

**Treball Final de Carrera**

*Programació d'una aplicació gràfica per  
l'anàlisi de senyals usant  
LDB (Local Discriminant Bases)  
i MLDB (Modified LDB)*

Lídia Casadesús Martínez

**Enginyeria Tècnica de Telecomunicacions, esp. Sistemes de  
Telecomunicació**

Director: Pere Martí Puig

Vic, setembre de 2008

# Índex

<b>1. Resum</b>	<b>5</b>
<b>2. Introducció</b>	<b>7</b>
<b>3. Els otòlits</b>	<b>8</b>
3.1. Què són	8
3.2. Estudis	9
3.3. Presa de les imatges	9
<b>4. Presentació del funcionament de l'aplicació gràfica desenvolupada</b>	<b>10</b>
4.1. Funcionament de l'Assistent	11
4.2. Funcionament de la interfície principal	16
4.2.1. Menús	16
4.2.2. Importació de variables	18
4.2.3. Representació del senyal	20
4.2.4. Wavelet Packet Transform i la seva reconstrucció	23
4.2.5. Discriminant Analysis (LDB) <i>Standard</i> i la seva reconstrucció	25
4.2.6. Discriminant Analysis (LDB) <i>Custom</i> i la seva reconstrucció	28
4.2.7. Exportació de variables	31
<b>5. Funcions desenvolupades en la realització de la interfície gràfica</b>	<b>32</b>
5.1. Creació de la interfície principal	33
5.2. Creació de la interfície Assistent (Wizard)	37
5.3. Creació de la interfície Exportar/Importar	38
5.4. Funcions d'execució dels elements	42
5.4.1. Creació de $N^k$ combinacions. Funció <i>ButtonCreateComb</i>	44
5.4.2. Creació d'un vector a partir d'una cadena. Funció <i>ComparaCadena</i>	47
5.4.3. Habilitació o inhabilitació dels components. Funció <i>EnableDisable</i>	48
5.4.4. Modificació de la interfície inicial. Funció <i>Modif_interf_inicial</i>	51
5.4.5. Arxivament de la sessió actual. Funció <i>MenuSaveSession</i>	52
5.4.6. Càrrega de la sessió desada. Funció <i>MenuLoadSession</i>	54
<b>6. Funcions externes a la interfície</b>	<b>55</b>
6.1. Algorisme de discriminació de trets característics	56
6.1.1. Algorisme de discriminació de trets característics — LDB	56
6.1.1.1. Introducció	56
6.1.1.2. Procediment	56
6.1.2. Algorisme de discriminació de trets característics — MLDB	58
6.1.2.1. Procediment	58
6.2. Extracció LDB. Funció <i>f_mlldb2</i>	59
6.2.1. Arguments de Standard Discriminant Analysis	62
6.2.2. Arguments de Custom Discriminant Analysis	64
6.2.3. Valors de sortida	66
6.3. Extracció Wavelet packet transform. Funció <i>fWP_DecTree</i>	70
6.3.1. Funció prèvia al pas d'arguments. <i>wfilters</i>	71
6.3.2. Arguments	71
6.3.3. Valor de sortida	72
6.4. Reconstrucció. Funció <i>fIWPT_RecTree</i>	72
6.4.1. Reconstrucció de senyals provinents d'extracció LDB	73
6.4.1.1. Funcions prèvies al pas d'arguments. <i>wfilters</i> i <i>Paraunitary_m</i>	73

6.4.1.2. Arguments per la reconstrucció de Discriminant Analysis	74
6.4.2. Reconstrucció de senyals provinents d'extracció WPT	75
6.4.2.1. Funcions prèvies al pas d'arguments. fgetCoefitient i wfilters	75
6.4.2.2. Arguments per la reconstrucció de Wavelet packet transform	77
6.4.3. Valor de sortida	77
<b>7. Conclusions</b>	<b>78</b>
<b>8. Bibliografia</b>	<b>79</b>
<b>ANNEX A</b>	<b>80</b>
<b>ANNEX B</b>	<b>188</b>

## Resum de Treball Final de Carrera

E.T.Telecomunicacions, esp. Sistemes de Telecomunicació

Paraules clau: *LDB*, *MLDB*, *Wavelet packet transform*, *Discriminant Analysis*, *wavelet*, *processament de senyals*

Autora: Lídia Casadesús Martínez

Director: Pere Martí Puig

## 1. Resum

Tot seguit presentem un entorn per analitzar senyals de tot tipus amb LDB (*Local Discriminant Bases*) i MLDB (*Modified Local Discriminant Bases*).

