TRAP

**Logiciel de traitement des signaux de parole**

**Version 7**

**GUIDE**

**DE**

**L’UTILISATEUR**

**Réalisé par**

**Christophe SAVARIAUX**

**13/10/2017**

**Département Parole et Cognition, GIPSA-lab**

Table des matiÈres 1

I. PrÉsentation du logiciel 5

Table des matiÈres

I.1 Principe gÉnÉral 5

I.2 Le menu principal 6

I.3 La barre d’outils 6

I.4 Description des variables globales 7

I.5 Quelques prÉcautions À prendre 11

II. Utilisation du logiciel 12

II.1 Lancement du logiciel 12

II.1.1 Initialisation des options 12

II.1.1.1 Options pour les données acoustiques 13

II.1.1.2 Options pour les données EMA 13

II.1.1.3 Options pour les données EPG 14

II.1.1.4 Options pour les données TACLE 15

II.1.1.5 Options pour le filtrage de Chebyshev 15

II.1.2 Ouvrir un fichier 15

II.1.2.1 Ouverture d’un fichier acoustique binaire 16

II.1.2.2 Ouverture d’un fichier articulatoire EMA 16

II.1.2.3 Ouverture d’un fichier de paramètres TACLE 16

II.1.2.4 Ouverture d’un fichier de mesures ASCII 16

II.1.2.5 Ouverture des fichiers EVA 17

II.1.2.6 Ouverture d’un fichier EGG 17

II 1.3 Enregistrer des données temporelles dans un fichier au format binaire ou ASCII 18

II.2 Configuration des fenÊtres d’affichage 18

II.2.1 Fenêtre principale 18

II.2.2 Fenêtre synchrone 19

II.3 Traitements sur les signaux acoustiques 19

II.3.1 Traitements s’affichant dans la fenêtre principale 19

II.3.1.1 Affichage du signal acoustique 19

II.3.1.2 Filtrage du signal acoustique 19

II.3.1.3 Evolution de l’intensité du signal (RMS) 20

II.3.1.4 Spectrogramme 20

II.3.1.5 Affichage des formants 20

II.3.1.6 Affichage de la fréquence fondamentale F0 21

II.3.1.7 Ecoute du signal 21

II.3.1.8 Exemple d’affichage de traitements effectués sur un fichier acoustique 22

II.3.2 Traitements s’affichant dans la fenêtre synchrone 22

II.3.2.1 Calcul de la FFT et/ou de la LPC 22

II.3.2.2 Calcul du cepstre 23

II.3.2.3 Exemple d’affichage de traitements spectraux à partir d’un fichier acoustique 24

II.4 Traitements sur les signaux EMA 24

II.4.1 Trajectoire des bobines 24

II.4.2 Vitesse 25

II.4.3 Vitesse tangentielle 25

II.4.4 Accélération 25

II.4.5 Détection des minima, maxima, passages par zéro (ppz) ou moyennes 26

II.4.6 Exemple de traitements sur un fichier EMA 27

II.4.7 Coupe sagittale / Fichier de contour du palais 28

II.4.7.1 Extraction du fichier de contour du palais 28

II.4.7.2 Coupe sagittale 29

II.4.8 Animation des trajectoires des bobines 30

II.4.9 Affichage des tilts des bobines 31

II.5 Traitements sur les signaux EPG 31

II.5.1 Sélection des électrodes d’une région 31

II.5.2 Sauvegarde de ces régions dans le fichier .usr 32

II.5.3 Calcul du taux de constriction dans une région 32

II.5.4 Chargement des fichiers de résultat de traitement sur données EPG 32

II.5.5 Repérage des minima, maxima, passages par zéro ou moyennes 33

II.5.6 Exemple de traitements EPG 33

II.5.7 Visualisation des électrodes contactées du palais à un instant donné 34

II.5.8 Visualisation des électrodes contactées du palais sur plusieurs échantillons 34

II.6 Traitements sur les signaux TACLE 35

II.6.1 Trajectoire 35

II.6.2 Vitesse 35

II.6.3 Accélération 36

II.6.4 Filtrage 36

II.6.5 Repérage des minima, maxima, passages par zéro ou moyennes 36

II.6.6 Exemple de traitements sur un fichier TACLE 36

II.7 Traitements sur les signaux de mesures 37

II.7.1 Trajectoire 37

II.7.2 Vitesse 38

II.7.3 Accélération 38

II.7.4 Repérage des minima, maxima, passages par zéro ou moyennes 38

II.8 Traitements sur les signaux EVA 38

II.8.1 Trajectoire 38

II.8.2 Vitesse 39

II.8.3 Accélération 39

II.8.4 Filtrage 39

II.8.5 Repérage des minima, maxima, passages par zéro ou moyennes 39

II.8.6 Exemple de traitements sur des fichiers EVA 39

II.9 Traitements sur les signaux EGG 41

II.9.1 Trajectoire 41

II.9.2 Vitesse 41

II.9.3 Accélération 41

II.9.4 Filtrage 41

II.9.5 Repérage des minima, maxima, passages par zéro ou moyennes 42

II.9.6 Exemple de traitements sur des fichiers EGG 42

II.10 Outils divers 43

II.10.1 Etiquetage des signaux 43

II.10.1.1 Etiqueter les signaux 43

II.10.1.2 Sauvegarder les étiquettes 44

II.10.1.3 Charger un fichier d’étiquettes 45

II.10.1.4 Effacer une étiquette 45

II.10.1.5 Effacer un repère min-max-ppz 45

II.10.1.6 Sauvegarder les repères min-max-ppz-moy 46

II.10.2 Fonctions du menu déroulant d’un subplot 46

II.10.2.1 Changer l’échelle d’un *subplot* 47

II.10.2.2 Nommer les axes d’un *subplot* 47

II.10.2.3 Renommer un *subplot* 47

II.10.2.4 Changer le type du trait d’un *subplot* 47

II.10.2.5 Filtrage d’un *subplot* 48

II.10.3 Autres fonctions de la barre d’outils 48

II.10.3.1 Couper 48

II.10.3.2 Zoomer 49

II.10.3.3 Superposer 49

II.10.3.4 Titre de la fenêtre principale 49

II.11 fonctions de commande sous Matlab 50

III. Exemples de scripts sous TRAP 53

III.1 Exemple de Traitement ACOUStique 53

III.1.1 Exemple de fichier script 53

III.1.2 Résultat obtenu après l’exécution du script 54

III.2 Exemple de Traitement des donnÉes EMA 55

III.2.1 Exemple de fichier script pour EMA 55

III.2.2 Résultat obtenu après l’exécution du script 57

III.3 Exemple de Traitement des donnÉes EPG 58

III.3.1 Exemple de fichier script pour EPG 58

III.3.2 Résultat obtenu après l’exécution du script 59

III.4 Exemple de Traitement des donnÉes TACLE 60

III.4.1 Exemple de fichier script pour TACLE 60

III.4.2 Résultat obtenu après l’exécution du script 61

III.5 Exemple de Traitement des donnÉes EVA 62

III.5.1 Exemple de fichier script pour EVA 62

III.5.2 Résultat obtenu après l’exécution du script 63

III.6 Exemple d’utilisation des étiquettes 64

III.6.1 Exemple de script utilisant un fichier d’étiquettes issu de PRAAT 64

III.6.2 Résultat obtenu après l’exécution du script 65

**Liste des Figures :**

[Figure 1 : exemples de graphes obtenus avec TRAP 6](#_Toc144540291)

[Figure 2 : barre d’outils disponible sur la fenêtre principale 6](#_Toc144540292)

[Figure 3 : fenêtre principale de TRAP 12](#_Toc144540293)

[Figure 4 : fenêtre permettant de définir la fréquence d’échantillonnage du signal acoustique 13](#_Toc144540294)

[Figure 5 : fenêtre permettant de définir le nom des 10 bobines 13](#_Toc144540295)

[Figure 6 : fenêtre permettant de définir l’unité de visualisation des données EMA 14](#_Toc144540296)

[Figure 7 : fenêtre permettant de définir la fréquence de coupure du filtre PBas de Chebyshev 15](#_Toc144540297)

[Figure 8 : fenêtre permettant de définir les signaux EVA à ouvrir 17](#_Toc144540298)

[Figure 9 : fenêtre permettant la configuration de la fenêtre principale 19](#_Toc144540299)

[Figure 10 : fenêtre permettant de régler les paramètres du spectrogramme 20](#_Toc144540300)

[Figure 11 : fenêtre permettant d’ouvrir un fichier de formants existant 20](#_Toc144540301)

[Figure 12 : exemple de traitements acoustiques effectués sous TRAP 22](#_Toc144540302)

[Figure 13 : fenêtre permettant de définir le type de spectre à visualiser 23](#_Toc144540303)

[Figure 14 : choix du sexe du locuteur pour le calcul du cepstre 23](#_Toc144540304)

[Figure 15 : exemples de traitements spectraux sous TRAP 24](#_Toc144540305)

[Figure 16 : fenêtre permettant de choisir la trajectoire de la bobine à visualiser 24](#_Toc144540306)

[Figure 17 : fenêtre de dialogue pour le choix du repérage à effectuer 26](#_Toc144540307)

[Figure 18 : exemple de traitements sur des signaux EMA sous TRAP 28](#_Toc144540308)

[Figure 19 : fenêtre permettant de choisir un fichier de contour du palais 28](#_Toc144540309)

[Figure 20 : exemple de coupe sagittale 30](#_Toc144540310)

[Figure 21 : exemple d’animation de 8 bobines entre 2.01 et 2.48 secondes 30](#_Toc144540311)

[Figure 22 : fenêtre de sélection d’une région 31](#_Toc144540312)

[Figure 23 : fenêtre de sélection de la région n° 1 appelée « front » 31](#_Toc144540313)

[Figure 24 : fenêtre permettant de choisir le fichier EPG à visualiser 32](#_Toc144540314)

[Figure 25 : exemple de traitements possibles sur des signaux EPG avec TRAP 34](#_Toc144540315)

[Figure 26 : électrodes contactés du fichier m009s2.nsp à t = 0.89s 34](#_Toc144540316)

[Figure 27 : série de palais pour 3 échantillons avant et après l’instant t = 0.88s 35](#_Toc144540317)

[Figure 28 : fenêtre permettant de choisir le paramètre à visualiser 35](#_Toc144540318)

[Figure 29 : exemple de traitements sur les signaux TACLE sous TRAP 37](#_Toc144540319)

[Figure 30 : fenêtre permettant de choisir la trajectoire du signal à visualiser 38](#_Toc144540320)

[Figure 31 : exemple de traitements sur les signaux EVA sous TRAP 40](#_Toc144540321)

[Figure 32 : fenêtre permettant de choisir la trajectoire du signal EGG à visualiser 41](#_Toc144540322)

[Figure 33 : fenêtre permettant de choisir le type de filtre à appliquer 42](#_Toc144540323)

[Figure 34 : exemple de traitements sur les signaux EGG sous TRAP 43](#_Toc144540324)

[Figure 35 : fenêtre permettant de définir le nom et le type d’étiquette à poser 44](#_Toc144540325)

[Figure 36 : exemple d’un fichier d’étiquettes sauvegardées 45](#_Toc144540326)

[Figure 37 : exemple de fichier de repères 46](#_Toc144540327)

[Figure 38 : menu déroulant d’un subplot 46](#_Toc144540328)

[Figure 39 : fenêtre permettant de modifier l’échelle du subplot courant 47](#_Toc144540329)

[Figure 40 : fenêtre permettant de modifier le trait du subplot courant 47](#_Toc144540330)

[Figure 41 : fenêtre permettant de spécifier le type de filtrage à appliquer sur le signal du subplot courant 48](#_Toc144540331)

[Figure 42  :liste des fonctions utilisées en mode script sous TRAP 52](#_Toc144540332)

[Figure 43 : résultat des différentes fenêtres obtenues après traitement acoustique. 54](#_Toc144540333)

[Figure 44 : résultat des différentes fenêtres obtenues après traitement des données EMA. 57](#_Toc144540334)

[Figure 45 : résultat des différentes fenêtres obtenues après traitement des données EPG. 59](#_Toc144540335)

[Figure 46 : résultat des différentes fenêtres obtenues après traitement des données TACLE 61](#_Toc144540336)

[Figure 47 : résultat des différentes fenêtres obtenues après traitement des données EVA 63](#_Toc144540337)

[Figure 48 : exemple d’utilisation du fichier de segmentation issu de PRAAT 65](#_Toc144540338)

[Figure 49 : zoom sur une partie du signal 65](#_Toc144540339)

# PrÉsentation du logiciel

TRAP, logiciel intégré pour le traitement de signaux de parole, permet d’effectuer des traitements et des analyses sur différents types de signaux :

* signaux acoustiques au format binaire EMA, EPG, ISIS, Matlab ou WAV.
* signaux articulatoires de type EMA (format binaire .mat AG200 ou WAVE).
* signaux articulatoires de type EPG (format binaire *Kay Elemetrics*).
* signaux aérodynamiques de type EVA (format binaire Europec-EVA).
* signaux articulatoires de type EGG (format binaire issu de la DT3016)
* signaux de mesures labiales de type TACLE (format ASCII).

Il permet de visualiser en même temps des signaux temporels, fréquentiels et paramétriques.

Ce guide est rédigé pour les utilisateurs de Windows, mais TRAP peut fonctionner (avec quelques modifications mineures) sous n’importe quel système d’exploitation.

Remarque : A compter de la version 7 (celle-ci) TRAP ne fonctionne qu’avec une version de Matlab 2015b ou supérieure.

Dernières modifications majeures :

* Mars 2016 : passage à l’EMA 3D
* Sept. 2016 : passage à 16 canaux

## Principe gÉnÉral

Tous les traitements peuvent être lancés soit en mode interactif (à la souris, en utilisant le menu ou la barre des outils), soit en mode commande, en tapant le nom de la commande correspondante dans la fenêtre Matlab. L’utilisateur a ainsi la possibilité de faire des fichiers de commande (scripts), particulièrement utiles pour effectuer des séries de traitements identiques sur plusieurs fichiers. Dans les explications suivantes, nous mentionnons les deux cas en précisant Matlab >> pour la commande équivalente à exécuter sous le prompt Matlab.

TRAP est principalement composé de deux types de fenêtre :

* une *Fenêtre principale* dans laquelle s’affichent tous les signaux temporels, (trajectoire de bobines, signal acoustique, spectrogramme, F0, aire aux lèvres, etc.)
* une *Fenêtre synchrone* dans laquelle s’affichent des informations fréquentielles et positionnelles (FFT, LPC, coupe sagittale, palais, etc.), à un instant donné.

Ces deux fenêtres sont synchronisées à l’aide d’un *curseur temporel* se trouvant dans la fenêtre principale. Les données affichées dans la fenêtre synchrone sont calculées pour l’instant *t* indiqué par la position du curseur dans le *subplot* (cadre) *actif* (encadré en bleu-cyan) de la fenêtre principale. Par exemple, pour rendre le 3ème graphe de la fenêtre principale actif, il suffit de cliquer dessus ou de taper la commande Matlab suivante :

Matlab >> INDEX = 3

Matlab >> active\_graphe

Ou bien, si le *subplot* actif courant est le 2ème *subplot* de la fenêtre principale, pour activer le 3ème on peut simplement taper :

Matlab >> graphe\_suivant

Tous les traitements du menu principal (filtrage, calcul de formants, repérage de minima, etc.) ainsi que l’outil « écouter » se font sur le ou les signaux contenus dans le *subplot* actif.

|  |  |  |
| --- | --- | --- |
| *Fenêtre principale :* |  | *Fenêtre synchrone :* |
|  |  | fensync |

*curseur à t = 1.36 s*

Figure  : exemples de graphes obtenus avec TRAP

## Le menu principal

Le schéma ci-dessous représente le menu principal de TRAP. Chacune des fonctions entraîne l’ouverture d’une boîte de dialogue permettant à l’utilisateur de paramétrer les opérations. Toutes ces fonctions seront décrites en détail plus loin.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Fichier** | **Options** | **Traitement acoustique** | **Traitement EMA** | **Traitement EPG** | **Traitement TACLE** | **Traitement EVA** | **Traitement EGG** | **Fenêtres** |
| Ouvrir ⇨ | Acoustique ⇨ | Détection F0 ⇨ | Trajectoire | Sélection des régions | Trajectoire | Trajectoire | Trajectoire | Principale ⇨ |
| Charger étiquettes | EMA ⇨ | Intensité | Vitesse | Sauvegarder régions | Vitesse | Vitesse | Vitesse | Synchrone ⇨ |
| Enregistrer⇨ | EPG ⇨ | Filtrage ⇨ | Vitesse Tangentielle | Taux de constriction | Accélération | Accélération | Accélération |  |
| Quitter | TACLE/Mesures⇨ | FFT – LPC | Accélération | Trajectoire EPG ⇨ | Filtrage | Filtrage | Filtrage |  |
|  | Filtre PBas (Chebyshev) | Cepstre | Coupe sagittale | Vitesse | Repérage ⇨ | Repérage ⇨ | Repérage ⇨ |  |
|  |  | Formants ⇨ | Animation | Accélération |  |  |  |  |
|  |  | Spectrogramme | Repérage ⇨ | Palais ⇨ |  |  |  |  |
|  |  | Afficher Signal | Facteurs Tilt | Repérage ⇨ |  |  |  |  |
|  |  | Ecouter ⇨ |  |  |  |  |  |  |

Tableau  : ensemble des fonctions disponibles via le menu de la fenêtre principale

## La barre d’outils

La fenêtre principale contient une barre d’outils qui s’appliquent sur l’ensemble des signaux contenus dans tous les *subplots* de la fenêtre principale. Ces outils seront décrits en détail plus loin. Seuls les outils « Ecouter », « Copier » et « Coller » ne s’appliquent qu’au signal contenu dans le *subplot* actif.

barre_outils

Figure  : barre d’outils disponible sur la fenêtre principale

## Description des variables globales

Le logiciel TRAP utilise beaucoup de variables globales permettant ainsi à l’utilisateur de modifier leur valeur directement sous le prompt Matlab.

Le tableau suivant répertorie donc toutes les variables globales du logiciel avec leur valeur d’initialisation et une bref description.

