signal par figure en mettant l'ensemble des figures dans une même fenêtre graphique d'affichage.
- Quels signaux semblent visuellement être les plus semblables?

Nous allons à présent vérifier le résultat précédent à l'aide du produit scalaire normalisé.

- Appliquez la fonction "ProdScalNorm" aux signaux épileptiques étudiés. Que remarquez-vous?

¹<http://perso.univ-rennes1.fr/laurent.albera/>