Aquest entorn utilitza funcions desenvolupades en el marc d'una tesi en fase de desenvolupament. Per entendre part d'aquestes funcions es requereix un nivell de coneixement avançat de processament de senyals. S'han extret dels treballs realitzats per Naoki Saito [3], que s'han agafat com a punt de partida per la realització de l'algorisme de la tesi doctoral no finalitzada de Jose Antonio Soria.

Aquesta interfície desenvolupada accepta la incorporació de nous paquets i funcions. Hem deixat un menú preparat per integrar *Sinus IV packet transform* i *Cosine IV packet transform*, tot i que també podem incorporar-n'hi altres.

L'aplicació consta de dues interfícies, un Assistent i una interfície principal. També hem creat una finestra per importar i exportar les variables desitjades a diferents entorns.

Per fer aquesta aplicació s'han programat tots els elements de les finestres, en lloc d'utilitzar el GUIDE (*Graphical User Interface Development Enviroment*) de MATLAB, per tal que sigui compatible entre les diferents versions d'aquest programa.

En total hem fet 73 funcions en la interfície principal (d'aquestes, 10 pertanyen a la finestra d'importar i exportar) i 23 en la de l'Assistent. En aquest treball només explicarem 6 funcions i les 3 de creació d'aquestes interfícies per no fer-lo excessivament extens. Les funcions que explicarem són les més importants, ja sigui perquè s'utilitzen sovint, perquè, segons la complexitat McCabe, són les més complicades o perquè són necessàries pel processament del senyal.

Passem cada entrada de dades per part de l'usuari per funcions que ens detectaran errors en aquesta entrada, com eliminació de zeros o de caràcters que no siguin números, com comprovar que són enters o que estan dins dels límits màxims i mínims que li pertoqueu.

## **Abstract of final degree project**

E.T.Telecomunicacions, esp. Sistemes de Telecomunicació

Keywords: *LDB, MLDB, Wavelet packet transform, Discriminant Analysis, wavelet, signal processing*

Author: Lídia Casadesús Martínez

Director: Pere Martí Puig

We present an environment to analyze signals of different kind with LDB (*Local Discriminant Bases*) and MLDB (*Modified LDB*).

This environment uses functions developed within the framework of a thesis under development. To understand part of these functions it is required an advanced level of signal processing knowledge. They have been extracted from the work done by Naoki Saito [3], and have been taken as a starting point for the realization of different algorithms of the not finished doctoral thesis of Jose Antonio Soria.

This developed interface accepts the incorporation of new packages and functions. We have made a menu prepared for integrating *Sine IV packet transform* and *Cosine IV packet transform*, though we can also incorporate other ones.

The application consists of two interfaces, a Wizard and a main interface. Also we have created a window for importing and exporting the variables to different environments.

All the elements of the windows have been programmed, instead of using the GUIDE (*Graphical User Interface Development Environment*) of MATLAB, in order to preserve the compatibility among the different versions of this program.

Altogether we have made 73 functions for the main interface (10 of these are related with the of import and export window) and 23 for the Wizard. In this work we will explain only 6 of these functions and the 2 ones which creates these interfaces with the intention of not to make it overly extensive. The functions that we will explain are the most important, either because they are fairly often used, or for their complexity, according to the McCabe complexity, or because they are necessary for the signal processing.

We will check every data input user with functions to detect errors, as the elimination of zeros or other characters than numbers, such as checking they are integers or they are within the maximum and minimum limits in each case.

## 2. Introducció

Aquesta aplicació funciona a partir de les funcions externes de l'algorisme LDB i MLDB creades anteriorment. És una utilitat per facilitar l'entrada de dades a aquestes funcions, el control d'errors d'aquestes entrades i la visualització de senyals i resultats.