|  |  |  |
| --- | --- | --- |
| **Nom** | **Initial.** | **Description** |
| COL\_SYNC | 0 | Fractionnement en colonne de la fenêtre synchrone. |
| NUM\_SYNC | 0 | Numéro de la fenêtre synchrone courante. |
| IND\_SYNC | 0 | Index du *subplot* courant dans la fenêtre synchrone. |
| NUM\_FEN | 0 | Numéro de la fenêtre principale courante. |
| INDEX | 1 | Index du *subplot* courant dans la fenêtre principale. |
| NB\_COL | 0 | Fractionnement en colonne de la fenêtre principale. |
| NB\_LIG | 0 | Fractionnement en ligne de la fenêtre principale. |
| NIV\_ETIQ | 5 | Nombre total de niveau d’étiquettes. |
| TYPE\_ETIQ | Cf. ci-contre | Nom des différents types d’étiquettes possibles :  TYPE\_ETIQ(1,:)=’Phonétique’  TYPE\_ETIQ(2,:)=’Tonal’  TYPE\_ETIQ(3,:)=’Break Index’  TYPE\_ETIQ(4,:)=’Mismatch’  TYPE\_ETIQ(5,:)=’Commentaire’ |
| COULEUR\_ETIQ | Cf. ci-contre | Couleur des étiquettes.  COULEUR\_ETIQ(1,:)=[0 3\*0.99/4 0]  COULEUR\_ETIQ(2,:)=[0 7\*0.99/8 0]  COULEUR\_ETIQ(3,:)=[0 11\*0.99/12 0]  COULEUR\_ETIQ(4,:)=[0 15\*0.99/16 0]  COULEUR\_ETIQ(5,:)=[0 19\*0.99/20 0] |
| COULEUR\_GRAPH | [0 0 1] | Couleur des graphes (bleu). |
| COULEUR\_FOND | [1 1 1] | Couleur du fond (blanc). |
| COULEUR\_ACTIVE | [0 1 1] | Couleur des axes du graphe actif (cyan). |
| COULEUR\_TITRE\_FEN | [0 0 0] | Couleur du titre de la fenêtre (noir) |
| COULEUR\_POINTEUR | [1 0 0] | Couleur du pointeur (rouge) |
| COULEUR\_AXES | [0 0 0] | Couleur des axes (noir). |
| COULEUR\_MAX | [1 0.6 0.2] | Couleur des repères de maxima (vert). |
| COULEUR\_MIN | [0.5 1 0.5] | Couleur des repères de minima (orange). |
| COULEUR\_ZERO | [1 1 0] | Couleur des repères de passages par zéro (jaune). |
| COULEUR\_MOY | [0 1 1] | Couleur des repères de moyenne (cyan). |
| FLAG\_ZOOM | 0 | Drapeau pour savoir si un zoom a été effectué. |
| NB\_ZOOM | 1 | Indique le nombre de zoom déjà effectué |
| ZOOM\_PREC\_DEB | vide | Valeur temporelle de début du zoom précédent |
| ZOOM\_PREC\_FIN | vide | Valeur temporelle de fin du zoom précédent |
| REP\_SEUIL | 10 | Réglage du seuil pour le repérage min – max des données EMA, EPG, EVA et des mesures. |
| SUPERPOSE | 0 | Drapeau pour savoir si l’on est en mode superposition de graphes. |
| CURSEUR | 1 | Drapeau pour savoir si on affiche le curseur de mesure. |
| CHEBY\_FREQ\_COUP | 20 | Fréquence de coupure du filtre de Chebyshev |
| ACOU\_CALC\_CEPS | 0 | Drapeau permettant de savoir si le cepstre doit être calculé et affiché. |
| ACOU\_CALC\_FFT | 0 | Drapeau permettant de savoir si la FFT doit être calculée et affichée. |
| ACOU\_CALC\_LPC | 0 | Drapeau permettant de savoir si la LPC doit être calculée et affichée. |
| ACOU\_CEPS\_VIS | 4 | Taille de la fenêtre du filtrage du cepstre (4ms pour homme, et 3.2ms pour femme). |
| ACOU\_FECH | 20000 | Fréquence d’échantillonnage du signal acoustique. |
| ACOU\_FEN\_TMP | Hanning | Fenêtre temporelle pour calcul de la FFT et du sonagramme.  ACOU\_FEN\_TMP=hanning (ACOU\_LONG\_FEN) |
| ACOU\_LONG\_FEN | 128 | Longueur de la fenêtre temporelle pour tous les traitements acoustiques instantanés (FFT, LPC, sonagramme, cepstre). |
| ACOU\_OVER\_FEN | 64 | Pas de déplacement de la fenêtre temporelle pour le sonagramme |
| ACOU\_NOM\_FIC | Vide | Nom du fichier binaire contenant les données acoustiques chargées. |
| ACOU\_NOM\_AFFIC | Vide | Nom du fichier binaire contenant les données acoustiques corrigé pour affichage correct à l’écran. |
| ACOU\_STEP\_FRM | 0.010 | Pas du traitement formantique en seconde. |
| ACOU\_LENGTH\_FRM | 0.032 | Longueur de la fenêtre pour le traitement formantique en seconde. |
| ACOU\_TPS\_FRM | 0 | Valeurs sur l’échelle des temps pour les 5 formants. |
| ACOU\_DATA\_FRM | 0 | Valeurs des 5 premiers formants. |
| ACOU\_ORD\_FFT | 512 | Nombre d’échantillons pour le calcul de la FFT. |
| ACOU\_ORD\_LPC | 22 | Nombre de coefficients pour le calcul de la LPC. |
| ACOU\_SIG | Vide | Signal acoustique. |
| ACOU\_SIG\_CUT | Vide | Signal acoustique coupé. |
| ACOU\_SPECTRO | Vide | Spectrogramme du signal acoustique. |
| ACOU\_TPS | Vide | Valeurs sur l’échelle des temps pour le signal acoustique. |
| ACOU\_TPS\_CUT | Vide | Valeurs sur l’échelle des temps pour le signal acoustique coupé. |
| ACOU\_TYP\_FIC | Vide | Type du fichier acoustique binaire chargé (EMA, EPG, ISIS, Matlab .mat, WAV, binaire quelconque). |
| EMA\_FLAG | 0 | Drapeau permettant de savoir si les données EMA ont déjà été traitées. |
| EMA\_BOB1..16 | ’B1..16’ | Noms des bobines de 1 à 16 |
| EMA\_COUP\_SAG | 0 | Drapeau permettant de savoir si la coupe sagittale EMA doit être affichée. |
| EMA\_COUP\_SAG\_3D | 0 | Drapeau permettant de savoir si la coupe sagittale EMA en 3D doit être affichée. |
| EMA\_NUM\_SAG | 0 | Numéro de la fenêtre contenant l’animation des bobines EMA pour la coupe sagittale. |
| EMA\_ECHELLE | 1 | Facteur d’échelle permettant une visualisation en *cm* ou *mm* (1 pour *cm*, 0.1 pour *mm*). |
| EMA\_FECH | 500 | Fréquence d’échantillonnage du signal EMA. |
| EMA\_FIC\_PAL | Vide | Nom du fichier d’enregistrement du contour du palais. |
| EMA\_BOB\_PAL | 0 | Numéro de la bobine de suivi du palais. |
| EMA\_PALAIS | Vide | Valeurs en x et y du contour du palais à enregistrer. |
| EMA\_PAL\_X | Vide | Trajectoire de la bobine de suivi du palais en X. |
| EMA\_PAL\_Y | Vide | Trajectoire de la bobine de suivi du palais en Y. |
| EMA\_PAL\_Z | Vide | Trajectoire de la bobine de suivi du palais en Z. |
| EMA\_SIG | Vide | Signal EMA non traité. |
| EMA\_TILT | 0 | Numéro de la fenêtre contenant les tilts des bobines EMA. |
| EMA\_TPS | Vide | Valeurs sur l’échelle des temps pour le signal EMA. |
| EMA\_X1 ... 16 | Vide | Trajectoires des 16 bobines en X. |
| EMA\_Y1 ... 16 | Vide | Trajectoires des 16 bobines en Y. |
| EMA\_Z1 ... 16 | Vide | Trajectoires des 16 bobines en Z. |
| EMA\_TILT1 ... 10 | Vide | Tilts des 10 bobines. |
| COUL\_B1..16 |  | Couleur de visualisation des bobines |
| EPG\_CHANNEL | Vide | Correspondance entre les canaux et les électrodes. |
| EPG\_DATA\_CONS | Vide | Données contenues dans les fichiers de constrictions chargés. |
| EPG\_FECH | 100 | Fréquence d’échantillonnage du signal EPG. |
| EPG\_NB\_CONS | 0 | Nombre de fichiers de constrictions chargés. |
| EPG\_NB\_ELEC | 96 | Nombre d’électrodes sur le palais pour les données EPG |
| EPG\_NB\_PAL | 3 | Nombre de palais à visualiser avant et après celui du curseur. |
| EPG\_NOM\_CONS | Vide | Nom des différents types de constrictions. |
| EPG\_NOM\_LOC | Vide | Nom du locuteur, pour les fichiers EPG. |
| EPG\_NOM\_REG | Vide | Noms des différentes régions. |
| EPG\_PAL\_FEN | 0 | Numéro de la fenêtre d’affichages des palais série. |
| EPG\_PAL\_SERIE | 0 | Drapeau pour l’affichage du palais série. |
| EPG\_PAL\_SIMPLE | 0 | Drapeau pour l’affichage du palais simple. |
| EPG\_POS\_ELEC | 0 | Matrice des positions des électrodes sur le palais en EPG. |
| EPG\_REGIONS | Vide | Appartenance des différentes électrodes aux différentes régions. |
| EPG\_SIG | Vide | Signal EPG. |
| EPG\_TPS | Vide | Valeurs sur l’échelle des temps pour le signal EPG. |
| MES\_FECH | 50 | Fréquence d’échantillonnage des données contenues dans le fichier de mesures. |
| MES\_NOM\_FIC | Vide | Nom du fichier ASCII contenant la ou les mesures chargées. |
| MES\_TPS | Vide | Valeurs sur l’échelle des temps pour la mesure générale. |
| MES\_NB | 0 | Nombre de mesures dans le fichier de mesures générales. |
| MES\_NOM | Vide | Nom des différentes mesures dans le fichier de mesures générales . |
| MES\_DATA | Vide | Données contenues dans le fichier de mesures générales. |
| EVA\_SIG | Vide | Signal EVA non traité. |
| EVA\_TPS1 | Vide | Valeurs sur l’échelle des temps pour les signaux EVA :  WA1, WA2, F0, EGG. |
| EVA\_TPS2 | Vide | Valeurs sur l’échelle des temps pour les signaux EVA :  OAF, NAF, PR1, PR2. |
| EVA\_FECH1 | 25000 | Fréquence d’échantillonnage des signaux EVA : WA1, WA2, F0, EGG. |
| EVA\_FECH2 | 6250 | Fréquence d’échantillonnage des signaux EVA : OAF, NAF, PR1, PR2. |
| FE\_ACTIF | 0 | Fréquence d’échantillonnage du signal EVA en cours d’analyse. |
| TPS\_EVA\_ACTIF | 0 | Valeurs sur l’échelle des temps du signal EVA en cours d’analyse. |
| FLAG\_EVA | 0 | Drapeau permettant de savoir si les données EVA ont déjà été traitées. |
| NB\_EVA | 0 | Nombre de données EVA analysées. |
| EVA\_WA1 | vide | Valeurs du signal acoustique canal1. |
| EVA\_WA2 | vide | Valeurs du signal acoustique canal2. |
| EVA\_F0 | vide | Valeurs de la fréquence fondamentale. |
| EVA\_EGG | vide | Valeurs de l’EGG. |
| EVA\_OAF | vide | Valeur du débit d’air oral. |
| EVA\_NAF | vide | Valeur du débit d’air nasal. |
| EVA\_PR1 | vide | Valeur de la pression supra glottique 1. |
| EVA\_PR2 | vide | Valeur de la pression supra glottique 2. |
| EGG\_1 | vide | Valeurs des signaux EGG du canal 1. |
| EGG\_2 | vide | Valeurs des signaux EGG du canal 2. |
| EGG\_3 | vide | Valeurs des signaux EGG du canal 3. |
| EGG\_4 | vide | Valeurs des signaux EGG du canal 4. |
| NB\_EGG | 4 | Nombre de données EGG analysées. |
| EGG\_NOM1..4 | canal1..4 | Nom des 4 canaux EGG. |
| EGG\_TPS | vide | Valeurs sur l’échelle des temps des signaux EGG. |
| EGG\_FECH | 50000 | Fréquence d’échantillonnage des signaux EGG. |
|  |  |  |

Tableau : liste des variables globales du programme

## Quelques prÉcautions À prendre

* Il faut obligatoirement mettre à jour les options avant d’ouvrir un fichier. Si par exemple en cours d’utilisation, on change la fréquence d’échantillonnage, il faut ré-ouvrir le fichier concerné. De même si l’on change le nom du locuteur associé à un fichier EPG. Par contre si l’on change des options concernant un fichier EMA déjà ouvert (à part sa fréquence d’échantillonnage), il suffit de re-sélectionner le sous-menu "Trajectoire" du menu "Traitement EMA", ou d’appeler la fonction *ema\_traitement* sous le prompt Matlab.
* TRAP offre la possibilité de gérer plusieurs fenêtres principales. Pour changer de fenêtre principale, il suffit de cliquer sur celle dans laquelle on désire afficher le prochain signal. Par contre, il ne peut gérer qu’une seule fenêtre synchrone.
* Lorsque TRAP demande à l’utilisateur de rentrer un nom de fichier, il faut qu’il donne tout le chemin du fichier si celui-ci ne se trouve pas dans le répertoire courant (la plupart des cas). En particulier pour donner le nom du locuteur (option pour les signaux EPG) qui est en fait le nom du fichier *.usr*.
* Les noms de fichiers doivent avoir une extension d’exactement **3 caractères** (comme ‘.wav’, ‘.mat’, ‘.eva’, ‘.tac’, ‘.dcf’).
* Pour rendre un *subplot* actif (qu’il soit encadré de bleu), il faut cliquer dans le *subplot*, mais pas directement sur le graphe tracé dans le *subplot*.
* Avant d’effectuer un *zoomer* ou *couper* sur une partie du graphe affiché, il faut d’abord que le *subplot,* dans lequel ce graphe est tracé, soit actif, c’est-à-dire qu’il soit encadré de bleu. Cliquer dessus si nécessaire (ou utiliser la commande *active\_graphe*, cf. I.1).
* La détection de F0, de formants, le calcul du sonagramme, de la LPC ou de la FFT, se fait sur le *dernier* signal acoustique tracé. Si celui-ci a été coupé, ces traitements se feront sur la partie de signal restant après la coupe.

# Utilisation du logiciel

## Lancement du logiciel

Il faut d’abord lancer Matlab et préciser le chemin d’accès adéquat (ou *path* spécifiant où se trouve le programme Trap ) puis sous le prompt Matlab taper :

Matlab >> trap

La fenêtre suivante apparaît sur l’écran :

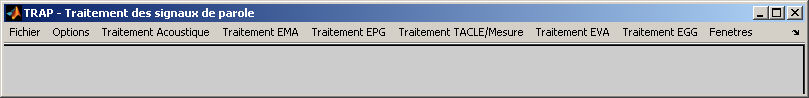


Figure  : fenêtre principale de TRAP

TRAP est composé des neuf menus principaux suivants :

* le menu ***Fichier*** : permet d’ouvrir et d’enregistrer des fichiers,
* le menu ***Options*** : permet de configurer les options des différents signaux,
* le menu ***Traitement acoustique*** : permet d’effectuer différents traitements sur des signaux acoustiques,
* le menu ***Traitement EMA*** : permet d’effectuer différents traitements sur des signaux provenant de l’articulographe EMA de Carstens,
* le menu ***Traitement EPG*** : permet d’effectuer différents traitements sur des signaux provenant de l’électropalatographe de Kay Elemetrics,
* le menu ***Traitement TACLE/Mesure*** : permet d’effectuer différents traitements sur des paramètres labiaux (aires aux lèvres S, protrusion P1 et P2, etc.) au format TACLE ou sur des mesures physiologiques quelconques (au format ASCII),
* le menu ***Traitement EVA*** : permet d’effectuer différents traitements sur des signaux aérodynamiques provenant d’Europec-EVA,
* le menu ***Traitement EGG*** : permet d’effectuer différents traitements sur des signaux EGG provenant de la carte d’acquisition DT3016 via le logiciel Scope,
* le menu ***Fenetres*** : permet de paramétrer les fenêtres de visualisation des signaux.

Pour quitter le logiciel, aller dans le menu « Fichier » puis le sous-menu « Quitter ».

Matlab >> quitter

### Initialisation des options

Avant d’ouvrir un fichier il est nécessaire d’initialiser les options correspondant au type de fichier que l’on désire ouvrir. Ceci se fait à l’aide du menu « Options ».

#### Options pour les données acoustiques

Il convient de préciser la fréquence d’échantillonnage du signal acoustique que l’on veut tracer. Sinon, la base de temps sur l’axe des abscisses risque d’être fausse et les données articulatoires éventuellement associées (EMA, EPG) ne seront pas correctement synchronisées.

Lorsque l’on choisit « Options » puis « Acoustique » puis « Fréquence d’échantillonnage », la fenêtre ci-dessous apparaît (Figure 4) :



Figure  : fenêtre permettant de définir la fréquence d’échantillonnage du signal acoustique

Il suffit de rentrer au clavier la bonne valeur et de cliquer sur « OK » ou taper au clavier sur la touche « Enter ». Par défaut, cette valeur est de 20000 Hz.

En mode script ou sous le prompt Matlab :

Matlab >> ACOU\_FECH = 20000

#### Options pour les données EMA

Si l’on désire ouvrir un fichier EMA, il faut aller dans le sous-menu « EMA » et rentrer la fréquence d’échantillonnage, le nom des bobines et l’unité de visualisation des données.

La fenêtre pour la fréquence d’échantillonnage a la même apparence que celle des signaux acoustiques. La valeur d’échantillonage par défaut des signaux EMA vaut 200 Hz.

Lorsqu’on choisit « Nom des bobines », la fenêtre suivante apparaît (Figure 5) :

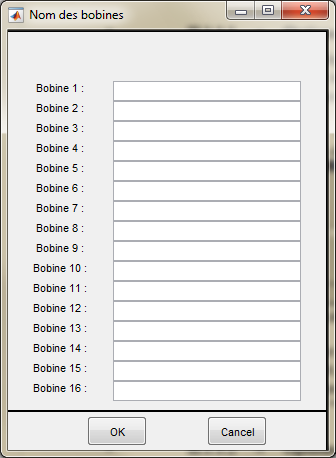


Figure  : fenêtre permettant de définir le nom des 10 bobines

Pour remplir cette boîte de dialogue, il faut rentrer le nom des 10 bobines (de haut en bas). Si une bobine n’a pas été utilisée, laisser la case vide.

En mode script ou sous le prompt Matlab :

Matlab >> EMA\_FECH = 200;

Matlab >> EMA\_BOB1 = 'NEZ';

Matlab >> EMA\_BOB2 = 'INC-SUP';

Matlab >> EMA\_BOB3 = ''; %Pour une bobine non utilisée

Matlab >> EMA\_BOB4 = 'LEV-SUP';

Etc…

NOTE : un filtrage des données EMA est réalisé par défaut avec un filtre de Chebyshev à 20 Hz. Pour ne pas opérer de filtrage, il faut spécifier une valeur de filtrage nulle : CHEBY\_FREQ\_COUP = 0 (cf. II.1.1.5).

Il est possible de visualiser les données en millimètre (centimètre par défaut). Pour cela, aller dans le menu « Options » puis « EMA » puis « Unité de visualisation ». Il suffit ensuite de cliquer sur l’unité désirée dans la fenêtre suivante (Figure 6) :

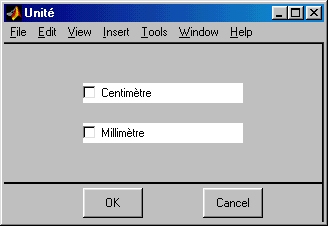


Figure  : fenêtre permettant de définir l’unité de visualisation des données EMA

En mode script ou sous le prompt Matlab :

Matlab >> EMA\_ECHELLE = 0.1 % pour une echelle en millimètre

Puisque les données EMA sont associées à des données acoustiques, il faut aussi penser à rentrer la fréquence d’échantillonnage des données acoustiques associées en sélectionnant menu « Options » puis « Acoustique » puis « Fréquence d’échantillonnage ».

En mode script ou sous le prompt Matlab :

Matlab >> ACOU\_FECH = 20000

#### Options pour les données EPG

Si l’on désire ouvrir un fichier EPG, il faut obligatoirement donner le nom du locuteur qui permettra d’ouvrir le fichier *.usr*. Pour cela, il faut aller dans le sous-menu « EPG » et cliquer sur « Nom du locuteur ». On peut aussi rentrer la fréquence d’échantillonnage des données EPG, par défaut, elle est de 100Hz.

Comme pour les données EMA, il faut aussi penser à rentrer la fréquence d’échantillonnage des données acoustiques associées en sélectionnant menu « Options » puis « Acoustique » puis « Fréquence d’échantillonnage ».

En mode script ou sous le prompt Matlab :

Matlab >> EPG\_NOM\_LOC = 'D:\trap\exemples\data\';

Matlab >> ACOU\_FECH = 12800;

Matlab >> EPG\_FECH = 100;

#### Options pour les données TACLE

Si l’on désire ouvrir un fichier de paramètres labiaux (.tac), il convient de préciser d’abord les 3 paramètres suivants dans le sous-menu « TACLE/Mesures » :

* la fréquence d’échantillonnage (sous-menu « Fréquence d’échantillonnage »), par défaut cette fréquence est fixée à 50 Hz ;
* le nombre total de mesures contenues dans le fichier (sous-menu « Nombre de params/mesures »);
* les noms de ces mesures (sous-menu « Nom des params/mesures »), en prenant soin de respecter le même ordre que celui du fichier .tac à savoir : A, B, S, P1, P2 et C.

En mode script ou sous le prompt Matlab, pour visualiser 2 paramètres (S et P1) :

Matlab >> MES\_FECH = 50

Matlab >> MES\_NB = 2

Matlab >> MES\_NOM(1,1:length('S')) = 'S';

Matlab >> MES\_NOM(2,1:length('P1')) = 'P1';

#### Options pour le filtrage de Chebyshev

Avant de lancer un filtrage passe-bas par Chebyshev, il est nécessaire de rentrer la fréquence de coupure de ce filtre. Ceci peut se faire par le sous-menu « Filtrage PBas (Chebyshev) », une fenêtre identique à celle ci-dessous apparaît (Figure 7). Il faut ensuite rentrer la fréquence de coupure du filtre.

Par défaut celle-ci est fixée à 20 Hz et si cette valeur n’est pas modifiée via ce sous-menu, elle sera utilisée pour tous les filtrages opérés par la suite.



Figure  : fenêtre permettant de définir la fréquence de coupure du filtre PBas de Chebyshev

En mode script ou sous le prompt Matlab :

Matlab >> CHEBY\_FREQ\_COUP = 20

### Ouvrir un fichier

Dans le menu « Fichier » choisissez le sous-menu « Ouvrir » . On a alors le choix entre l’ouverture d’un fichier « acoustique » (binaire), un fichier EMA (.mat), un fichier de paramètres TACLE (.tac), des fichiers EVA, des fichiers EPG (.nsp) ou un fichier de mesures (ASCII).

Pour les données EPG, l’ouverture du fichier acoustique correspond en réalité à l’ouverture du fichier binaire contenant à la fois les données acoustiques et articulatoires.

#### Ouverture d’un fichier acoustique binaire

TRAP peut ouvrir les fichiers acoustiques binaires de types suivants : *.mXX* (ancienne norme EMA)*, .nsp* (EPG)*, .ech* (ISIS)*, .mat* (binaire *Matlab*), *.wav* (classique)et autre (traité comme un binaire *Matlab*). Cocher la case contenant le type de fichier que vous souhaitez ouvrir. Puis, dans la case d’édition adjacente, entrer au clavier le nom du fichier acoustique avec extension que vous désirez ouvrir avec son chemin complet. Les extensions des fichiers sont rappelées pour guider l’utilisateur.

Si le fichier a l’extension *.mXX* et le type EMA, TRAP ouvrira aussi le fichier EMA d’extension *.0XX* contenant les données articulatoires associées. Cependant, cette façon de faire est obsolète et déconseillée.

Cliquez ensuite sur « OK » ou tapez « Enter ».

Si plusieurs cases ont été cochées, seule la dernière sera prise en compte.

En mode script ou sous le prompt Matlab :

Matlab >> acou\_ouv(nom\_fichier, type\_fichier);

Où type\_fichier est à choisir parmi : ’EMA’, ’EPG’, ’ECH’, ’MAT’, ’WAV’, ’BIN’.

#### Ouverture d’un fichier articulatoire EMA

Dans ce menu, on ne considère que les nouveaux formats pour les fichiers EMA. À savoir, les fichiers matlab (.mat) issus du programme de post-traitement emmaproa. Pour ouvrir ces fichiers il suffit donc de rentrer le chemin et le nom avec extension du fichier .mat.

En mode script ou sous le prompt Matlab :

Matlab >> ema\_ouv\_mat(nom\_fich\_ema);

#### Ouverture d’un fichier de paramètres TACLE

Ce menu ouvre les fichiers (.tac) issus de Tacle. Il est nécessaire auparavant de préciser le nombre de paramètres voulus et leurs noms (cf. II.1.1.4).

En mode script ou sous le prompt Matlab :

Matlab >> tac\_ouv(nom\_fichier\_tac);

#### Ouverture d’un fichier de mesures ASCII

Les fichiers de mesures (.mes) sont des fichiers ASCII pouvant contenir plusieurs mesures articulatoires en colonne. Chaque colonne contient les données articulatoires en fonction du temps pour une mesure donnée. Avant d’ouvrir un fichier de mesures au format ASCII, il faut spécifier un certain nombre de paramètres, via le menu « Options », sous-menu « TACLE/Mesures » :

* la fréquence d’échantillonnage des données contenues dans le fichier ;
* le nombre total de mesures (colonnes) contenues dans le fichier.

En mode script ou sous le prompt Matlab, pour visualiser 4 mesures :

Matlab >> MES\_FECH = 50;

Matlab >> MES\_NB = 4;

Matlab >> mes\_ouv(nom\_fich\_lev);

#### Ouverture des fichiers EVA

Ce menu permet d’ouvrir plusieurs fichiers issus d’EVA. Il ouvre une fenêtre identique à celle ci-dessous (Figure 8) :

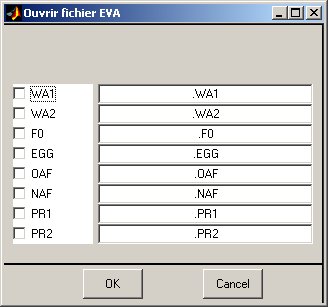


Figure  : fenêtre permettant de définir les signaux EVA à ouvrir

Il faut dans un premier temps cocher le type de fichier à ouvrir et ensuite spécifier pour chacun le nom du fichier avec extension.

Note : pour ce type de fichier EVA, il n’est pas nécessaire de spécifier la fréquence d’échantillonnage des signaux car celle-ci varie en fonction du type de données EVA et TRAP la redéfinit automatiquement en fonction de l’extension du fichier.

En mode script ou sous le prompt Matlab :

Matlab >> eva\_ouv(nom\_fich\_wa1,'WA1'); %Fichier Wav principal

Matlab >> eva\_ouv(nom\_fich\_wa2,'WA2'); %Fichier Wav secondaire

Matlab >> eva\_ouv(nom\_fich\_f0,'F0'); %Fichier contenant les valeurs de fréquence fondamentale

Matlab >> eva\_ouv(nom\_fich\_egg,'EGG'); %Fichier contenant les valeurs issus de l’Electro Glotto Graphe

Matlab >> eva\_ouv(nom\_fich\_oaf,'OAF'); %Fichier de débit d’air oral

Matlab >> eva\_ouv(nom\_fich\_naf,'NAF'); %Fichier de débit d’air nasal

Matlab >> eva\_ouv(nom\_fich\_pr1,'PR1'); %Fichier de pression principale

Matlab >> eva\_ouv(nom\_fich\_pr2,'PR2'); %Fichier de pression secondaire

#### Ouverture d’un fichier EGG

Ce menu permet d’ouvrir les fichiers EGG issus de la carte DT3016 via le logiciel Scope au format DCF (*Data Communication Facility)*.

En mode script ou sous le prompt Matlab :

Matlab >> egg\_ouv(nom\_fichier\_dcf);

### II 1.3 Enregistrer des données temporelles dans un fichier au format binaire ou ASCII

Après avoir effectué des traitements sur un fichier temporel, ou l’avoir coupé, on peut vouloir sauvegarder les résultats de ces traitements. S’il s’agit de données acoustiques, on peut souhaiter les sauvegarder dans un format binaire.

Pour cela, dans le menu « Fichier » choisir le sous-menu « Enregistrer » puis « Signal temporel ». Il faut alors fournir le nom du fichier dans lequel les données temporelles seront enregistrées. Puis il faut cocher un des 3 types de fichier possibles. Deux formats binaires sont proposés au choix : ISIS (.ech avec en-tête contenant la fréquence d’échantillonnage) ou Matlab (.mat). Le format ASCII simple est également proposé. Il reste ensuite à spécifier la fréquence d’échantillonnage des données.

En mode script ou sous le prompt Matlab :

Matlab >> enr\_tempo(nom\_fichier, type, freq\_ech);

Où type\_fichier est à choisir parmi :’ECH’, ’MAT’, ’ASC’.