Per mitjà de l'algorisme LDB podem saber tant l'edat del peix com la diferenciació entre espècies, entre d'altres. Una altra utilitat és la detecció dels moviments del cos humà: si la persona està en decúbit, de peu o en moviment.

L'aplicació consta d'una primera interfície principal des de la qual podem fer diferents tipus d'extracció i reconstrucció de senyals i diferents tipus de visualització. Des d'aquesta interfície podem cridar una nova figura capaç d'importar o exportar variables a diferents entorns, com veurem en punts posteriors. Consta d'una segona interfície, un Assistent, o *Wizard*, que facilita el camí a l'usuari cap a l'extracció i la posterior reconstrucció de senyals per mitjà de diversos diccionaris wavelet.

Hem utilitzat una aplicació GUI (*interfície gràfica d'usuari*) programada en lloc d'utilitzar el GUIDE (GUI Builder) per tal que sigui compatible entre diferents versions de MATLAB, ja que hi ha incompatibilitats entre figures fetes en les versions noves i les antigues com, per exemple, entre les versions R2007a i posteriors i les R14. Per tant, hem optat per la programació de tota l'aplicació i la creació i el posicionament dels panells, els botons, els eixos, els menús i tots els elements que formen la interfície.

Per tal de poder utilitzar aquesta interfície serà necessari tenir instal·lada la Wavelet Toolbox.

## 3. Els otòlits

### 3.1. Què són

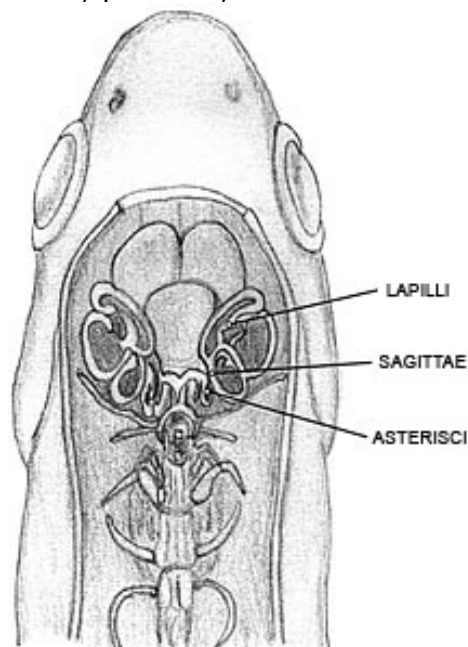
Els otòlits són unes formacions calcàries de l'oïda interna dels peixos teleostis (d'esquelet ossi) que s'utilitzen per saber l'edat del peix, les etapes del seu desenvolupament i les condicions de l'entorn on ha viscut.

Les diferents precipitacions de carbonat càlcic entorn d'un nucli inicial que conformen l'otòlit ens indiquen d'edat de l'individu, les condicions en les que es va desenvolupar, les condicions ambientals, si va sofrir algun dèficit nutritiu durant la seva vida, etc.

Avui en dia encara els "llegeixen" experts amb una llarga experiència i coneixement. Encara que això no deixa de ser subjectiu i comporta un marge d'error considerable.

Tots els vertebrats tenen estructures amb les mateixes característiques que els otòlits (otocònies) però en els mamífers han quedat reduïdes a petits cristalls.

Les seves funcions són estabilitzar la posició de l'individu (percebre l'acceleració angular i la gravetat) i rebre sons (funció acústica). Existeixen tres parells d'òrgans d'otòlits: *sagittae*, *lapilli* i *asterisci*. Estan associats al sàcul, l'utricle i la lagena respectivament. El *sagitta* és el quin morfològicament canvia més i, per tant, és el més estudiat.



**Il·lustració 3.1** Localització dels otòlits en l'aparell vestibular

### 3.2. Estudis

Tot i que Aristòtil (s. IV aC) va ser el primer en mencionar la seva existència i el folklore dels temps s'ha anat referint als otòlits no va ser fins al segle XIX quan es van començar a estudiar. Els naturalistes més rellevants van ser Georges Cuvier, que va començar a identificar-los, i E. Koken, que va començar a interpretar-los (el seu primer treball és de 1884, i va generar un punt d'inflexió per la seva detallada obra).