S’il s’agit de repérages (étiquettes, min-max-ppz ou formants), le fichier créé devra être au format ASCII (type\_fichier=’ASC’). Il est donc préférable de rentrer un nom de fichier avec une extension .txt.

En mode script ou sous le prompt Matlab :

Matlab >> enr\_etiq(nom\_fich\_etiq) ;

Matlab >> enr\_minmaxppz(nom\_fich\_min);

Matlab >> enr\_formant(nom\_fich\_frm); %enregistre les formants en mode ‘ASC’ (cf. II.3.1.5).

## Configuration des fenÊtres d’affichage

L’utilisateur a la possibilité de configurer deux types de fenêtre : la fenêtre principale qui permet d’afficher toutes les données temporelles, et la fenêtre synchrone permettant d’afficher les données fréquentielles et paramétriques à un instant donné, déterminé par le curseur de temps dans la fenêtre principale. Il existe une configuration par défaut pour ces deux types de fenêtre. On peut les ouvrir en allant dans le menu « Fenetre » et en choisissant « Défaut » du sous-menu « Principale » ou « Synchrone ».

### Fenêtre principale

Pour configurer ce type de fenêtre, il faut choisir « Configurer » du sous-menu « Principale ». La boîte de dialogue suivante apparaît (Figure 9) :

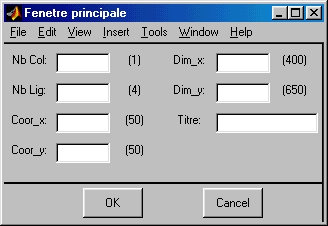


Figure  : fenêtre permettant la configuration de la fenêtre principale

Le nombre de lignes et le nombre de colonnes correspondent à la disposition et au nombre de graphiques que l’on désire afficher dans la fenêtre principale (*Nb Lig × Nb Col* graphiques).

Les coordonnées en x et en y correspondent à la position du coin en haut et à gauche de la fenêtre, en pixel, par rapport au même coin de l’écran.

Les dimensions en x et en y correspondent à la largeur et à la longueur de la fenêtre en pixels.

Les valeurs par défaut sont mises entre parenthèses à côté de chaque champ.

Si l’utilisateur a choisi de créer une fenêtre principale avec plusieurs colonnes, le curseur temporel apparaîtra sous la colonne la plus à gauche.

Matlab >> fenetre(nb\_col, nb\_lig, [coord\_x coord\_y], [dim\_x dim\_y], titre)

### Fenêtre synchrone

Pour configurer ce type de fenêtre, il faut choisir « Configurer » du sous-menu « Synchrone ». Le même type de boîte de dialogue apparaît excepté qu’il n’y a pas à rentrer le nombre de ligne puisqu’il en a qu’une seule dans ce type de fenêtre, étant donné que le nombre maximum de graphes à afficher dans cette fenêtre est de quatre.

Matlab >> synchrone(nb\_col, [coord\_x coord\_y], [dim\_x dim\_y], titre)

## Traitements sur les signaux acoustiques

Une fois que les fenêtres d’affichage sont ouvertes, il est possible d’effectuer les traitements.

### Traitements s’affichant dans la fenêtre principale

#### Affichage du signal acoustique

Dans le menu « Traitement acoustique », choisir le sous-menu « Signal ». Le signal acoustique apparaîtra dans la fenêtre principale. La base de temps sur l’axe des abscisses sera fonction de la fréquence d’échantillonnage spécifiée au préalable (cf. II.1.1.1).

Matlab >> acou\_signal

#### Filtrage du signal acoustique

On peut effectuer soit un filtrage de Chebyshev (passe-haut ou passe-bas) soit un Filtrage passe-haut de Remez. Dans tous les cas, il faut aller dans le menu « Traitement acoustique », choisir le sous-menu « Filtrage », puis cliquer sur le type de filtrage que l’on désire effectuer. Pour le filtre de Chebyshev, une boîte de dialogue s’ouvre dans laquelle il faut rentrer la fréquence de coupure. Pour le filtre de Remez, on ne peut pas modifier la fréquence de coupure mais il est nécessaire de rentrer une valeur (0) en mode script. Après avoir cliqué sur « OK », le signal filtré s’affiche.

Matlab >> acou\_filtres(type\_filtre,freq\_coupure)

type\_filtre = 1 : filtre passe-haut Chebyshev, ex : acou\_filtres(1, 10),

type\_filtre = 2 : filtre passe-bas Chebyshev, ex : acou\_filtres(2, 20),

type\_filtre = 3 : filtre passe-haut Remez , ex : acou\_filtres(3, 0).

#### Evolution de l’intensité du signal (RMS)

Dans le menu « Traitement acoustique », choisir le sous-menu « Intensité ». Elle apparaîtra dans la fenêtre principale.

Matlab >> acou\_intensite

#### Spectrogramme

Dans le menu « Traitement acoustique », choisir le sous-menu « Spectrogramme ». Il s’affichera dans la fenêtre principale. De plus, la fenêtre suivante (Figure 10) apparaîtra afin de régler la pré-emphase et le niveau de gris du spectrogramme.



Figure  : fenêtre permettant de régler les paramètres du spectrogramme

Matlab >> acou\_spectro

#### Affichage des formants

On peut afficher les formants soit en faisant une détection des formants du signal, soit en chargeant un fichier contenant déjà ces formants. Dans les deux cas, il faut aller dans le menu « Traitement acoustique », choisir le sous-menu « Formants ».

Si l’on clique ensuite sur « Détection des formants », ceux-ci s’affichent sur le spectrogramme (si SUPERPOSE = 1) après un petit temps de calcul. **Remarque** : il est conseillé de sauvegarder les formants ainsi détectés cf. page 18.

Si l’on clique sur « A partir du fichier », la boîte de dialogue (Figure 11) apparaît. Dans cette boîte, il faut rentrer le nom du fichier contenant les formants, puis cliquer sur le type de fichier correspondant : ECH ou ASC.



Figure  : fenêtre permettant d’ouvrir un fichier de formants existant

Matlab :

- par détection :

Matlab >> SUPERPOSE = 1

Matlab >> acou\_detect\_form

Puis enregistrement : enr\_formant(nom\_fich\_frm) ;

- par chargement de fichier :

Matlab >> SUPERPOSE = 1

Matlab >> acou\_affich\_form(nom\_fich,type\_fich)

avec type\_fich : ’ASC’ ou ’ECH’.

Le type ’ASC’ correspond à un fichier ASCII contenant 2 types d’information : les valeurs temporelles dans la première colonne et les 5 valeurs formantiques dans les colonnes suivantes.

Le type ’ECH’ est un fichier de formants typique au format ISIS.

#### Affichage de la fréquence fondamentale F0[[1]](#footnote-1)

On peut afficher la F0 soit en faisant une détection sur le signal (par appel de la fonction C *f0\_pca* qui utilise une méthode de passage par zéro avec filtrage adaptatif), soit en chargeant un fichier contenant la F0. Dans les deux cas, il faut aller dans le menu « Traitement acoustique » et choisir le sous-menu « Détection F0 ».

- Si l’on clique sur « Calcul de la F0 », celle-ci s’affiche après un temps de calcul.

- Si l’on clique sur « A partir du fichier », la même boîte de dialogue que précédemment apparaît (cf. Figure 11). Après avoir donné le nom du fichier et son type, la F0 s’affiche. Le type ‘ECH’ correspond à un fichier de valeurs de F0 au format ISIS, le type ‘ASC’ à un fichier ASCII à 2 colonnes, la première contenant les valeurs instants, la seconde les valeurs de F0 à ces instants.

Matlab :

- par détection :

Matlab >> acou\_detect\_f0

- par chargement de fichier :

Matlab >> acou\_affich\_f0(nom\_fich, type\_fich)

avec type\_fich : ’ECH’ ou ’ASC’.

#### Ecoute du signal

TRAP permet d’écouter soit le dernier signal acoustique chargé (en entier, même s’il a été coupé depuis), soit la partie du signal contenue dans le *subplot* actif, soit uniquement un extrait de ce même signal. Pour cela, il faut aller dans le menu « Traitement acoustique » puis « Ecouter » puis sélectionner « Signal Original » pour l’écouter en entier, ou « Contenu Fenêtre » pour écouter la fenêtre de signal affichée dans le *subplot* actif, ou enfin « Extrait ». Dans ce dernier cas, il faut sélectionner avec le bouton gauche de la souris, sur le *subplot* actif de la fenêtre principale, les deux instants entre lesquels on désire écouter le signal.

Matlab :

- Ecouter le signal en entier : Matlab >> acou\_ecout\_tout

- Ecouter la fenêtre de signal affichée : Matlab >> acou\_ecout\_fen

- Ecouter un extrait : Matlab >> acou\_ecout\_ext(instant1,instant2)

instant1,instant2 correspondant aux instants du signal en seconde

Il est également possible d’écouter en utilisant le bouton « Ecouter » de la barre d’outils de la fenêtre principale. Il faut d’abord s’assurer que le *subplot* qui contient le signal à écouter est bien actif (sinon l’activer en cliquant dessus), puis après avoir cliqué sur le bouton « Ecouter », 2 possibilités sont offertes. Soit on écoute toute la fenêtre de signal contenue dans ce *subplot* actif (cliquer avec le bouton droit de la souris), soit on écoute un extrait de signal contenu entre les 2 positions sélectionnées avec le bouton de gauche de la souris.

#### Exemple d’affichage de traitements effectués sur un fichier acoustique

La fenêtre ci-dessous (Figure 12) montre le résultat des traitements suivants à partir d’un fichier acoustique:

* Signal,
* Filtrage passe-haut par Remez,
* Spectrogramme + détection des formants,
* Intensité (évolution RMS).

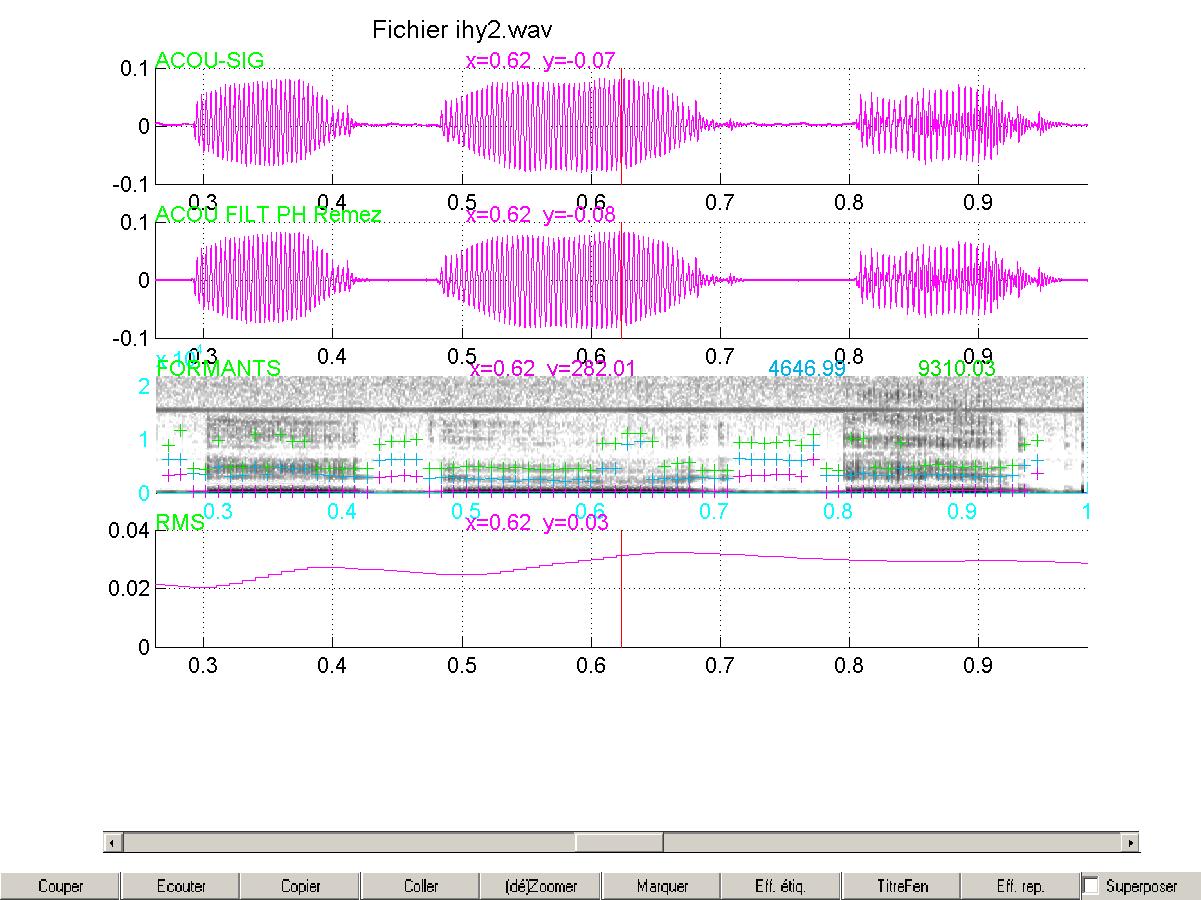


Figure  : exemple de traitements acoustiques effectués sous TRAP

### Traitements s’affichant dans la fenêtre synchrone

#### Calcul de la FFT et/ou de la LPC

Dans le menu « Traitement acoustique », choisir le sous-menu « FFT-LPC ». La boîte de dialogue suivante apparaîtra alors (Figure 13) :

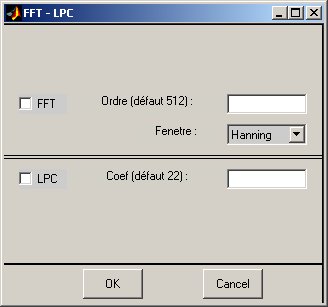


Figure  : fenêtre permettant de définir le type de spectre à visualiser

Si l’on désire juste visualiser la FFT, on coche la case puis on donne l’ordre de la FFT et le type de fenêtre temporelle. Si l’on veut la LPC, on coche la case et on donne le nombre de coefficients du filtre LPC. On peut aussi visualiser les deux en cochant les deux cases.

Matlab :

- pour la FFT :

Matlab >> ACOU\_ORD\_FFT = nombre\_echantillons

Matlab >> ACOU\_LONG\_FEN = longueur\_fenêtre

Matlab >> ACOU\_FEN\_TMP = Hanning(LONG\_FEN)

Matlab >> acou\_calc\_fft(pos\_curseur)

- pour la LPC :

Matlab >> ACOU\_ORD\_LPC = nombre\_coefficients

Matlab >> acou\_calc\_lpc(pos\_curseur)

#### Calcul du cepstre

Dans le menu « Traitement acoustique », choisir le sous-menu « cepstre ». La boîte de dialogue suivante apparaîtra alors (Figure 14) :

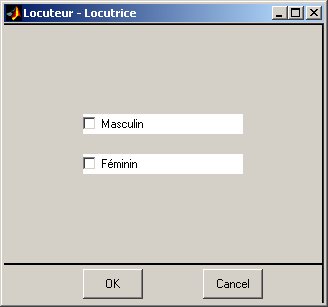


Figure  : choix du sexe du locuteur pour le calcul du cepstre

Après avoir coché la case correspondant au sexe du locuteur, le cepstre s’affichera dans la fenêtre synchrone.

Matlab >> ACOU\_CEPS\_VIS = taille\_fenetre\_de\_filtrage

Avec taille\_fenetre\_de filtrage = 4 pour un homme, 3.2 pour une femme.

Matlab >> acou\_calc\_cepstre(pos\_curseur)

#### Exemple d’affichage de traitements spectraux à partir d’un fichier acoustique

La Figure 15 montre les résultats de différents traitements spectraux avec le curseur de la fenêtre principale positionné à 1.46 secondes :

- FFT de 512 points avec une fenêtre de Hanning

- LPC avec 40 coefficients

- cepstre avec locuteur masculin.

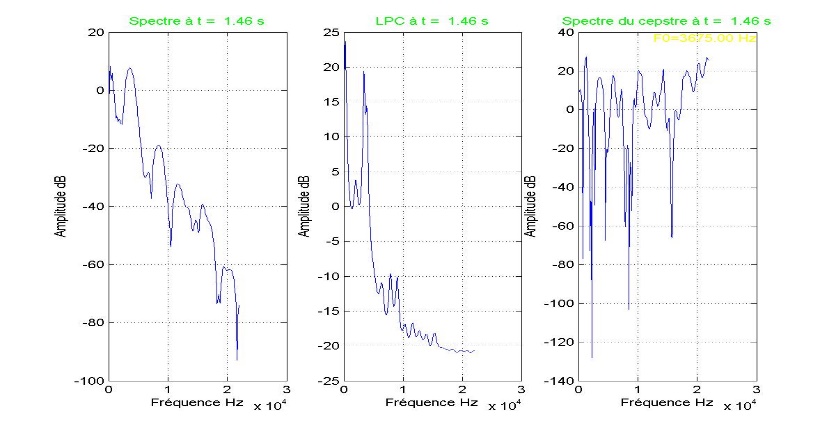


Figure  : exemples de traitements spectraux sous TRAP

En mode commande, il est aussi possible d’appeler simplement la fonction *curseur* pour réaliser ces traitements après avoir initialisé toutes les variables globales nécessaires à ces traitements et avoir mis les drapeaux à 1. On pourra ainsi reprendre le contrôle avec la souris en bougeant le curseur sans avoir besoin d’aller de nouveau sélectionner les traitements désirés dans le menu.

Matlab >> ACOU\_CALC\_FFT = 1

Matlab >> ACOU\_CALC\_FFT = 1

Matlab >> ACOU\_CALC\_CEPS = 1

Matlab >> curseur(pos\_curseur)

## Traitements sur les signaux EMA

### Trajectoire des bobines

Dans le menu « Traitement EMA » choisir le sous-menu « Trajectoire », la fenêtre suivante (Figure 16) apparaît à l’écran :



Figure  : fenêtre permettant de choisir la trajectoire de la bobine à visualiser

Cette fenêtre contient des boutons avec les noms des bobines définis précédemment (cf. II.1.1.2). Il suffit donc de cliquer sur un bouton pour afficher la trajectoire désirée (coordonnée X ou Y de la bobine) dans la fenêtre principale. On peut répéter cette action autant de fois qu’on le souhaite.

La première fois qu’on sélectionne le sous-menu « Traitement EMA / Trajectoire », un traitement est d’abord effectué sur les données articulatoires : les trajectoires (X ou Y) des bobines sont filtrées si CHEBY\_FREQ\_COUP ≠ 0.

Matlab >> ema\_traitement

(A appeler juste une fois avant tout traitement EMA).

Matlab >> ema\_traj(signal, nom\_signal)

Avec signal : de EMA\_X1 à EMA\_X16, de EMA\_Y1 à EMA\_Y16 ou de EMA\_Z1 à EMA\_Z16

### Vitesse

Dans le menu « Traitement EMA » choisir le sous-menu « Vitesse », une fenêtre du même type que la Figure 16 apparaît. Il suffit de cliquer sur le nom de la coordonnée de la bobine désirée pour que sa vitesse s’affiche dans la fenêtre principale.

Matlab >> ema\_vit(signal, nom\_signal)

Avec signal : de EMA\_X1 à EMA\_X16, de EMA\_Y1 à EMA\_Y16 ou de EMA\_Z1 à EMA\_Z16

**Remarque** : le calcul de la vitesse est fait avec un développement limité à l’ordre 2



Pour compatibilité avec l’ancienne version de TRAP, la fonction calc\_vit\_old.m est conservée : elle calcule la vitesse avec un développement dérivé à l’ordre 1 (moins précis et engendre un décalage d’1 échantillon) du type :



### Vitesse tangentielle

Dans le menu « Traitement EMA » choisir le sous-menu « Vitesse Tangentielle », une fenêtre de même type que la Figure 16 apparaît. Il suffit de cliquer sur le nom de la bobine désirée pour que sa vitesse tangentielle s’affiche dans la fenêtre principale.

Matlab >> ema\_vit\_tang(sig\_X, sig\_Y, sig\_Z, nom\_signal)

Avec sig\_X : de 1 à 16

### Accélération

Dans le menu « Traitement EMA » choisir le sous-menu « Accélération », une fenêtre de même type que la Figure 16 apparaît. Il suffit de cliquer sur le nom de la coordonnée de la bobine désirée pour que son accélération s’affiche dans la fenêtre principale.

Matlab >> ema\_acc(signal,nom\_signal)

Avec signal : de EMA\_X1 à EMA\_X16, de EMA\_Y1 à EMA\_Y16 ou de EMA\_Z1 à EMA\_Z16

### Détection des minima, maxima, passages par zéro (ppz) ou moyennes

Dans le menu « Traitement EMA » choisir le sous-menu « Repérage », puis sélectionner un des choix proposé dans la fenêtre de dialogue suivante (Figure 17) :

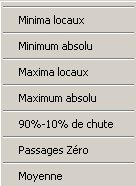


Figure  : fenêtre de dialogue pour le choix du repérage à effectuer

Il faut ensuite sélectionner avec la souris le signal sur lequel on désire faire la détection. Si on clique avec le bouton droit, le repérage se fera sur tout le signal. Si on clique avec le bouton gauche, il se fera entre les deux positions sélectionnées ensuite. Il s’affichera ensuite sur le signal, ou la partie de signal sélectionnée, des barres verticales à chaque emplacement d’un minimum, d’un maximum ou d’un passage par zéro, suivant le choix.

Pour les repérages de valeur moyenne, il s’affichera deux barres verticales aux instants de début et de fin du repérage, ainsi qu’une croix, positionnée entre ces deux barres, à la valeur moyenne du signal.

La fonction de repérage « 90%-10% de chute » est surtout utilisée sur le tracé d’aire aux lèvres et correspond aux valeurs pour la définition du MEM (Model of Expansion of Movement) :

* repérage de l’instant correspondant à 90% du maximum
* repérage des instants correspondant à une remontée de 10% du minimum de part et d’autre de celui-ci.

Ce menu est disponible pour les autres types de signaux (EMA, EVA ou EPG) mais dans ces cas il est à appliquer avec précaution.

Matlab >> minmaxppz(type\_repérage, type\_fichier, mode)

Avec :

- type\_repérage = 1 pour minima locaux,

2 pour maxima locaux,

3 pour ppz,

4 pour moy,

5 pour 90%-10% de chute,

6 pour minimum absolu

7 pour maximum absolu.

- type\_fichier à choisir parmi, ‘EPG’, ‘EMA’, ‘MES’ ou ‘EVA’.

- mode = ‘auto’ pour un repérage sur tout le signal tracé, ‘manuel’ pour un repérage sur la partie de signal située entre les 2 points ultérieurement marqués à la souris.

Le repérage des minima, maxima et passages par zéro se fait en recherchant les valeurs, du signal ou de sa dérivée, qui passent au-dessous d’un seuil donné. Ce seuil est réglable grâce à la variable globale REP\_SEUIL (initialisée à 10). Il faut l’ajuster en fonction de l’échelle du signal (la diminuer pour les grandes échelles et l’augmenter pour les petites échelles).

Matlab >> REP\_SEUIL = nouvelle\_valeur

Les repères de même type ont la même couleur. Initialement, les 4 couleurs de repère sont fixées à 4 couleurs différentes (vert, orange, jaune et cyan). Il est possible de modifier ces couleurs en modifiant les variables globales COULEUR\_MIN, COULEUR\_MAX, COULEUR\_ZERO et COULEUR\_MOY.

Matlab >> COULEUR\_MIN = [0.5 1 0.5]; %sort of green

Matlab >> COULEUR\_MAX = [1 0.6 0.2]; %sort of orange

Matlab >> COULEUR\_ZERO = [1 1 0]; %sort of yellow

Matlab >> COULEUR\_MOY = [0 1 1]; %sort of cyan

L’utilisateur a la possibilité ensuite d’enregistrer la position de ces repérages dans un fichier. Pour cela, il faut aller dans le menu « Fichier / Enregistrer» et sélectionner « Min – Max – PPZ - Moy ». Ensuite, il faut entrer dans la boîte de dialogue le nom du fichier de sauvegarde de ces repères.

Matlab >> enr\_minmaxppz(nom\_fichier)

Les outils associés à ces repérages sont également décrits en II.10.1.

### Exemple de traitements sur un fichier EMA

La fenêtre ci-dessous (Figure 18) montre le résultat des différents traitements possibles à appliquer sur des signaux EMA :

* trajectoire en X de la bobine TNG-DOS,
* vitesse en X de la bobine TNG-DOS et repérage du minimum absolu sur ce signal
* vitesse tangentielle de la bobine TNG-DOS et repérage des maxima locaux sur ce signal,
* accélération de la trajectoire en X de la bobine TNG-DOS,
* signal acoustique associé.

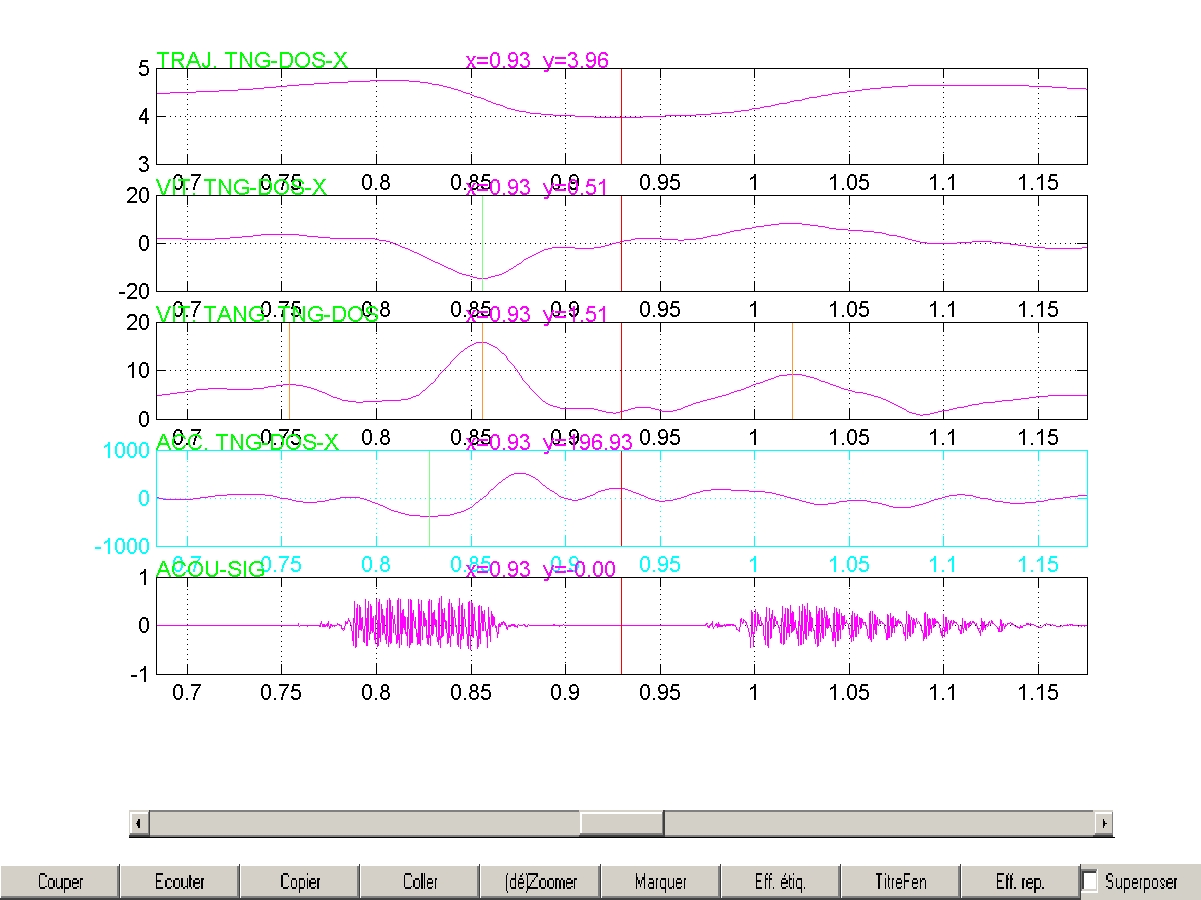


Figure  : exemple de traitements sur des signaux EMA sous TRAP

### Coupe sagittale / Fichier de contour du palais

Avant de visualiser la position des bobines dans le plan sagittal, il est préférable de créer un fichier de contour du palais du locuteur.

#### Extraction du fichier de contour du palais

Dans un premier temps, ouvrir le fichier (.mat) contenant le contour du palais en 2D ou 3D. Lancer la commande Matlab

Matlab >> ema\_traitement (ou ema\_traitement\_3D)

Tracer la trajectoire en X, Y (et Z) de la bobine utilisée pour s’assurer du bon enregistrement de celui-ci.

Définir ensuite un nom pour le fichier de contour du palais final : pour cela, dans le menu « Traitement EMA » choisir le sous-menu « Coupe sagittale », la boîte de dialogue suivante apparaît (Figure 19) :

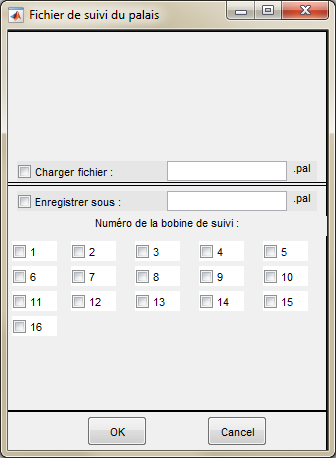


Figure  : fenêtre permettant de choisir un fichier de contour du palais

Cocher la case « Enregistrer sous » et rentrer alors un nom de fichier puis sélectionner la bobine avec laquelle a été effectué le contour. Cliquer sur « OK ». Ceci a pour effet d’ouvrir une fenêtre supplémentaire (« Fenêtre synchrone ») où s’affiche le contour palatal correspondant à toutes les positions de cette bobine au cours du temps.

Il est ensuite nécessaire de zoomer sur la partie utile de ce signal qui sera sauvegardée (pour ne pas tenir compte des mouvements inutiles du doigt qui a fait le suivi du palais). Pour cela on utilisera le bouton « zoomer » de la barre d’outils (cf. II.10.3.2). Une nouvelle fenêtre s’ouvre alors dans laquelle est tracée la partie du contour du palais sélectionnée. Si celle-ci est correcte, cliquer sur le bouton « Enregistrer » de cette fenêtre supplémentaire pour sauvegarder le palais dans le fichier rentré à l’aide de la fenêtre ci-dessus (Figure 19). Le contour du palais est alors sauvegardé dans un fichier.

Matlab >> EMA\_FIC\_PAL = nom\_fichier

Matlab >> EMA\_BOB\_PAL = numero\_bobine (avec laquelle l’utilisateur a fait le suivi du contour du palais ; une fenêtre supplémentaire s’ouvrira avec le contour palatal correspondant à cette bobine)

Matlab >> EMA\_COUP\_SAG = 1

Matlab >> zoom\_in(instant1, instant2)

Matlab >> ema\_enr\_pal (équivalent du bouton « Enregistrer » de la fenêtre supplémentaire)

#### Coupe sagittale

Pour visualiser les trajectoires de l’ensemble des bobines dans le plan sagittal : à partir du menu « Traitement EMA » choisir le sous-menu « Coupe sagittale », une boîte de dialogue identique à celle de la Figure 19 apparaît.

Dans cette fenêtre cocher la case « Charger fichier » et rentrer le nom du fichier *.pal* contenant le contour du palais précédemment sauvegardé (cf. II.4.7.1). Ce contour ainsi que la position de chaque bobine à l’instant courant (déterminé par le curseur de temps dans la fenêtre principale) s’afficheront dans la fenêtre synchrone.

Matlab >> EMA\_BOB\_PAL = 0

Matlab >> ema\_ouv\_pal(nom\_fichier)

puis

Matlab >> ema\_coupe\_sagit(pos\_curseur)

ou

Matlab >> EMA\_COUP\_SAG =1

Matlab >> curseur(pos\_curseur)

Ou si 3D

Matlab >> EMA\_COUP\_SAG\_3D =1

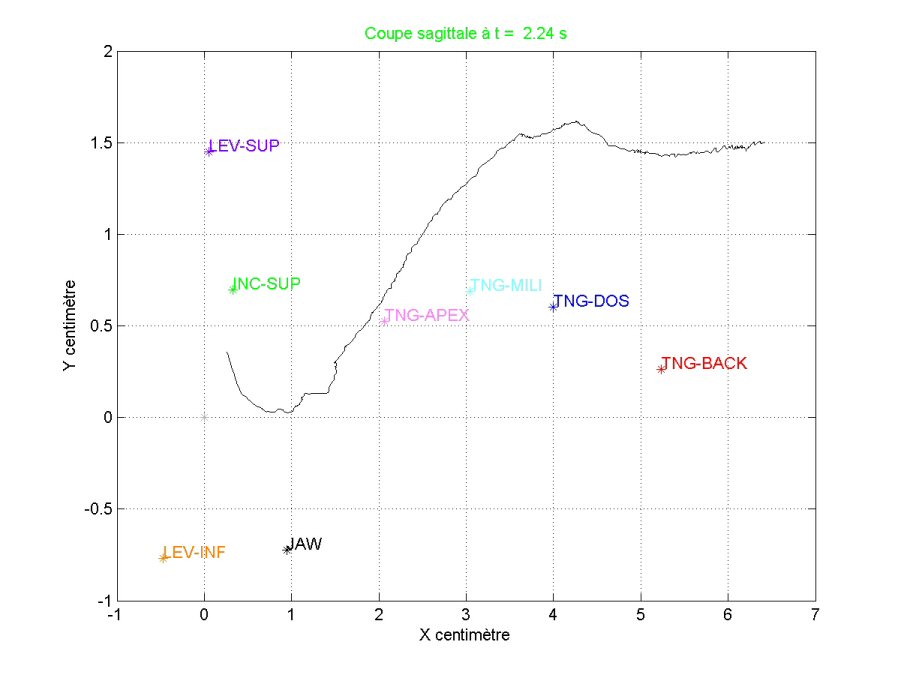


Figure  : exemple de coupe sagittale

### Animation des trajectoires des bobines

Dans le menu « Traitement EMA » choisir le sous-menu « Animation ». Il faut ensuite cliquer sur un graphique de la fenêtre principale. Si l’on désire obtenir l’animation intégrale on clique avec le bouton droit de la souris. Sinon on choisit, avec le bouton gauche de la souris, les deux positions entre lesquelles on désire voir l’animation.

Matlab >> ema\_animer(position1,position2)

Ou si 3D

Matlab >> ema\_animer\_3D(position1,position2)

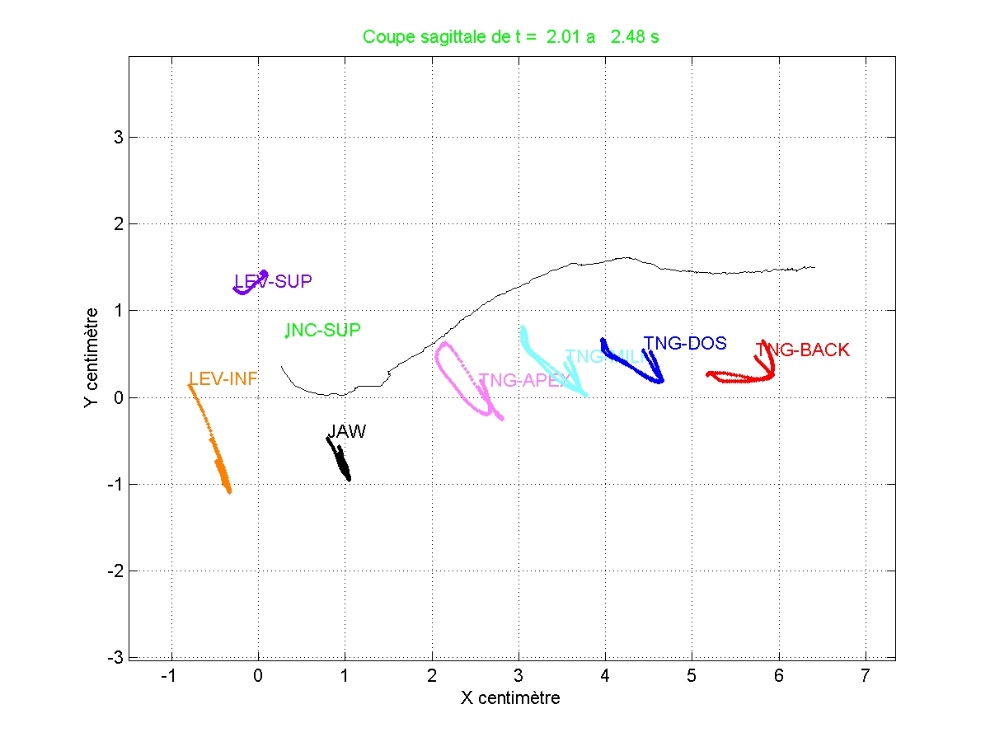


Figure  : exemple d’animation de 8 bobines entre 2.01 et 2.48 secondes

### Affichage des tilts des bobines

Il est possible de vérifier le facteur de tilt des bobines, pour estimer la confiance avec laquelle on peut considérer les données correspondant à une bobine particulière.

Pour cela, il faut sélectionner le sous-menu « Facteurs Tilt » du menu « Traitement EMA ». Une fenêtre supplémentaire apparaîtra, avec les tilts en fonction du temps pour les 10 bobines.

Matlab >> ema\_calc\_tilt

**Remarque** : Un tilt correct doit être compris entre 0,9 et 1.

## Traitements sur les signaux EPG

Lorsque l’on analyse des données EPG, il peut être utile de sélectionner des régions (dentale, alvéolaire, vélaire, latérale, etc.) dans lesquelles on pourra calculer ultérieurement des pourcentages de contact. TRAP permet de définir de telles régions et d’enregistrer leur spécification dans le fichier associé au locuteur (.usr).

### Sélection des électrodes d’une région

Dans le menu « Traitement EPG », choisir le sous-menu « Sélection des régions », la boîte de dialogue suivante (Figure 22) apparaît à l’écran.



Figure  : fenêtre de sélection d’une région

Après avoir donné le numéro de la région et son nom, la fenêtre permettant de sélectionner les électrodes de cette région s’ouvre (Figure 23).

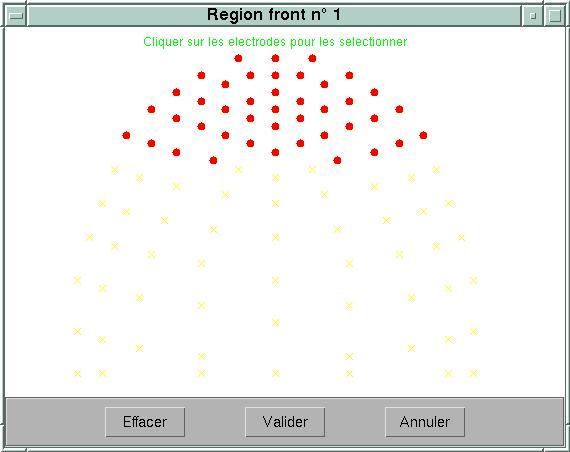


Figure  : fenêtre de sélection de la région n° 1 appelée « front »

Pour sélectionner une électrode, il suffit de cliquer dessus avec la souris. Il est possible de désélectionner toutes les électrodes sélectionnées en cliquant sur « Effacer ». Lorsque la sélection est terminée, il faut cliquer sur « Valider » afin que les modification soient prises en compte.

Matlab >> epg\_select\_reg([num\_elect1 … num\_electN], numero\_region, nom\_region)

### Sauvegarde de ces régions dans le fichier .usr

Afin que les modifications des régions soient enregistrées dans le fichier *.usr*, il faut aller dans le menu « Traitement EPG » et choisir « Sauvegarder régions ».

Matlab >> epg\_sauve\_reg

### Calcul du taux de constriction dans une région

Sélectionner le sous-menu « Traitement EPG/Taux constriction », et après avoir donné le numéro de la région sur laquelle on désire calculer ce taux, il s’affiche dans la fenêtre principale.

Matlab >> epg\_taux\_cons(numero\_region)

### Chargement des fichiers de résultat de traitement sur données EPG

Les traitements à effectuer sur des données EPG pour obtenir des degrés de constriction, des centres de gravité, des degrés de latéralité, etc., étant innombrables, nous avons choisi de laisser l’utilisateur effectuer les traitements à part et de lui offrir la possibilité de charger des fichiers contenant les résultats de ces traitements. Pour simplifier, nous faisons référence à ces traitements par le terme « constriction ».

Dans le menu « Traitement EPG / Trajectoire », sélectionner « Charger fichier », puis rentrer l’extension du fichier (la racine du nom du fichier est la même que celle du fichier EPG chargé) et le nom à affecter au signal. Celui-ci s’affichera dans la fenêtre principale.

Lorsqu’un ou plusieurs fichiers ont été chargés, il est possible de réafficher la trajectoire en sélectionnant « Trajectoires EPG chargées », mais on peut aussi afficher la vitesse et l’accélération de ces signaux en sélectionnant « Vitesse » ou « Accélération ».

Si par exemple on a chargé trois fichiers et que l’on demande les trajectoires chargées, la fenêtre suivante apparaît et il suffit de cliquer sur le nom d’un signal (donné par l’utilisateur lors du chargement du fichier) pour qu’il s’affiche.

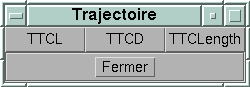


Figure  : fenêtre permettant de choisir le fichier EPG à visualiser

Matlab :

* chargement d’un fichier :

Matlab >> epg\_ouv\_cons(extension\_fichier, nom\_signal)

* affichage d’une trajectoire chargée :

Matlab >> trace\_graphe(EPG\_DATA\_CONS(num\_chargt, :), EPG\_TPS, nom\_signal, ‘-‘, [1 1 0])

*num\_chargt* correspond au numéro du fichier chargé. Le premier fichier chargé a le numéro 1, et le dernier a le numéro contenu dans la variable globale EPG\_NB\_CONS qui sert de compteur.

* affichage de la vitesse d’un signal chargé :

Matlab >> epg\_vit(EPG\_DATA\_CONS (num\_chargt, :), nom\_signal)

* affichage de l’accélération d’un signal chargé :

Matlab >> epg\_acc(EPG\_DATA\_CONS (num\_chargt, :), nom\_signal)

Pour les formules utilisées pour les calculs de vitesse et d’accélération, se reporter au §25II.4.2 et §II.4.4.

### Repérage des minima, maxima, passages par zéro ou moyennes

Sélectionner « Minima », « Maxima », « Passages Zéro » ou « Moyenne » dans le menu « Traitement EPG / Repérage ». Après c’est le même principe que pour le repérage des min - max – ppz – moy des signaux EMA (cf. II.4.5).

Cette fonction détecte aussi les plateaux en affichant une ligne verticale sur le premier échantillon d’un plateau.

### Exemple de traitements EPG

La fenêtre ci-dessous (Figure 25) montre le résultat des traitements suivants :

- Taux de constriction de la région numéro 1, plus détection des minima.

- Trajectoire du signal contenu dans le fichier appelé *TTCL*, plus détection des maxima.

- Vitesse du signal *TTCL*.

- Accélération du signal *TTCL*.

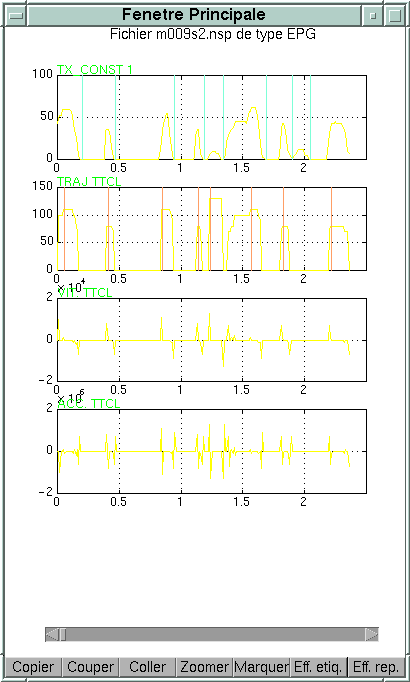


Figure  : exemple de traitements possibles sur des signaux EPG avec TRAP

### Visualisation des électrodes contactées du palais à un instant donné

Sélectionner « Simple » du menu « Traitement EPG / Palais », puis après avoir modifié la position du curseur temporel de la fenêtre principale, les emplacements de toutes les électrodes du palais s’afficheront dans la fenêtre synchrone en jaune, les électrodes contactées étant marquées par une astérisque rouge.

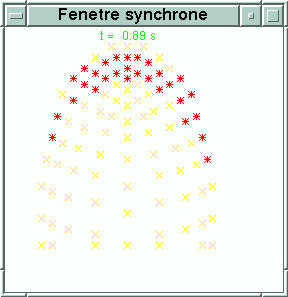


Figure  : électrodes contactés du fichier m009s2.nsp à t = 0.89s

Matlab >> epg\_simple(instant)

ou

Matlab >> EPG\_PAL\_SIMPLE = 1

Matlab >> curseur(instant)

### Visualisation des électrodes contactées du palais sur plusieurs échantillons

Sélectionner « Série » du menu « Traitement EPG / Palais », et donner le nombre de palais à visualiser avant et après l’instant courant. Une nouvelle fenêtre s’ouvre, puis après avoir modifié la position du curseur de la fenêtre principale, les palais s’affichent dans cette fenêtre.

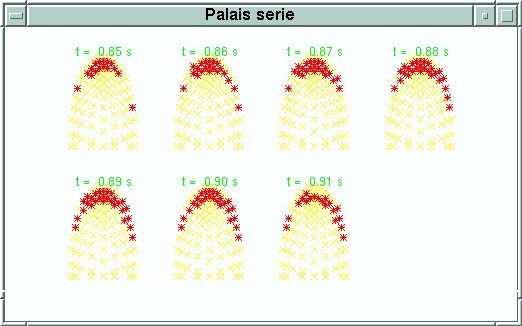


Figure  : série de palais pour 3 échantillons avant et après l’instant t = 0.88s

Matlab >> EPG\_NB\_PAL = nombre\_echantillons

Matlab >> epg\_serie(instant)

ou

Matlab >> EPG\_PAL\_SERIE = 1

Matlab >> curseur(instant)

## Traitements sur les signaux TACLE

### Trajectoire

Dans le menu « Traitement TACLE/Mesure », choisir le sous-menu « Trajectoire », ceci a pour conséquence d’ouvrir une fenêtre avec autant de boutons que de paramètres retenus. La Figure 28 montre le cas ou 4 paramètres (A, B, S, P1) ont été retenus. Il suffit donc de cliquer sur un bouton pour afficher la trajectoire désirée dans la fenêtre principale. On peut répéter cette action autant de fois qu’on le souhaite.



Figure  : fenêtre permettant de choisir le paramètre à visualiser

Matlab >>trace\_graphe(MES\_DATA(numéro\_mesure, :),MES\_TPS,nom\_mesure,'-', COULEUR\_GRAPH);

### Vitesse

Dans le menu « Traitement TACLE/Mesure », choisir le sous-menu « Vitesse ». Une fenêtre du même type que celle Figure 28 apparaît. Il suffit de cliquer sur un bouton pour afficher la vitesse désirée dans la fenêtre principale.

Matlab >> mes\_vit(signal, nom\_signal)

Pour la formule utilisées pour le calculs de la vitesse cf. § 25II.4.2.

### Accélération

Dans le menu « Traitement TACLE/Mesure », choisir le sous-menu « Accélération ». Une fenêtre du même type que celle Figure 28 apparaît. Il suffit de cliquer sur un bouton pour afficher l’accélération désirée dans la fenêtre principale.

Matlab >> mes\_acc(signal, nom\_signal)

Pour la formule utilisée pour le calculs de l’accélération cf. § II.4.4.

### Filtrage

Il est possible à tout moment de filtrer les données labiales à l’aide du filtre passe-bas par défaut (Chebyshev). Pour cela, il est nécessaire dans un premier temps de définir la fréquence de coupure du filtre (cf. II.1.1.5). Ensuite, dans le menu « Traitement TACLE/Mesure », choisir le sous-menu « Filtrage ». Une fenêtre du même type que celle Figure 28 apparaît. Il suffit de cliquer sur un bouton pour afficher le signal filtré dans la fenêtre principale.

Matlab >> [sig\_filt]=filtre\_cheby(signal,fech, CHEBY\_FREQ\_COUP)

Pour appliquer un autre type de filtre, se reporter au § II.10.2.5.

### Repérage des minima, maxima, passages par zéro ou moyennes

Dans le menu « Traitement TACLE/Mesure », choisir le sous-menu « Repérage » puis sélectionner une des options parmi « Minima locaux », « Minimum absolu », « Maxima locaux », « Maximum absolu », « 90%-10% de chute », « Passages par Zéro » ou « Moyenne ». Après c’est le même principe que pour le repérage des signaux EMA (cf. II.4.5).

### Exemple de traitements sur un fichier TACLE

La fenêtre ci-dessous (Figure 29) montre le résultat des différents traitements possibles à appliquer sur des signaux issus de TACLE :

* signal acoustique,
* trajectoire de l’étirement (A) et repérage du minimum absolu,
* trajectoire de l’ouverture (B) et repérage des maxima locaux,
* trajectoire de l’aire aux lèvres (S) et repérage des instants : 90% du maximum et 10% du minimum à droite et à gauche de celui-ci,
* trajectoire (en mode point) de la protrusion de la lèvre supérieure,
* vitesse de P1 filtrée par Chebyshev (CHEBY\_FREQ\_COUP = 7 Hz)

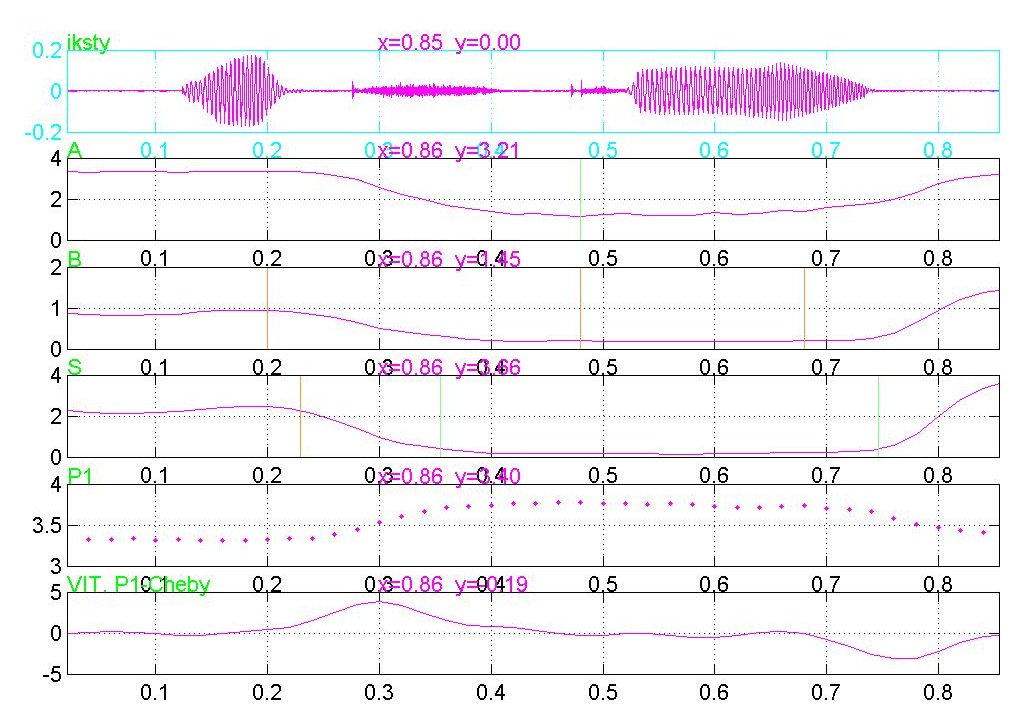


Figure  : exemple de traitements sur les signaux TACLE sous TRAP

## Traitements sur les signaux de mesures

Comme il a été précisé au § II.1.2.4, un fichier de mesure est un fichier ASCII pouvant contenir plusieurs mesures articulatoires, en colonne. L’ouverture du fichier et le chargement d’une mesure de ce fichier se fait par le menu « Fichier/ Ouvrir/ Fichier mesures (ASCII) ». On peut ensuite effectuer un certain nombre de traitements sur la mesure chargée de la même manière que précédemment avec les signaux issus de TACLE.

### Trajectoire

Lorsqu’un fichier de mesures a été chargé, il est possible d’afficher la trajectoire d’une des mesures en sélectionnant le sous-menu « Trajectoire» du menu « Traitement TACLE/Mesure ».

On voit alors apparaître une fenêtre comportant des boutons avec le nom des différentes mesures contenues dans le fichier. Il suffit de cliquer sur le nom de la mesure choisie (donné par l’utilisateur via le menu « Options/Mesure ») pour que sa trajectoire temporelle s’affiche dans la fenêtre principale.

Matlab >> trace\_graphe(MES\_DATA(numéro\_mesure, :), MES\_TPS, nom\_mesure, ‘-‘, [1 1 0])

### Vitesse

On peut aussi afficher la vitesse d’une mesure. Dans le menu « Traitement Mesure » choisir le sous-menu « Vitesse », puis sélectionner la mesure souhaitée et la vitesse de cette mesure s’affichera dans la fenêtre principale.

Matlab >> mes\_vit(MES\_DATA(numéro\_mesure, :), nom\_mesure)

### Accélération

Dans le menu « Traitement Mesure » choisir le sous-menu « Accélération », puis sélectionner la mesure souhaitée, son accélération s’affichera alors dans la fenêtre principale.

Matlab >> mes\_acc(MES\_DATA(numéro\_mesure, :), nom\_ mesure)

### Repérage des minima, maxima, passages par zéro ou moyennes

Dans le menu « Traitement TACLE/Mesure », choisir le sous-menu « Repérage » puis sélectionner une des options parmi « Minima locaux », « Minimum absolu », « Maxima locaux », « Maximum absolu », « 90%-10% de chute », « Passages par Zéro » ou « Moyenne ». Après c’est le même principe que pour le repérage des signaux EMA (cf. II.4.5).

## Traitements sur les signaux EVA

### Trajectoire

Dans le menu « Traitement EVA », choisir le sous-menu « Trajectoire », ceci a pour conséquence d’ouvrir une fenêtre avec autant de boutons que de type de fichiers ouverts. La Figure 30 montre le cas ou 5 fichiers (WA1, F0, OAF, NAF et PR1) ont été ouverts. Il suffit donc de cliquer sur un bouton pour afficher la trajectoire désirée dans la fenêtre principale. On peut répéter cette action autant de fois qu’on le souhaite.



Figure  : fenêtre permettant de choisir la trajectoire du signal à visualiser

Matlab >> eva\_traj(signal, nom\_EVA);

Où signal est à choisir parmi : EVA\_WA1, EVA\_WA2, EVA\_F0, EVA\_EGG, EVA\_OAF, EVA\_NAF, EVA\_PR1 et EVA\_PR2.

Et où nom\_EVA contient **obligatoirement** un des champs suivants : WA1, WA2, F0, EGG, OAF, NAF, PR1 et PR2. Par exemple nom\_EVA = ‘OAF-Filt’

**NOTE** : le signal de type EVA\_WA1 est considéré comme le signal acoustique de référence sur lequel seront appliqués les différents traitements acoustiques ultérieurs (sonagramme, formants, F0, etc…)

### Vitesse

Dans le menu « Traitement EVA », choisir le sous-menu « Vitesse ». Une fenêtre du même type que celle Figure 30 apparaît. Il suffit de cliquer sur un bouton pour afficher la vitesse désirée dans la fenêtre principale.

Matlab >> eva\_vit(type\_EVA, nom\_EVA)

### Accélération

Dans le menu « Traitement EVA », choisir le sous-menu « Accélération ». Une fenêtre du même type que celle Figure 30 apparaît. Il suffit de cliquer sur un bouton pour afficher l’accélération désirée dans la fenêtre principale.

Matlab >> eva\_acc(type\_EVA, nom\_EVA)

### Filtrage

Il est possible à tout moment de filtrer les données EVA à l’aide du filtre passe-bas par défaut (Chebyshev). Pour cela, il est nécessaire dans un premier temps de définir la fréquence de coupure du filtre (cf. II.1.1.5). Ensuite, dans le menu « Traitement EVA », choisir le sous-menu « Filtrage ». Une fenêtre du même type que celle Figure 30 apparaît. Il suffit de cliquer sur un bouton pour afficher le signal filtré dans la fenêtre principale.

Matlab >>[sig\_filt]=filtre\_cheby(type\_EVA, fech\_eva, CHEBY\_FREQ\_COUP)

Pour appliquer un autre type de filtre, se reporter au § II.10.2.5.

### Repérage des minima, maxima, passages par zéro ou moyennes

Dans le menu « Traitement EVA », choisir le sous-menu « Repérage » puis sélectionner une des options parmi « Minima locaux », « Minimum absolu », « Maxima locaux », « Maximum absolu », « 90%-10% de chute », « Passages par Zéro » ou « Moyenne ». Ensuite c’est le même principe que pour le repérage des signaux EMA (cf. II.4.5).

### Exemple de traitements sur des fichiers EVA

La fenêtre ci-dessous (Figure 31) montre le résultat des différents traitements possibles à appliquer sur des signaux issus de EVA :

* signal acoustique,
* trajectoire de F0,
* vitesse de F0 filtrée par Chebyshev,
* trajectoire de OAF,
* trajectoire de NAF,
* trajectoire de PR1.

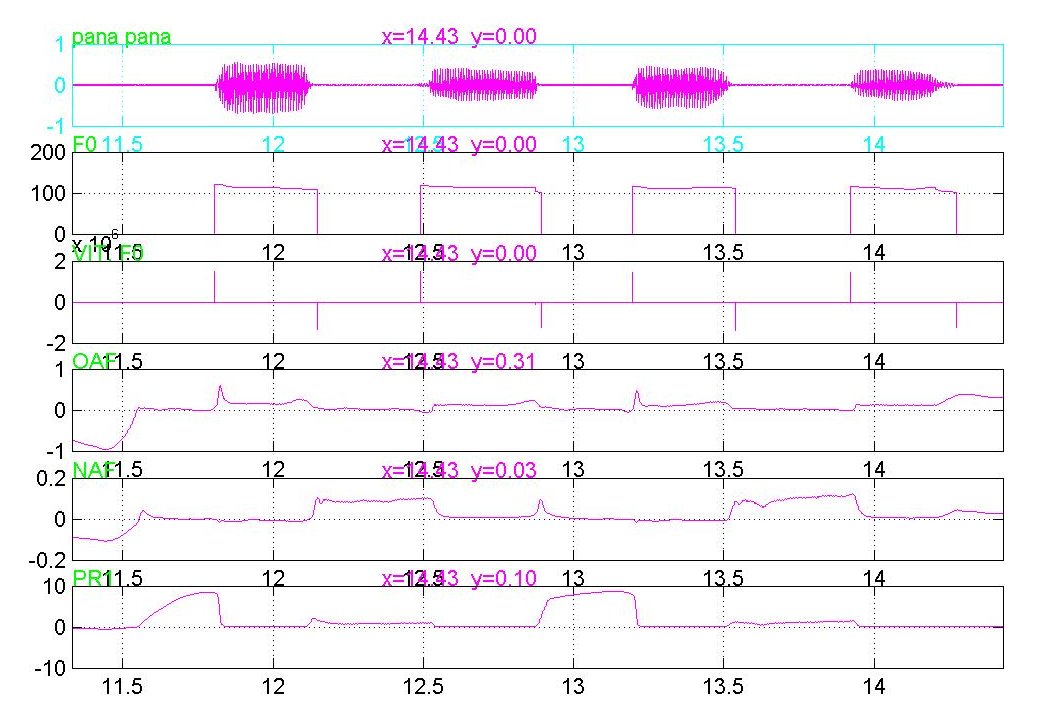


Figure  : exemple de traitements sur les signaux EVA sous TRAP

## Traitements sur les signaux EGG

Les signaux EGG sont les signaux enregistrés par l’electro-glottographe et numérisés via la carte d’acquisition DT3016. Le nombre de canaux peut varier de 2 à 4.

### Trajectoire

Dans le menu « Traitement EGG », choisir le sous-menu « Trajectoire », ceci a pour conséquence d’ouvrir une fenêtre avec autant de boutons que de canaux numérisés. La montre le cas où 4 canaux EGG ont été enregistrés. Il suffit donc de cliquer sur un des boutons pour afficher la trajectoire désirée dans la fenêtre principale. On peut répéter cette action autant de fois qu’on le souhaite.



Figure  : fenêtre permettant de choisir la trajectoire du signal EGG à visualiser

Matlab >> egg\_traj(signal, nom\_EGG);

Où signal est à choisir parmi : EGG\_1, EGG\_2, EGG\_3 ou EGG\_4.

### Vitesse

Dans le menu « Traitement EGG », choisir le sous-menu « Vitesse ». Une fenêtre du même type que celle apparaît. Il suffit de cliquer sur un bouton pour afficher la vitesse désirée dans la fenêtre principale.

Matlab >> egg\_vit(signal, nom\_EGG)

### Accélération

Dans le menu « Traitement EGG », choisir le sous-menu « Accélération ». Une fenêtre du même type que celle apparaît. Il suffit de cliquer sur un bouton pour afficher l’accélération désirée dans la fenêtre principale.

Matlab >> egg\_acc(signal, nom\_EGG)

### Filtrage

Il est possible à tout moment de filtrer les données EGG à l’aide des filtres proposés dans TRAP. Pour cela, dans le menu « Traitement EGG », choisir le sous-menu « Filtrage ». Une fenêtre du même type que celle apparaît. Après avoir sélectionné le canal à filtrer, une fenêtre identique à la Figure 33 apparaît. Il suffit alors de cocher le type de filtre à appliquer et, le cas échéant, de définir les paramètres du filtrage (Fréquence de coupure ou Nombre d’interpolations entre 2 échantillons).



Figure : fenêtre permettant de choisir le type de filtre à appliquer

Pour choisir le filtrage à appliquer sur les signaux EGG en mode script, il faut taper l’une des 3 commandes suivantes :

Matlab >> EGG\_FILT = filtre\_cheby(signal\_EGG, EGG\_FECH, CHEBY\_FREQ\_COUP);

Matlab >> [EGG\_FILT, donnees\_temporelles] = filtre\_spline(signal\_EGG, EGG\_FECH, nb\_interp);

Matlab >> EGG\_FILT = filtre\_remez(EGG, EGG\_FECH);

### Repérage des minima, maxima, passages par zéro ou moyennes

Dans le menu « Traitement EGG », choisir le sous-menu « Repérage » puis sélectionner une des options parmi « Minima locaux », « Minimum absolu », « Maxima locaux », « Maximum absolu », « 90%-10% de chute », « Passages par Zéro » ou « Moyenne ». Ensuite c’est le même principe que pour le repérage des signaux EMA (cf. II.4.5).

### Exemple de traitements sur des fichiers EGG

La fenêtre ci-dessous (Figure 34) montre le résultat des différents traitements possibles à appliquer sur des signaux EGG :

* trajectoire de EGG\_1,
* trajectoire de EGG\_1 filtrée par Remez,
* trajectoire de EGG\_2,
* vitesse de EGG\_3,
* trajectoire de EGG\_4.

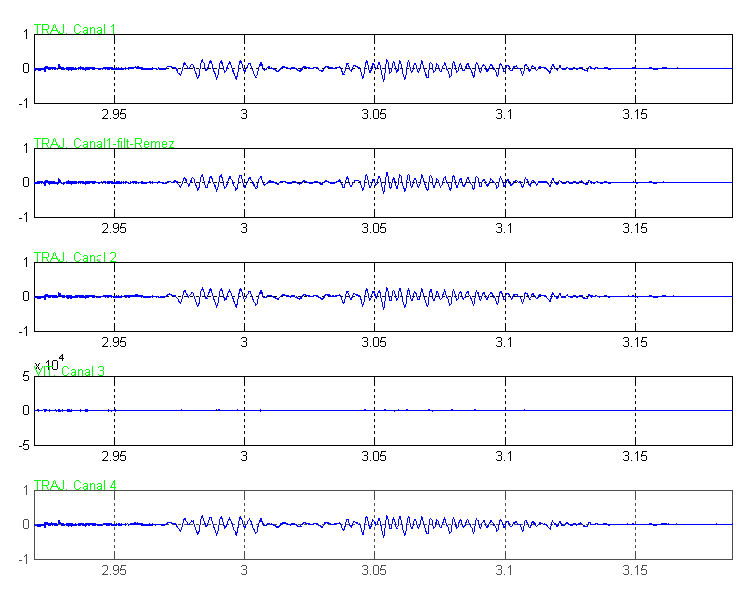


Figure : exemple de traitements sur les signaux EGG sous TRAP

## Outils divers

La plupart de ces outils sont disponibles sur la barre d’outils de la fenêtre principale cf. Figure 2 page 6.

### Etiquetage des signaux

#### Etiqueter les signaux

Cliquer sur le bouton « Marquer » de la barre d’outils de la fenêtre principale, la boîte de dialogue suivante apparaît (Figure 35) :



Figure  : fenêtre permettant de définir le nom et le type d’étiquette à poser

Après avoir donné le nom et spécifié le type de l’étiquette, il faut sélectionner avec la souris la position de cette étiquette. Une ligne verticale s’affichera sur tous les graphiques de la fenêtre principale à la position sélectionnée.

Matlab >> marquer(instant, nom\_etiquette, niveau\_etiquette)

Avec :

- *instant* : instant sur l’axe des abscisses (en s) de l’étiquette.

- *niveau\_etiquette* au choix de 1 à NIV\_ETIQ.

Initialement, on a le choix entre 5 types d’étiquettes : Phonétique, Tonal, Break Index, Mismatch, Commentaire. On peut modifier le nom de ces types (et en ajouter) en modifiant la variable globale TYPE\_ETIQ.

Matlab >> TYPE\_ETIQ(1, 1:length(’phrase’)) = ’phrase’

Matlab >> TYPE\_ETIQ(2, 1:length(’mots’)) = ’mots’

Matlab >> TYPE\_ETIQ(3, 1:length(’syll’)) = ’syll’

Le bas de la fenêtre contient une portée invisible contenant NIV\_ETIQ niveaux. La variable globale NIV\_ETIQ, qui contient donc le nombre total de niveaux (soit aussi de types) d’étiquettes, est initialisée à 5. Cinq niveaux d’étiquettes (et cinq types) sont donc possibles initialement. On peut bien sûr augmenter ou diminuer ce nombre total de niveaux (par exemple avec la commande :

Matlab >> NIV\_ETIQ = 10

Le type d’étiquette sélectionné impose donc le niveau sur lequel le nom de l’étiquette sera placé dans la portée.

Les étiquettes de même type apparaîtront donc sur la même ligne de la portée. Elles auront aussi la même couleur. Initialement, les 5 niveaux de couleurs d’étiquettes sont fixés à 5 nuances de vert. Il est possible de modifier ces niveaux de couleurs en modifiant la variable globale COULEUR\_ETIQ. Les commandes suivantes donnent des niveaux de couleur proches du cyan :

Matlab >> COULEUR\_ETIQ(1, :) = [0 1 0.9]

Matlab >> COULEUR\_ETIQ(2, :) = [0 1 0.8]

Matlab >> COULEUR\_ETIQ(3, :) = [0 1 0.7]

**Attention**, il est fortement recommandé de ne pas utiliser le vert (codé [0 1 0]) pour les couleurs des étiquettes. Cette couleur est réservée aux titres des graphiques et l’affichage des titres pourrait ne plus fonctionner.

Il est également possible de choisir le type de trait pour les étiquettes (‘:’, ‘-‘, ‘--‘) :

Matlab >> STYLE\_ETIQ(1,1:1)=’:’;

Matlab >> STYLE\_ETIQ(2,1:2)=’-’;

Matlab >> STYLE\_ETIQ(3,1:1)=’--’;

#### Sauvegarder les étiquettes

Il est possible de sauvegarder la position sur l’axe des abscisses, le nom et le type des étiquettes présentes dans la fenêtre principale. On sauvegarde également, pour chaque étiquette, la valeur sur l’axe des ordonnées de tous les graphes de **tous** les *subplots* de la fenêtre principale à l’instant déterminé par l’étiquette. Pour cela, il faut sélectionner dans le menu « Fichier » les sous-menus « Enregistrer » puis « Etiquettes » et donner le nom du fichier de sauvegarde.

Matlab >> enr\_etiq(nom\_fichier)

|  |
| --- |
| TRAP : Fichier d'etiquettes  signal : jls\_pata\_1.mat  type : EMA  comment : cree le 25-May-2004  color : 0.00 0.74 0.00  nfields : 8  0.000000 1 a -26.000000 0.000000  0.200000 2 b -22.000000 0.000000  0.400000 3 c -28.000000 0.000000  0.600000 4 d -6.000000 182.591000  0.800000 5 e 717.000000 0.000000  1.000000 6 f 201.000000 260.103000  1.200000 7 g 466.000000 0.000000  1.400000 8 h 795.000000 0.000000 |

Figure  : exemple d’un fichier d’étiquettes sauvegardées

#### Charger un fichier d’étiquettes

Il est tout à fait possible de charger un fichier de segmentation créé par le logiciel PRAAT (.TextGrid) ou ISIS. Le format sera reconnu automatiquement par TRAP et les étiquettes s’afficheront sur les signaux aux endroits spécifiés. Il est également possible d’afficher les étiquettes sauvegardées dans un fichier depuis TRAP..

Pour cela, il faut sélectionner dans le menu « Fichier », le sous-menu « Charger étiquettes » et donner le nom du fichier.

Matlab >> ouv\_etiq(nom\_fichier)

#### Effacer une étiquette

Cliquer sur le bouton « Eff. Etiq. » de la barre d’outils de la fenêtre principale, puis sélectionner avec la souris l’étiquette à effacer. Celle-ci s’effacera de tous les graphiques et son nom disparaîtra de la portée du bas de la fenêtre. Cet outil efface l’étiquette la plus proche possible de l’endroit cliqué ou spécifié.

Matlab >> eff\_etiq(instant)

#### Effacer un repère min-max-ppz

La description du repérage des minima, maxima et passages par zéro est donnée au paragraphe § II.4.5. Une fois les repérages effectués sur un signal, on peut vouloir effacer certains repères. Pour cela, cliquer sur le bouton « Eff rep. » de la barre d’outils de la fenêtre principale, puis sélectionner avec la souris le repère à effacer. Le repère le plus proche de l’endroit spécifié est effacé du signal et son nom disparaît de la portée du bas de la fenêtre.

Matlab >> eff\_minmaxppz(instant)

Où *instant* est une valeur temporelle (sur l’axe des abscisses) la plus proche possible du repère à effacer.

#### Sauvegarder les repères min-max-ppz-moy

La description du repérage des minima, maxima, passages par zéro et valeurs moyennes est donnée au paragraphe § II.4.5. Une fois les repérages effectués sur un signal, il est possible de sauvegarder la position et le type des repères présents sur un signal de la fenêtre principale. Pour cela il faut sélectionner « Min – Max – PPZ - Moy » dans le menu « Fichier / Enregistrer », et donner le nom du fichier de sauvegarde.

Matlab >> enr\_minmaxppz(nom\_fichier)

Le fichier obtenu (cf. Figure 37) contient (pour chaque repère de type min max ou ppz) dans la première colonne, la position dans le temps du repère, dans la deuxième colonne son type (min, max, moy ou ppz) dans la troisième colonne le nom de la bobine associée à la mesure et dans les colonnes suivantes, la valeur du signal à la position du repère pour toutes les fenêtres ouvertes.

Pour les repères de moyenne, les 2 premières colonnes correspondent aux instants de début et de fin, entre lesquels le repérage de la moyenne a été fait.

Il n’existe pas de fonctions permettant de charger un fichier de repères min-max-ppz-moy mais s’il le désire, l’utilisateur peut le charger en tant que fichier d’étiquettes. Dans ce cas, les repères s’afficheront sur tous les signaux.

|  |
| --- |
| TRAP : Fichier de repères min, max, ppz ou moy  signal : J:\bedei\EMA\WAVE\Labiglo\FR\JM\_160429\wav\JM\_kiss\_004\_sync.wav  comment : cree le 03-Mar-2017  46.695000 min TRAJ. DORS-Y -0.005829 5.683185  46.575000 min TRAJ. TIP-Y 0.451725 5.049613  46.495000 max TRAJ. LLIP-Y 0.676142 4.697318  9.035000 max TRAJ. JAW-X -0.163475 5.677570  46.425000 max TRAJ. LRX-Y 0.454815 4.803329  11.560000 ppz TRAJ. DORS-Y 1999.0000  2.00200 moy TRAJ. LLIP-Y -323.0000 |

Figure  : exemple de fichier de repères

### Fonctions du menu déroulant d’un *subplot*

Lorsque l’on clique sur un subplot, un menu déroulant identique à celui ci-dessous apparaît :



Figure  : menu déroulant d’un subplot

Les fonctions accessibles via ce menu sont expliquées ci-dessous.

#### Changer l’échelle d’un *subplot*

Sélectionner « Echelle » et la boîte de dialogue suivante (Figure 39) s’ouvre :

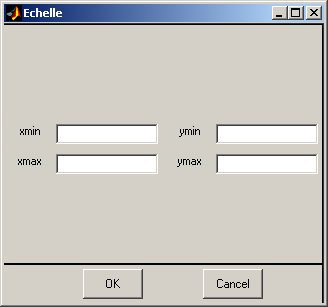


Figure  : fenêtre permettant de modifier l’échelle du subplot courant

On donne la nouvelle échelle en remplissant tous les champs dans l’ordre en faisant attention à donner une valeur à *xmax* supérieure à *xmin,* et de même pour *ymax* et *ymin*.

Matlab >> modif\_echelle(xmin,xmax,ymin,ymax)

#### Nommer les axes d’un *subplot*

Sélectionner « Nom » du menu déroulant et rentrer le nom de l’axe des *x* et celui de l’axe des *y* dans la boîte de dialogue.

Matlab >> xlabel(nom\_x)

Matlab >> ylabel(nom\_y)

#### Renommer un *subplot*

Sélectionner « Titre » du menu déroulant et rentrer le nouveau titre du *subplot*. Celui-ci sera affiché dans le coin haut et gauche du subplot courant.

Matlab >> titre(titre\_subplot)

#### Changer le type du trait d’un *subplot*

Sélectionner « Trait » du menu déroulant et choisir le nouveau trait avec lequel sera tracé le prochain signal parmi les choix ci-dessous :

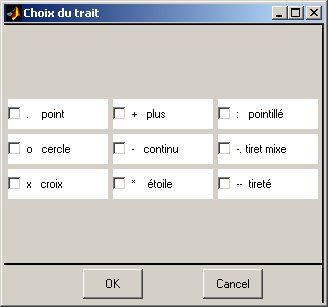


Figure  : fenêtre permettant de modifier le trait du subplot courant

Matlab >> modif\_trait(’trait\_choisi’)

#### Filtrage d’un *subplot*

Sélectionner « Filtrage » du menu déroulant, une fenêtre de dialogue identique à celle de la Figure 41 apparaît.



Figure  : fenêtre permettant de spécifier le type de filtrage à appliquer sur le signal du subplot courant

Choisir alors le type de filtre avec lequel sera filtré le signal du subplot courant. Dans le cas du filtre de Chebyshev, il est possible de spécifier une nouvelle fréquence de coupure. Mais cette option n’est pas possible dans le cas du filtre de Remez qui a été conçu plus particulièrement pour les signaux acoustiques issus de EMA.

À noter également que l’option « Cubic Spline » ne réalise pas un filtrage des données mais effectue une interpolation entre les points de la courbe.

Matlab >> filtre\_subplot(type\_filtre, freq\_coup, nb\_interp)

Où type\_filtre est l’une des chaîne suivante : 'Chebyshev', 'Spline' ou 'Remez'

**Remarque** : freq\_coup et nb\_interp seront mis à 0 lorsqu’ils ne sont pas utilisés.

### Autres fonctions de la barre d’outils

#### Couper

Il est possible de couper une partie du signal, et ce pour l’ensemble des signaux tracés dans la fenêtre principale. Pour cela, il suffit de cliquer sur le bouton « Couper » de la barre d’outils de la fenêtre principale, puis de sélectionner les positions temporelles de début et de fin de coupe, avec le bouton gauche de la souris. Dans chaque *subplot*, tout le signal compris entre ces 2 positions sélectionnées (ces positions comprises) sera coupé.

La détermination des positions temporelles de début et de fin de coupe se fait à partir des positions de la souris lues sur l’axe des abscisses du *subplot* actif (celui encadré en bleu). Tous les graphes de la fenêtre principale seront coupés de façon synchrone entre ces 2 positions.

Lorsqu’un signal acoustique a été coupé, c’est ce signal coupé qui devient le signal acoustique de référence : tous les traitements acoustiques ultérieurs (spectrogramme, LPC, FFT, formants, F0) se feront donc sur ce dernier.

Matlab >> couper (instant1, instant2)

#### Zoomer

Cliquer sur le bouton « (dé)Zoomer » de la barre d’outils de la fenêtre principale, puis sélectionner les deux positions, avec le bouton gauche de la souris, entre lesquelles il faut effectuer ce zoom. Dans le cas où on effectue plusieurs zooms successifs on peut revenir au zoom précédent en cliquant sur le bouton du milieu de la souris. Si on clique sur le bouton droit, le signal reviendra à l’échelle initiale (dézoome).

Matlab :

* agrandir :

Matlab >> zoom\_in(position1,position2)

* revenir d’un niveau de zoom

Matlab >> NB\_ZOOM = NB\_ZOOM - 1;

Matlab >> zoom\_precedent

* revenir à l’échelle initiale :

Matlab >> zoom\_out

NOTE : Si lors du remplissage de la boîte de dialogue ouverte par la demande d’une coupe sagittale (Traitement EMA), la case « Enregistrer sous » a été cochée, ainsi qu’un numéro de bobine, l’utilisation du zoom entraînera l’ouverture d’une fenêtre supplémentaire(cf. II.4.7.1).

#### Superposer

Si l’on a cliqué sur le bouton « Superposer » de la barre d’outils de la fenêtre principale, le bouton apparaît comme enclenché et le prochain graphe qui sera tracé dans le *subplot* actif sera superposé au graphe précédemment tracé dans ce *subplot*.

Pour revenir en mode de non-superposition, il suffit de cliquer de nouveau sur le bouton « Superposer ».

NB : cette fonction sera surtout utilisée pour pouvoir superposer les formants sur le sonagramme.

Matlab >> SUPERPOSE = 1 Pour passer en mode superposition.

Matlab >> SUPERPOSE = 0 Pour passer en mode non-superposition.

#### Titre de la fenêtre principale

Pour modifier le titre qui apparaît en blanc en haut de la fenêtre principale, il faut cliquer sur le bouton « TitreFen » de la barre d’outils de la fenêtre principale. Puis, dans la fenêtre de dialogue qui s’ouvre alors, il faut spécifier les coordonnées x et y du titre ainsi que le titre à afficher.

Attention, les coordonnées x et y sont calculées relativement au premier *subplot* dessiné dans la fenêtre. En général on prend donc pour x la valeur de l’abscisse au milieu de ce *subplot* et pour y une valeur légèrement supérieure à l’ordonnée maximale de ce *subplot*.

Matlab >> titre\_fen\_princ(x, y, titre\_a\_afficher)

## fonctions de commande sous Matlab

|  |  |  |
| --- | --- | --- |
| **Commandes** | **Fonctionnalités** | **Page** |
| active\_graphe | Active (encadre en bleu) le *subplot* courant. | 5 |
| couper(instant1,instant2) | Coupe tous les signaux de la fenêtre principale entre les instants 1 et 2. | 48 |
| curseur(instant) | Place le curseur à l’instant choisi et effectue les traitements synchrones demandés pour cette position. | 24 |
| eff\_etiq(pos\_etiquette) | Efface l’étiquette la plus proche de cette position. | 45 |
| eff\_minmaxppz(pos\_repere) | Efface le repère min-max le plus proche de cette position. | 45 |
| enr\_etiq(nom\_fichier) | Enregistre les étiquettes dans un fichier ASCII. | 18, 44 |
| enr\_formant(nom\_fichier) | Enregistre les formants F1 à F5 dans un fichier ASCII. | 18 |
| enr\_minmaxppz(nom\_fichier) | Enregistre dans un fichier les repères min-max-ppz-moy de chaque signal de la fenêtre. | 18, 27, 46 |
| enr\_tempo(nom\_fichier, type\_fichier, freq\_ech) | Enregistre l’extrait de signal temporel contenu dans le *subplot* courant. Trois types de format sont offerts : ’ECH’, ’MAT’ ou ’ASC’. | 18 |
| fenetre(nb\_col,nb\_lig,[coord\_x coord\_y],[dim\_x,dimy],titre) | Ouvre une fenêtre principale. | 18 |
| sig\_filt = filtre\_cheby(sig, fech, freq\_coup) | Renvoie le signal filtré par Chebyshev suivant freq\_coup. | 36, 39 |
| sig\_filt = filtre\_remez(sig, fech) | Renvoie le signal filtré par Remez. | 48 |
| filtre\_subplot(type\_filtre, freq\_coup, nb\_interp) | Filtre le signal affiché dans le subplot courant.  - type\_filtre : 'Chebyshev', 'Spline' ou 'Remez' | 48 |
| graphe\_suivant | Active (encadre de bleu) le graphe suivant le graphe couramment activé. | 5 |
| marquer(position,nom\_etiquette,type\_etiquette) | Met une étiquette à la position donnée. | 43 |
| minmaxppz((type\_reperage, type\_fichier, mode) | Repérage des minima, maxima, passages par zéro ou moyennes sur le graphe contenu dans le *subplot* courant.  - type\_repérage = 1 : min, 2 : max, 3 : ppz,  4 : moy  - type\_fichier = ’EPG’, ’EMA’, ou ’MES’.  - mode = ’auto’ ou ’manuel’. | 26 |
| modif\_echelle(xmin,xmax,ymin,ymax) | Modifie l’échelle du signal contenu dans le *subplot* courant. | 47 |
| modif\_trait(’trait\_choisi’) | Modifie le type de tracé du signal contenu dans le *subplot* courant.  trait\_choisi : ’:’, ’-’, ’.- ’, ’+’, ’x’ ou ’o’ | 47 |
| ouv\_etiq(nom\_fichier) | Affiche dans la fenêtre principale les étiquettes contenues dans le fichier. | 45 |
| quitter | Quitte TRAP. | 12 |
| synchrone(nb\_col,[coord\_x coord\_y],[dim\_x,dimy],titre) | Ouvre une fenêtre synchrone. | 19 |
| titre(titre\_signal) | Change le titre du *subplot* courant. | 47 |
| trace\_graphe(DATA (num\_chargement,:),nom\_signal) | Affiche les données d’un fichier déjà chargées. | 32, 35, 37 |
| trap | Lance le programme principal TRAP | 12 |
| zoom\_in(position1, position2) | Effectue un zoom entre ces deux instants. | 28, 49 |
| zoom\_out | Revient à l’échelle initiale. | 49 |
| zoom\_precedent | Revient au zoom précédent si plus d’un zoom déjà effectué. | 49 |
| acou\_affich\_f0(nom\_fich, type\_fich, freq\_ech) | Charge un fichier de F0 et l’affiche.  type\_fich = ’ASC’ ou ’ECH’. | 21 |
| acou\_affich\_form(nom\_fich, type\_fich) | Charge un fichier de formants et l’affiche.  type\_fich = ’ASC’ ou ’ECH’. | 20 |
| acou\_calc\_cepstre(pos\_curseur) | Calcule et affiche le cepstre du signal pour cet instant. | 23 |
| acou\_calc\_fft(pos\_curseur) | Calcule et affiche la FFT du signal pour cet instant. | 22 |
| acou\_calc\_lpc(pos\_curseur) | Calcule et affiche la LPC du signal pour cet instant. | 22 |
| acou\_detect\_f0 | Fait une détection du F0 du signal et l’affiche. | 21 |
| acou\_detect\_form | Fait une détection des formants du signal acoustique et l’affiche. | 20 |
| acou\_ecout\_ext(instant1, instant2) | Permet d’écouter un extrait du signal acoustique. | 21 |
| acou\_ecout\_fen | Permet d’écouter la fenêtre de signal acoustique visible dans le *subplot* actif. | 21 |
| acou\_ecout\_tout | Permet d’écouter tout le signal acoustique. | 21 |
| acou\_filtres(type\_filtre, freq\_coupure) | Filtre le signal acoustique et l’affiche. | 19 |
| acou\_intensite | Calcule l’évolution de l’intensité (RMS) du signal acoustique et l’affiche | 20 |
| acou\_ouv(nom\_fich, type\_fich) | Ouvre un fichier acoustique binaire de type : ’EMA’, ’EPG’, ’ECH’, ’MAT’, ’WAV’ ou ’BIN. | 16 |
| acou\_signal | Trace le signal acoustique dans la fenêtre principale. | 19 |
| acou\_spectro | Calcul le sonagramme du signal acoustique contenu dans le *subplot* actif et l’affiche. | 20 |
| egg\_acc(signal\_egg, nom\_EGG) | Calcule l’accélération d’un signal de type EGG et l’affiche. | 41 |
| egg\_ouv(signal\_egg, nom\_EGG) | Ouvre le fichier EGG spécifié. | 17 |
| egg\_traj(signal\_egg, nom\_EGG) | Affiche la trajectoire d’un fichier EGG. | 41 |
| egg\_vit(signal\_egg, nom\_EGG) | Calcule la vitesse d’un signal de type EGG et l’affiche. | 41 |
| ema\_acc(signal,nom\_signal) | Calcule l’accélération d’une bobine et l’affiche. | 25 |
| ema\_animer(position1, position2) | Affiche la progression de la trajectoire des bobines entre ces deux instants. | 30 |
| ema\_calc\_tilt | Calcul et affiche les facteurs tilt des 5 bobines dans une fenêtre à part. | 31 |
| ema\_coupe\_sagit(instant) | Affiche la coupe sagittale à cet instant. | 29 |
| ema\_enr\_pal | Enregistre le contour du palais, affiché dans la fenêtre réservée au suivi de contour, dans le fichier *.pal* dont le nom a été préalablement spécifié. | 28 |
| ema\_ouv\_mat(nom\_ fichier) | Ouvre un fichier de données articulatoires EMA (.mat). | 16 |
| ema\_ouv\_pal(nom\_fichier) | Charge le fichier contenant le contour du palais. | 29 |
| ema\_traitement | Effectue le traitement des données EMA (normalisation, rotation et filtrage éventuels). | 11, 24 |
| ema\_traj (signal\_bobine, nom\_signal) | Affiche la trajectoire X ou Y d’une bobine (EMA\_X1…5, EMA\_Y1..5). | 24 |
| ema\_vit(signal\_bobine, nom\_signal) | Calcule la vitesse d’une bobine et l’affiche. | 25 |
| ema\_vit\_tang(signal\_x\_bobine, signal\_y\_bobine, nom\_signal) | Calcule la vitesse tangentielle d’une bobine et l’affiche. | 25 |
| epg\_acc(EPG\_DATA\_CONS(num\_chargement,:), nom\_signal) | Calcule l’accélération de données de constriction et l’affiche. | 32 |
| epg\_ouv\_cons(extension\_fich,nom\_signal) | Charge un fichier associé au fichier EPG chargé et contenant des données de constriction. Puis affiche ces données. | 32 |
| epg\_sauve\_reg | Sauvegarde les régions dans le fichier *.usr*. | 32 |
| epg\_select\_reg ([num\_elect1…num\_electN],numero\_region,nom\_region) | Définit une région de numéro numero\_region et de nom nom\_region avec les électrodes données entre crochets. | 31 |
| epg\_serie(instant) | Affiche une série de palais avant et après l’instant donné. | 34 |
| epg\_simple(instant) | Affiche les électrodes du palais contactées à cet instant. | 34 |
| epg\_taux\_cons(numero\_region) | Calcule le taux de constriction dans la région spécifiée et affiche l’évolution temporelle de ce taux. | 32 |
| epg\_vit(EPG\_DATA\_CONS (num\_chargement,:), nom\_signal) | Calcule la vitesse de données de constriction et l’affiche. | 32 |
| eva\_acc(signal\_eva, type\_fic) | Calcule l’accélération d’un signal de type EVA et l’affiche. | 39 |
| eva\_ouv(fichier\_eva, type\_fic) | Ouvre le fichier EVA spécifié. | 17 |
| eva\_traj(signal\_eva, type\_fic) | Affiche la trajectoire d’un fichier EVA. | 38 |
| eva\_vit(signal\_eva, type\_fic) | Calcule la vitesse d’un signal de type EVA et l’affiche. | 39 |
| mes\_acc (MES\_DATA(numero\_mesure, :), nom\_ mesure) | Calcule l’accélération d’une mesure et l’affiche. | 36, 38 |
| mes\_ouv(nom\_fichier) | Ouvre un fichier de mesures ASCII. | 16 |
| mes\_vit(MES\_DATA(numero\_mesure, :), nom\_mesure) | Calcule la vitesse d’une mesure et l’affiche. | 35, 38 |
| tac\_ouv(fichier\_tac) | Ouvre le fichier TACLE spécifié. | 16 |

Figure   :liste des fonctions utilisées en mode script sous TRAP

# Exemples de scripts sous TRAP

## Exemple de Traitement ACOUStique

### Exemple de fichier script

% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Fonction : exemple\_acous('nom\_fich\_wav')

%

% Utilisation :

exemple\_acous('Q:\scripts\_matlab\trap\trap\_v5\exemples\data\wav\jls\_pata\_1.wav')

% Source : Matlab

% Auteur : D. Lefebvre

% Date de création : 21.06.99

% Modifié par C. Savariaux et H. Loevenbruck

% Dernière Modification CS le 02/06/04

% Contenu : Fichier d'exemple de traitement acoustique

% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function exemple\_acous(nom\_fich\_wav)

% Lancement de l'application

trap;

% Initialisation des variables globales

ACOU\_FECH = 20000; % Freq. d'échantillonnage du signal acoustique

% Ouverture du fichier son

acou\_ouv(nom\_fich\_wav, 'WAV');

% Ouverture de la fenetre principale

fenetre(1,2,[20 90],[800 500],'Fenetre principale');

% Ouverture de 3 subplot ds la fenetre synchrone

synchrone(3,[20 620],[1000 300],'Fenetre synchrone');

% Visualisation du signal acoustique (filtré)

acou\_signal;

% Visualisation du spectrogramme

ACOU\_OVER\_FEN = 64;

graphe\_suivant;

acou\_spectro;

% Détermination des formants + affichage sur le spectrog.

SUPERPOSE=1;

% La premiere fois

acou\_detect\_form;

%on les sauve dans un fichier

enr\_formant('Q:\scripts\_matlab\trap\trap\_v5\exemples\data\wav\jls\_pata\_1\_formant.txt');

% A partir de la seconde fois

%acou\_affich\_form('Q:\scripts\_matlab\trap\trap\_v5\exemples\data\wav\jls\_pata\_1\_formant.txt','ASC')

SUPERPOSE=0;

my\_axis = axis;

modif\_echelle(my\_axis(1), my\_axis(2), 0 , 5000);

% Visualisation du spectre du signal, de la LPC, du cepstre à t = 0.83 sec

ACOU\_ORD\_FFT = 512;

ACOU\_LONG\_FEN = 128;

ACOU\_FEN\_TMP = hanning(ACOU\_LONG\_FEN); % Vecteur colonne

ACOU\_CALC\_FFT = 1;

ACOU\_ORD\_LPC = 44;

ACOU\_CALC\_LPC = 1;

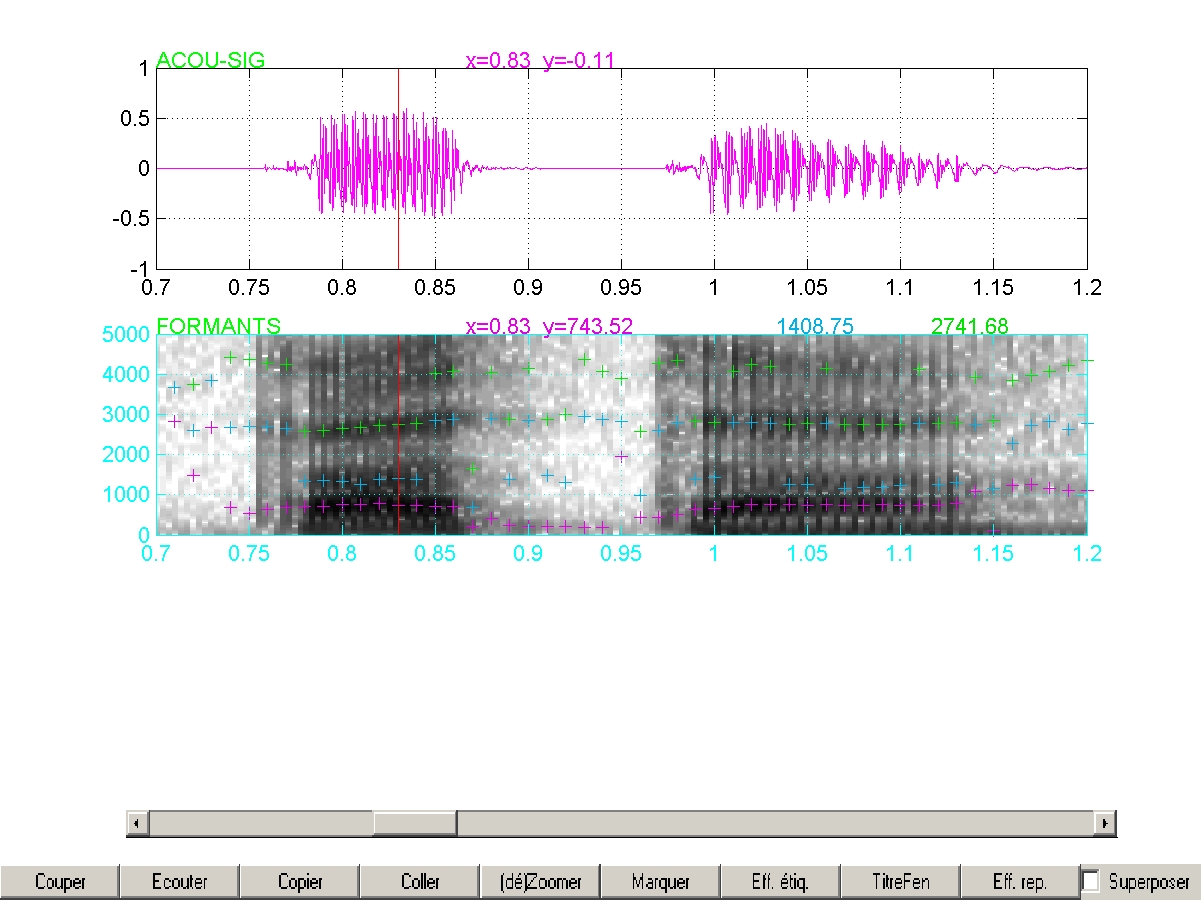
ACOU\_CEPS\_VIS = 4; % Locuteur Homme

ACOU\_CALC\_CEPS = 1;

zoom\_in(0.7,1.2);

curseur(0.83);

### Résultat obtenu après l’exécution du script



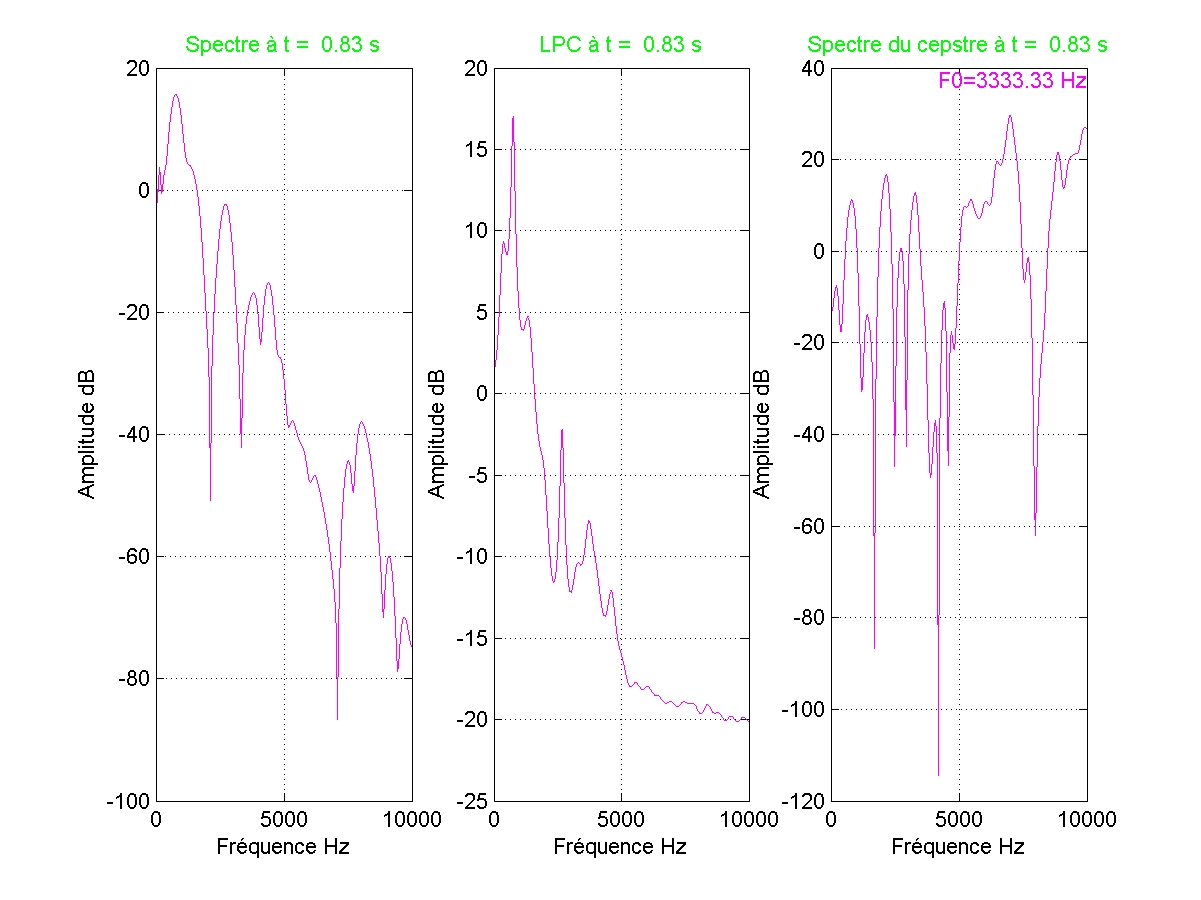


Figure : résultat des différentes fenêtres obtenues après traitement acoustique.

## Exemple de Traitement des donnÉes EMA

### Exemple de fichier script pour EMA

% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Fonction : exemple\_ema('nom\_fich\_ema', 'nom\_fich\_wav')

%

% Source : Matlab

% Auteurs : C. Savariaux & H. Loevenbruck

% Date de création : 1 mars 2002

% Dernière modification : 18 mai 2004

%

% exemple\_ema('Q:\scripts\_matlab\trap\trap\_v5\exemples\data\ema\jls\_pata\_1.mat',

% 'Q:\scripts\_matlab\trap\trap\_v5\exemples\data\ema\jls\_pata\_1.wav')

% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function exemple\_ema(nom\_fich\_ema, nom\_fich\_wav)

% Lancement de l'application

trap;

% Initialisation des variables globales

ACOU\_FECH = 20000; % Freq. d'échantillonnage du signal acoustique

EMA\_FECH = 500; % Freq. d'échantillonnage du signal EMA

CHEBY\_FREQ\_COUP = 20; % Freq de coupure du Pbas de Chebyshev

% si CHEBY\_FREQ\_COUP=0 pas de filtrage (des données, de la vitesse ou de l'accélération)

EMA\_ECHELLE = 1; % Unité de visualisation en centimètre

% Définition du nom des bobines

EMA\_BOB1 = 'NEZ';

EMA\_BOB2 = 'INC-SUP';

EMA\_BOB3 = 'TNG-BACK';

EMA\_BOB4 = 'TNG-DOS';

EMA\_BOB5 = 'TNG-MILI';

EMA\_BOB6 = 'TNG-APEX';

EMA\_BOB7 = 'LEV-SUP';

EMA\_BOB8 = 'LEV-INF';

EMA\_BOB9 = 'JAW';

EMA\_BOB10 = '';

% NOTE : à bien faire avant d'ouvrir les fenetres

% Chargement du fichier de données articulatoires

ema\_ouv\_mat(nom\_fich\_ema);

% Chargement du fichier de données acoustiques

acou\_ouv(nom\_fich\_wav, 'WAV');

% Ouverture des fenetres principales et synchrone

% Ouverture de 10 subplot sur 2 colonnes

fenetre(2,5,[20 90],[900 500],'Fenetre principale');

% Ouverture du subplot

synchrone(1,[20 600],[500 500],'Fenetre synchrone');

% Mise en forme des données EMA

ema\_traitement;

%CHEBY\_FREQ\_COUP = 20;

% Affichage des trajectoires, vitesses et accélérations de quelques bobines

% ---------------------------------------------------------------------------

% Trajectoire de la bobine 4 en X

ema\_traj(EMA\_X4,[EMA\_BOB4,'-X']);

% Trajectoire de la bobine 4 en Y

graphe\_suivant;

ema\_traj(EMA\_Y4,[EMA\_BOB4,'-Y']);

% Vitesse de la bobine 4 en X non filtrée

graphe\_suivant;

CHEBY\_FREQ\_COUP=0; % Pour ne pas appliquer de filtrage

ema\_vit(EMA\_X4,[EMA\_BOB4,'-X']);

% Accélération de la bobine 4 en X

graphe\_suivant;

ema\_acc(EMA\_X4,[EMA\_BOB4,'-X']);

% Vitesse tangentielle de la bobine 4

graphe\_suivant;

ema\_vit\_tang(EMA\_X4, EMA\_Y4, EMA\_BOB4);

% Trajectoire de la bobine 9 en Y

graphe\_suivant;

ema\_traj(EMA\_Y9,[EMA\_BOB9,'-Y']);

% Visualisation du signal acoustique

graphe\_suivant;

acou\_signal;

% Trajectoire de la bobine 7 en X

graphe\_suivant;

ema\_traj(EMA\_X7,[EMA\_BOB7,'-X']);

% Visualisation du spectrogramme

ACOU\_OVER\_FEN = 64;

graphe\_suivant;

acou\_spectro;

% Trajectoire de la bobine 7 en Y

graphe\_suivant;

ema\_traj(EMA\_Y7,[EMA\_BOB7,'-Y']);

% zoom sur une partie du signal

zoom\_in(2,2.5);

% Vérification des facteurs Tilts (doivent etre proches de 1)

ema\_calc\_tilt;

% Chargement du fichier contenant le contour du palais

EMA\_BOB\_PAL = 0; % pas d'ouverture de fenetre supplémentaire

ema\_ouv\_pal('Q:\scripts\_matlab\trap\trap\_v5\exemples\data\ema\jls\_palais.pal');

% Visualisation de la coupe sagittale à t = 2.1 sec

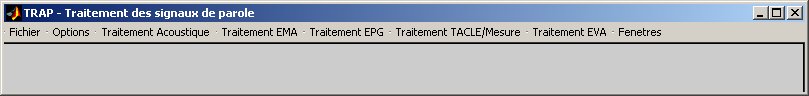
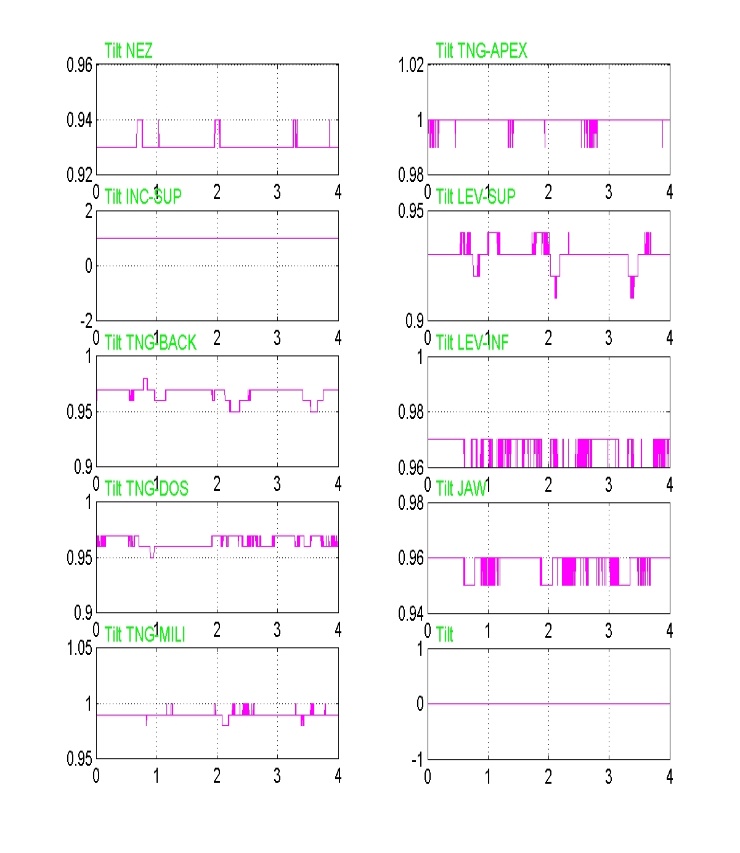
EMA\_COUP\_SAG = 1;

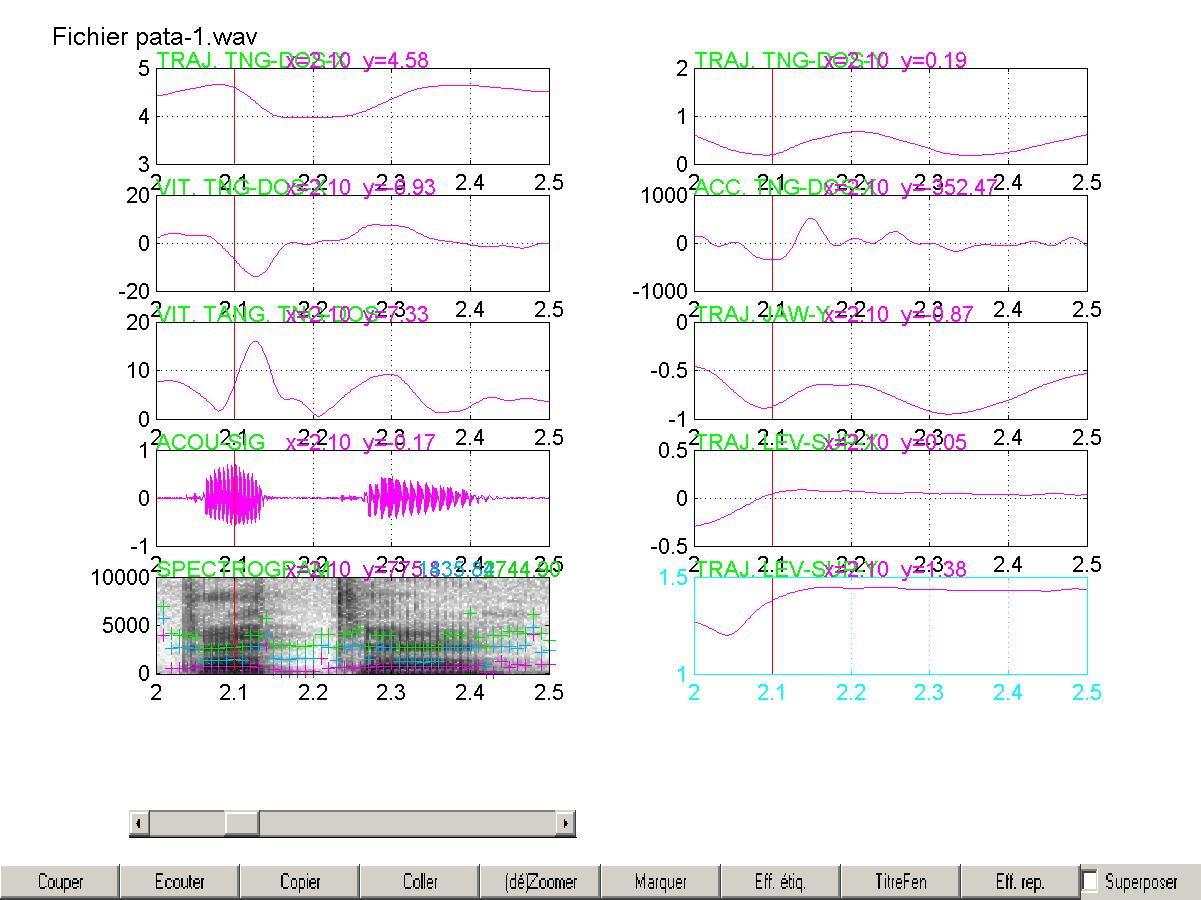
curseur(2.1);

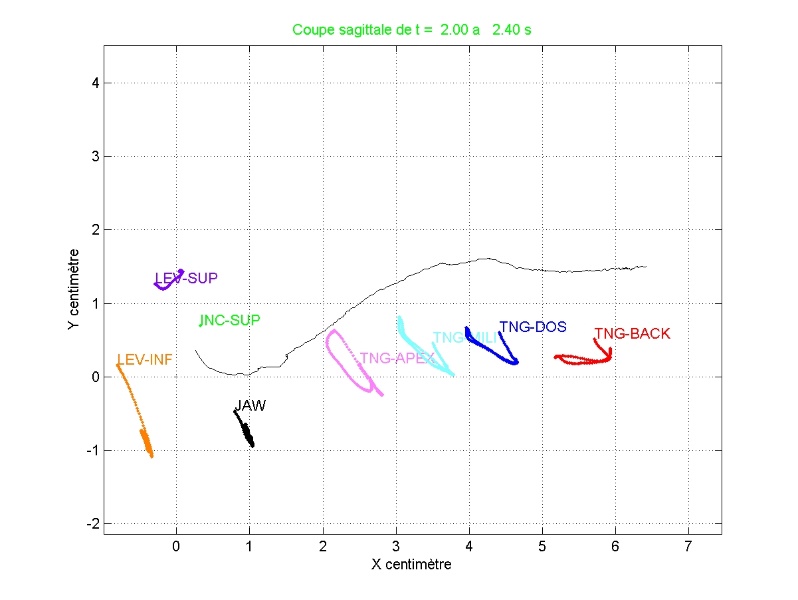
% animation sur une répétition de t = 2 sec à 2.4

ema\_animer(2,2.4)

### Résultat obtenu après l’exécution du script





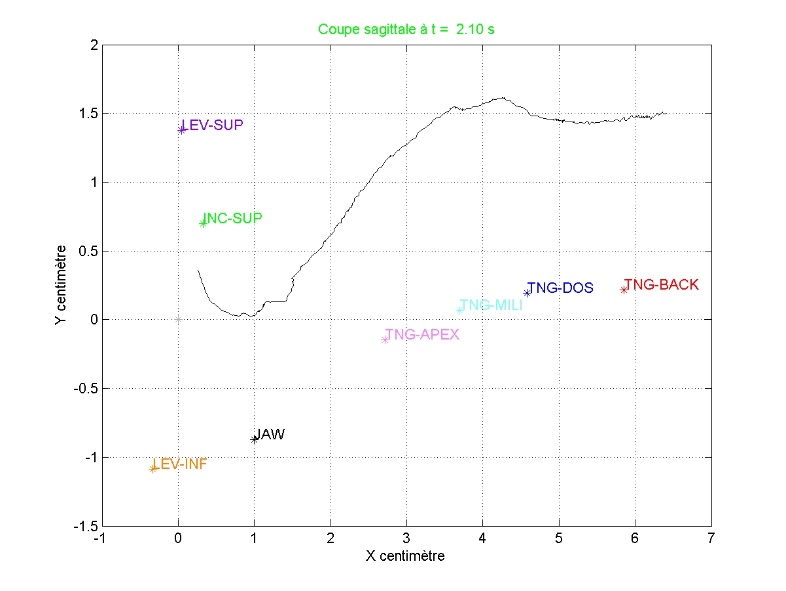


Figure  : résultat des différentes fenêtres obtenues après traitement des données EMA.

## Exemple de Traitement des donnÉes EPG

### Exemple de fichier script pour EPG

% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Fichier : exemple\_epg.m

% Source : Matlab

% Auteur : D. Lefebvre

% Date de création : 21.06.99

% Dernière Modif : CS le 18/05/04

% Contenu : Fichier d'exemple pour les données EPG

% du fichier m009s2.nsp

% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Lancement de l'application

trap;

% Initialisation des variables globales servant a l'ouverture du

% fichier

EPG\_NOM\_LOC = 'Q:\Scripts\_matlab\trap\trap\_v5\exemples\data\epg\mbeckman'; % Nom du fichier .usr

ACOU\_FECH = 12800; % Freq. d'echantillonnage du signal acoustique

EPG\_FECH = 100; % Freq. d'echantillonnage du signal EPG

% Chargement du fichier binaire

acou\_ouv('Q:\Scripts\_matlab\trap\trap\_v5\exemples\data\epg\mbeckman.nsp', 'EPG');

% Ouverture des fenetres principales et synchrones

fenetre(1,6,[20 90],[600 700],'Fenetre principale');

synchrone(2,[630 420],[500 300],'Fenetre synchrone');

% Visualisation du signal acoustique

acou\_signal;

% Visualisation du spectrogramme

ACOU\_ORD\_FFT = 512;

ACOU\_LONG\_FEN = 128;

ACOU\_FEN\_TMP = hanning(ACOU\_LONG\_FEN); % Vecteur colonne

ACOU\_OVER\_FEN = 64;

graphe\_suivant;

acou\_spectro;

% Visualisation des TTCL et TTCD

graphe\_suivant;

epg\_ouv\_cons('ca1','TTCLoc');

modif\_trait('--');

%Recherche des maxima sur TTCLoc

minmaxppz(2,'EPG','auto')

graphe\_suivant;

epg\_ouv\_cons('ca2','TTCDeg');

% Visualisation des vitesses des TTCL et TTCD

graphe\_suivant;

epg\_vit(EPG\_DATA\_CONS(1,:),'TTCLoc');

modif\_trait('--');

graphe\_suivant;

epg\_vit(EPG\_DATA\_CONS(2,:),'TTCDeg');

% Visualisation du spectre du signal et des palais simple et séries à

% t=0.85 sec

ACOU\_CALC\_FFT = 1;

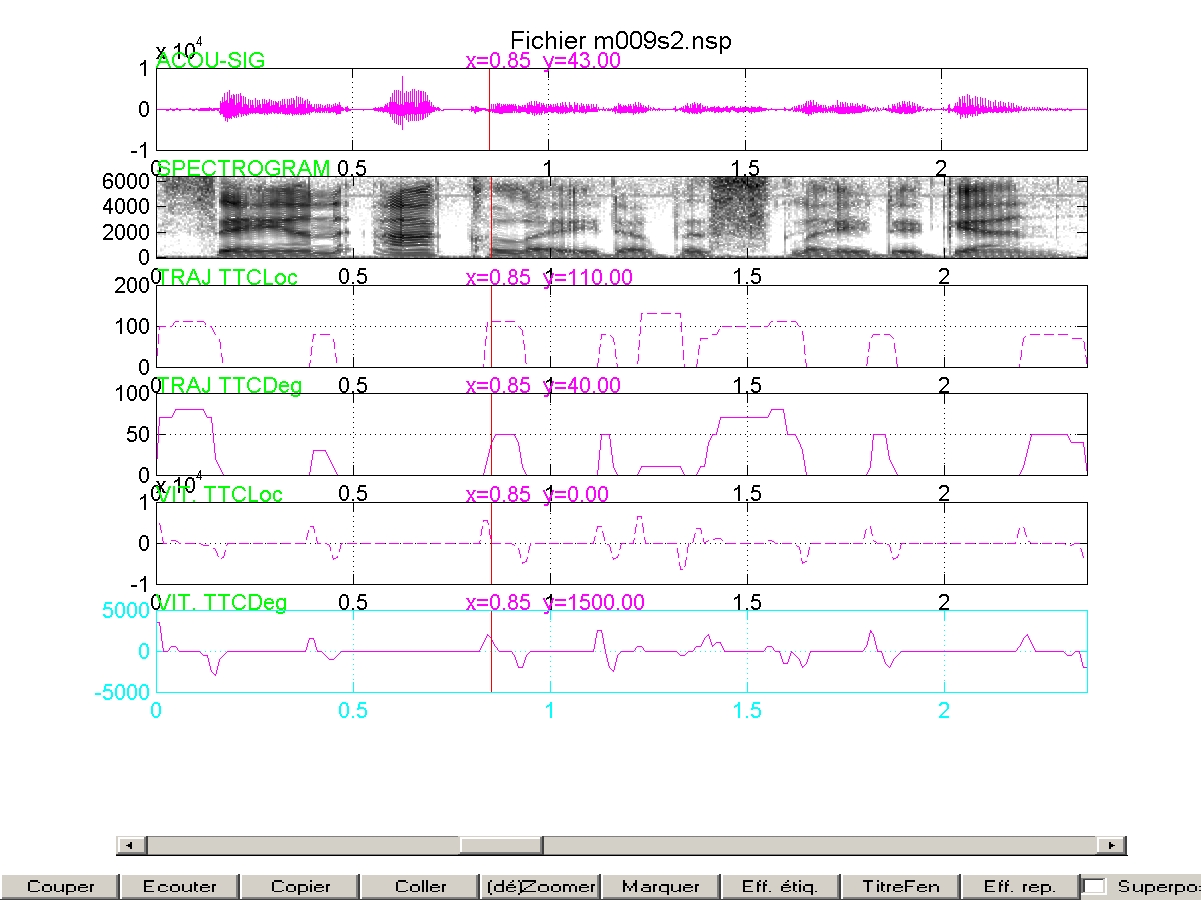
EPG\_PAL\_SIMPLE = 1;

EPG\_PAL\_SERIE = 1;

EPG\_NB\_PAL = 3;

curseur(0.85);

### Résultat obtenu après l’exécution du script



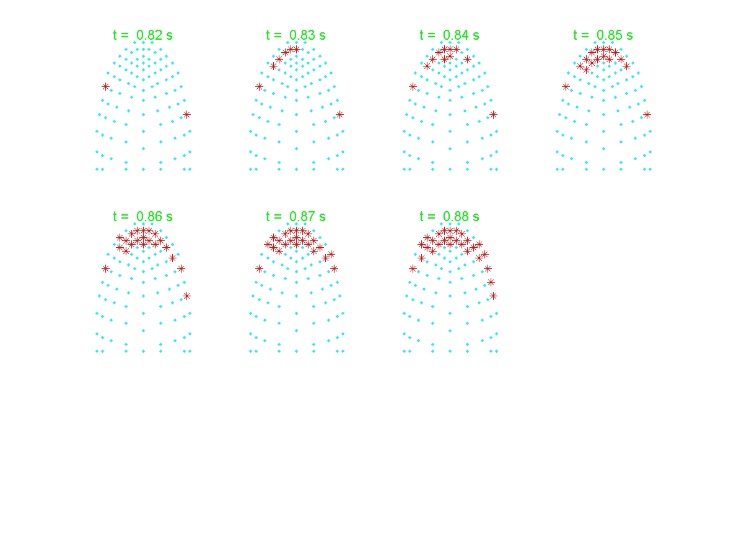
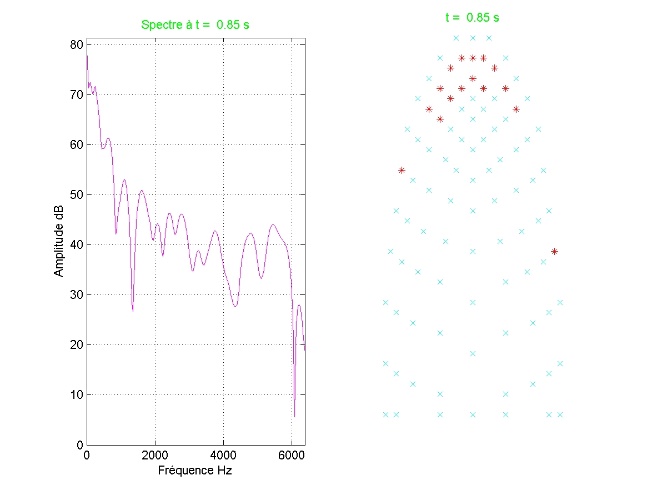


Figure  : résultat des différentes fenêtres obtenues après traitement des données EPG.

## Exemple de Traitement des donnÉes TACLE

### Exemple de fichier script pour TACLE

% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Fichier : exemple\_tac.m

% Source : Matlab

% Auteur : C. Savariaux

% Date de création : 18.09.02

% Date de dernière modification : CS le 18/05/04

% Contenu : Fichier d'exemple pour les fichiers

% de paramètres labiaux issus de TACLE (.tac)

% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Lancement de l'application

trap;

COULEUR\_GRAPH = [0 0 0];

COULEUR\_FOND = [1 1 1];

COULEUR\_ACTIVE = [.3 .3 .3];

COULEUR\_AXES = [0 0 0];

% Initialisation des variables globales

ACOU\_FECH = 22050; % Freq. d'échantillonnage du signal acoustique

CHEBY\_FREQ\_COUP = 10; % Freq. coupure pour le filtrage des mesures

% Chargement du fichier acoustique binaire

acou\_ouv('Q:\scripts\_matlab\trap\trap\_v5\exemples\data\tac\marie\_skult.wav', 'WAV')

% Ouverture des fenetres principales et synchrones

fenetre(1,7,[20 90],[800 650],'Fenetre principale');

% Visualisation du signal acoustique

acou\_signal;

titre\_signal = sprintf('sikskult');

titre(titre\_signal);

% Visualisation du spectrogramme

%graphe\_suivant;

ACOU\_ORD\_FFT = 1024;

ACOU\_LONG\_FEN = 128;

ACOU\_FEN\_TMP = hanning(ACOU\_LONG\_FEN); % Vecteur colonne

ACOU\_OVER\_FEN = 64;

%acou\_spectro;

%mes\_axes = axis;

% Determination des formants + affichage sur le spectrog.

%SUPERPOSE = 1;

%acou\_detect\_form;

%modif\_echelle(mes\_axes(1),mes\_axes(2), 0, 5000);

%SUPERPOSE = 0;

% Nombre et nom des paramètres souhaités

MES\_NB = 4;

MES\_NOM(1,1:length('A')) = 'A';

MES\_NOM(2,1:length('B')) = 'B';

MES\_NOM(3,1:length('S')) = 'S';

MES\_NOM(4,1:length('P1')) = 'P1';

% Fréquence d'échantillonage des données TACLE

MES\_FECH = 50;

% La premiere fois, le fichier .lev n'existe pas

% Ouverture du fichier TACLE

%tac\_ouv('Q:\scripts\_matlab\trap\trap\_v5\exemples\data\tac\marie\_skult.tac');

% A partir de la seconde fois

mes\_ouv('Q:\scripts\_matlab\trap\trap\_v5\exemples\data\tac\marie\_skult.lev');

% Tracé de A

graphe\_suivant;

trace\_graphe(MES\_DATA(1,:),MES\_TPS,'A', '-', COULEUR\_GRAPH);

% Tracé de B

graphe\_suivant;

trace\_graphe(MES\_DATA(2,:),MES\_TPS,'B', '.', COULEUR\_GRAPH);

% Tracé de S

graphe\_suivant;

trace\_graphe(MES\_DATA(3,:),MES\_TPS,'S', '-', COULEUR\_GRAPH);

% Tracé de P1

graphe\_suivant;

trace\_graphe(MES\_DATA(4,:),MES\_TPS,'P1', '-', COULEUR\_GRAPH);

% on filtre P1

graphe\_suivant;

P1\_filt = filtre\_cheby(MES\_DATA(4,:), MES\_FECH, CHEBY\_FREQ\_COUP);

trace\_graphe(P1\_filt,MES\_TPS,'P1-filt', '-', COULEUR\_GRAPH);

% et on calcule sa vitesse

graphe\_suivant;

mes\_vit(P1\_filt,'P1-filt')

% on zoom

zoom\_in(0.5, 2)

### Résultat obtenu après l’exécution du script

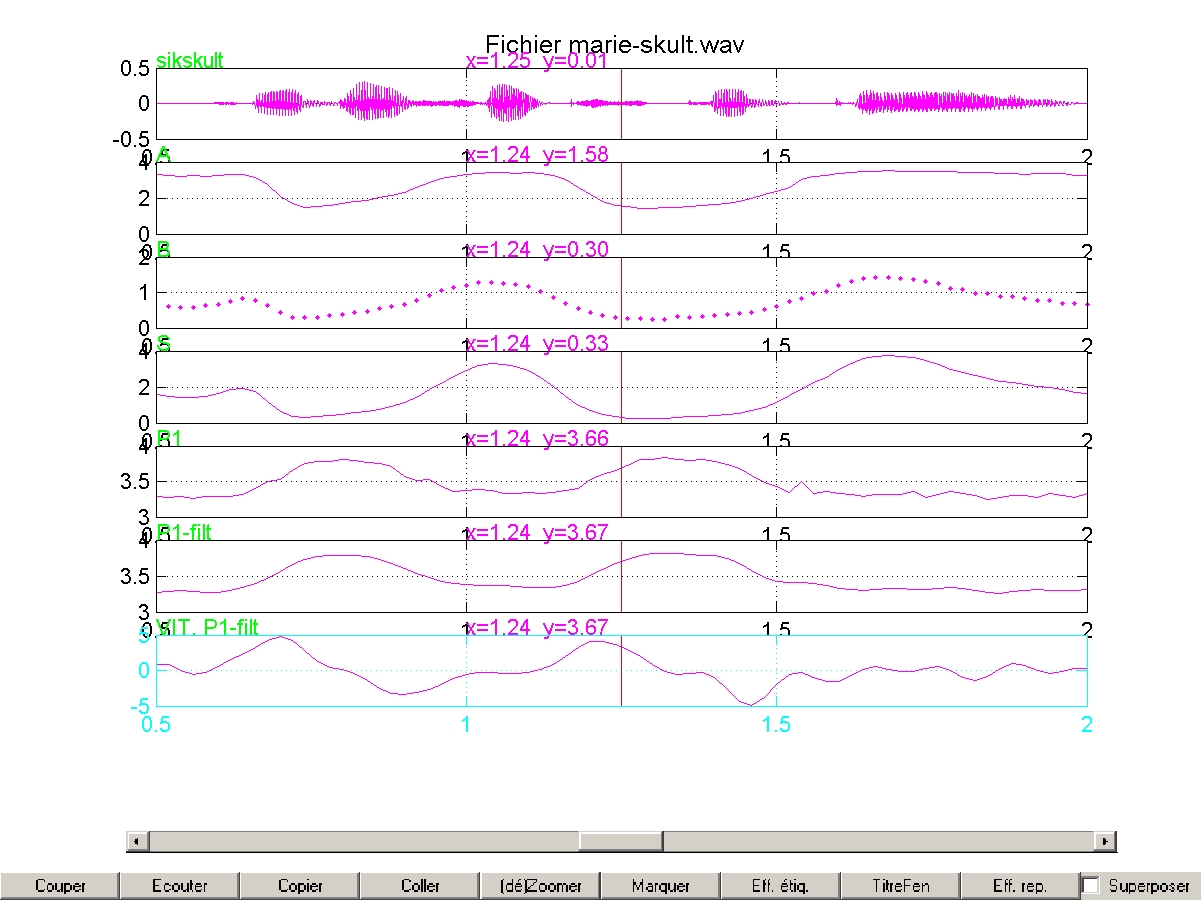


Figure  : résultat des différentes fenêtres obtenues après traitement des données TACLE

## Exemple de Traitement des donnÉes EVA

### Exemple de fichier script pour EVA

% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Fichier : exemple\_eva.m

% Source : Matlab

% Auteur : C. Savariaux

% Date de création : 02/12/03

% Derniere modif : CS le 18/05/04

% Contenu : Fichier d'exemple pour les données

% issues d'EVA

% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Lancement de l'application

trap;

CHEBY\_FREQ\_COUP = 20; % Freq de coupure du Pbas de Chebyshev

% si CHEBY\_FREQ\_COUP=0 pas de filtrage (des données, de la vitesse ou de l'accélération)

% ouverture du fichier Wav

% On considere ce signal comme signal de ref Wav sur lequel on applique le

% spectrogramme

eva\_ouv('Q:\Scripts\_matlab\trap\trap\_v5\exemples\data\eva\pb\_pana.wa1','WA1');

% Ouverture des fenetres principales et synchrones

fenetre(1,7,[20 90],[800 650],'Fenetre principale');

% Visualisation du signal acoustique

acou\_signal;

titre\_signal = sprintf('pana pana');

titre(titre\_signal);

% ouverture du fichier de F0

eva\_ouv('Q:\Scripts\_matlab\trap\trap\_v5\exemples\data\eva\pb\_pana.F0','F0');

% Visualisation

graphe\_suivant;

eva\_traj(EVA\_F0, 'F0');

% calcule et tracé de la vitesse de F0

graphe\_suivant;

CHEBY\_FREQ\_COUP = 0; % pour ne pas filtrer la vitesse

eva\_vit(EVA\_F0, 'F0');

% ouverture du fichier oaf

eva\_ouv('Q:\Scripts\_matlab\trap\trap\_v5\exemples\data\eva\pb\_pana.oaf','OAF');

% Visualisation

graphe\_suivant;

eva\_traj(EVA\_OAF, 'OAF');

% ouverture du fichier naf

eva\_ouv('Q:\Scripts\_matlab\trap\trap\_v5\exemples\data\eva\pb\_pana.naf','NAF');

% Visualisation

graphe\_suivant;

eva\_traj(EVA\_NAF, 'NAF');

% ouverture du fichier de pression

eva\_ouv('Q:\Scripts\_matlab\trap\trap\_v5\exemples\data\eva\pb\_pana.pr1','PR1');

% Visualisation

graphe\_suivant;

eva\_traj(EVA\_PR1, 'PR1');

% on filtre les données

CHEBY\_FREQ\_COUP = 20;

[EVA\_PR1\_filt] = filtre\_cheby(EVA\_PR1, EVA\_FECH2, CHEBY\_FREQ\_COUP);

graphe\_suivant;

eva\_traj(EVA\_PR1\_filt, 'PR1-filt');

zoom\_in(11.5, 14.5);

titre\_fen\_princ(13, 2, 'Fichier pb\_pana.wav')

### Résultat obtenu après l’exécution du script

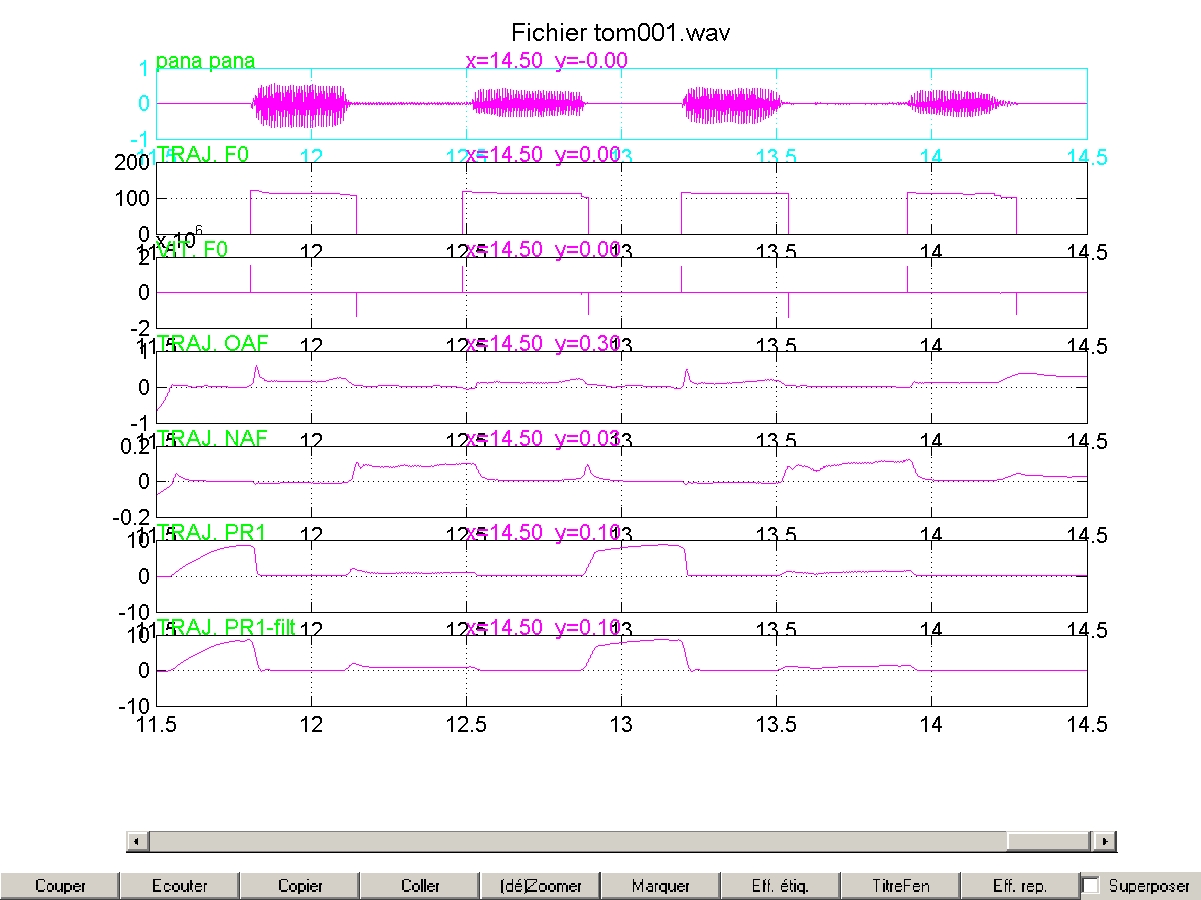


Figure  : résultat des différentes fenêtres obtenues après traitement des données EVA

## Exemple d’utilisation des étiquettes

### Exemple de script utilisant un fichier d’étiquettes issu de PRAAT

% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Fonction : exemple\_etiq\_praat('nom\_fich\_wav', 'nom\_fich\_etiq')

%

% Utilisation : exemple\_etiq\_praat('Q:\Scripts\_matlab\trap\trap\_v5\exemples\data\wav\cs\_pg4.wav',

% 'Q:\Scripts\_matlab\trap\trap\_v5\exemples\data\wav\cs\_pg4.textgrid')

% Source : Matlab

% Auteur : C. Savariaux

% Date de création : 25/05/04

% Contenu : Fichier d'exemple pour la pose des étiquettes a partir d'un fichier

% .textgrid issu de PRAAT

% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function exemple\_etiq\_praat(nom\_fich\_wav, nom\_fich\_etiq)

% Lancement de l'application

trap;

% Initialisation des variables globales

ACOU\_FECH = 20000; % Freq. d'echantillonnage du signal acoustique

NIV\_ETIQ = 4;

COULEUR\_ETIQ(1,:) = [0 0.9 0.2];

COULEUR\_ETIQ(2,:) = [1 0.9 0.4];

COULEUR\_ETIQ(3,:) = [1 0 0.6];

COULEUR\_ETIQ(4,:) = [0 0.9 0.8];

TYPE\_ETIQ(:,:)='';

TYPE\_ETIQ(1,1:length('Durée\_C'))='Durée\_C';

TYPE\_ETIQ(2,1:length('VOT\_L'))='VOT\_L';

TYPE\_ETIQ(3,1:length('VOT\_K'))='VOT\_K';

TYPE\_ETIQ(4,1:length('Bruit'))='Bruit';

% Chargement du fichier de données acoustiques

acou\_ouv(nom\_fich\_wav, 'WAV');

% Ouverture des fenetres principales et synchrones

fenetre(1,1,[20 90],[800 500],'Fenetre principale');

% Affichage du signal acoustique

acou\_signal;

% Ouverture du fichier d'étiquettes

ouv\_etiq(nom\_fich\_etiq)

% Sauvegarde des etiquettes au format TRAP

enr\_etiq('Q:\Scripts\_matlab\trap\trap\_v5\exemples\data\wav\cs\_pg4.etiq')

### Résultat obtenu après l’exécution du script

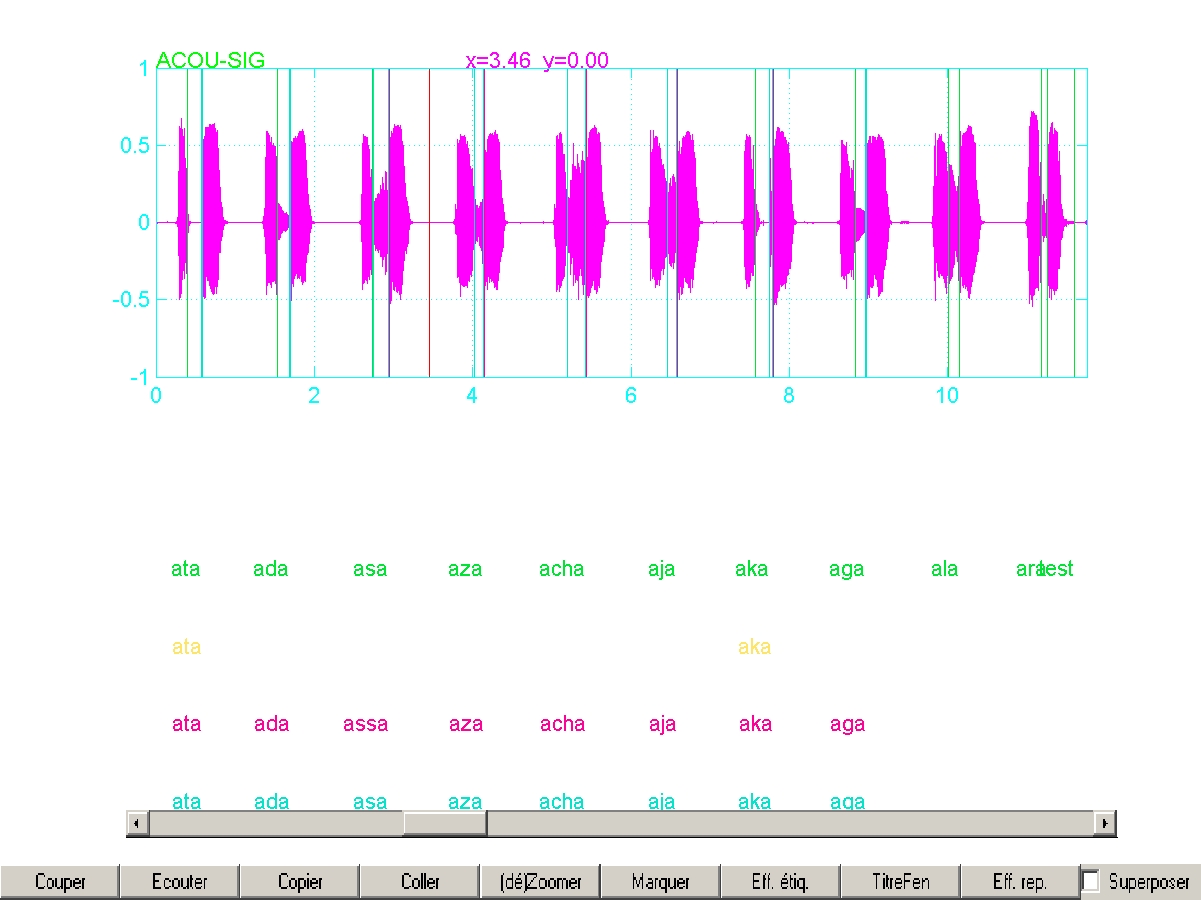


Figure  : exemple d’utilisation du fichier de segmentation issu de PRAAT

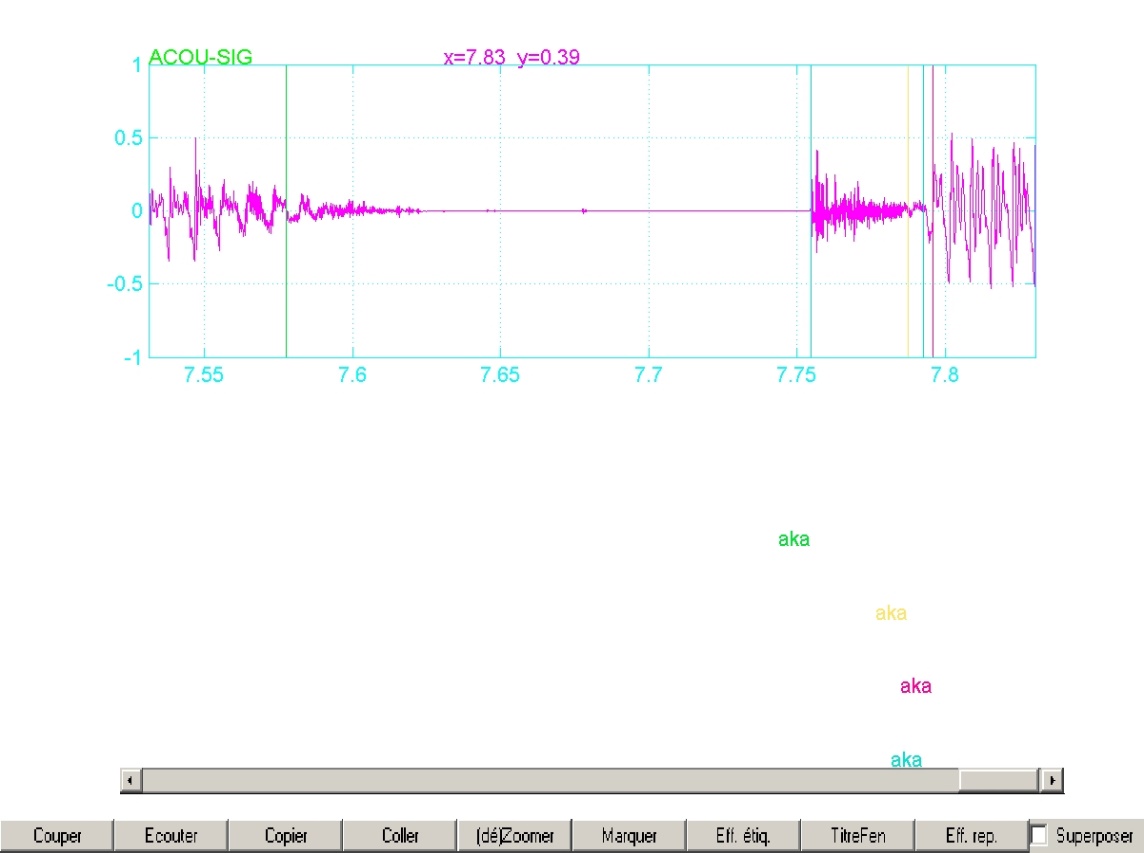


Figure  : zoom sur une partie du signal

1. Cette fonctionnalité est temporairement supprimée. [↑](#footnote-ref-1)