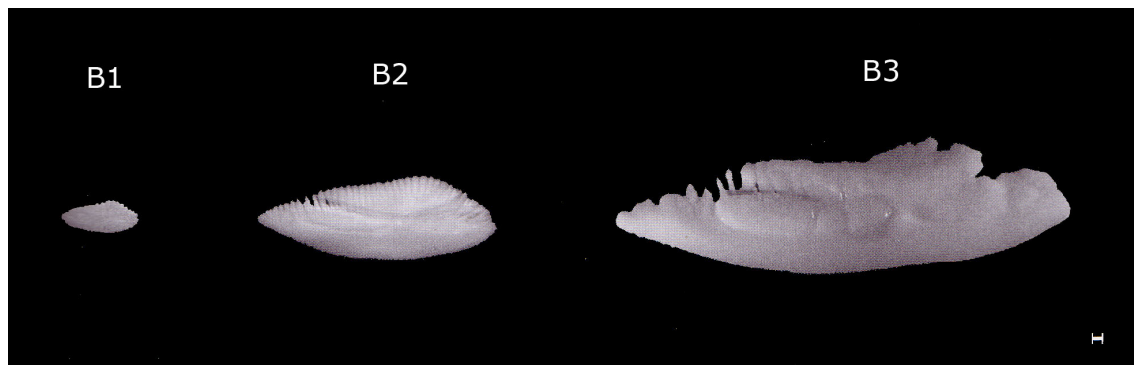
Se'n poden estudiar algunes matèries com descripcions de noves espècies, diferenciació dels diferents grups d'organismes específics i les seves relacions evolutives, existències d'espècies i població, morfologia i funcionalitat de l'oïda del peix, relacions entre la seva mida durant el creixement i analogies entre la mida dels otòlits recents i els otòlits fòssils.

### 3.3. Presa de les imatges

Els primers estudis es basaven en descripcions amb dibuixos en blanc i negre.

Actualment, per prendre les imatges se sol utilitzar una càmera sota un microscopi. La imatge es pren per la cara interna de l'otòlit, per on presenta el *sacculus acusticus* (un solc longitudinal a la superfície de la *sagitta*). Per obtenir una bona representació del contorn la imatge ha d'estar ben contrastada amb un fons negre. Els otòlits sempre es representen amb el marge dorsal respecte al capdamunt de la imatge i la regió anterior (rostral) a la dreta [3].

Per extreure'n el contorn es binaritza la imatge a dos nivells de color i es fa una detecció de marges cercant un canvi bruscat de gradient [4].



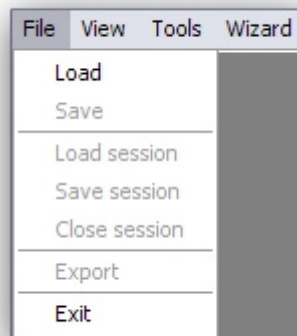
**Il·lustració 3.2 Otòlits de lluç (*Merluccius merluccius*)**  
**Longituds dels peixos: B1: 15.4 cm, B2: 53.0cm. B3: 98.8cm.**



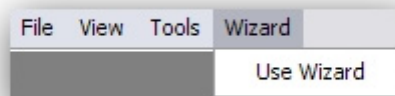
## 4. Presentació del funcionament de l'aplicació gràfica desenvolupada

En aquest capítol analitzarem el funcionament de l'Assistent i del programa principal, els seus menús, la importació de variables, la representació del senyal, els mètodes d'extracció i reconstrucció i de l'exportació de variables.

En primer lloc podem veure que el menú *File* de la interfície principal només té activades l'opció *Load* i *Exit*. Les opcions *Save*, *Close session* i *Export* només s'activaran un cop haguem carregat les variables que contenen la informació del senyal o una sessió. L'opció *Load session* s'activarà només si tenim una sessió arxivada al directori d'usuari per defecte de MATLAB. El mateix passa amb les opcions dels menús *View* i *Tools*, només s'activaran quan haguem carregat les variables corresponents o una sessió. En el cas del menú *Use Wizard* el tenim activat ja que des d'un principi podem utilitzar aquest Assistent.



**Il·lustració 4.1 Menú *File*. No hem carregat cap variable i hi tenim opcions inhabilitades**



**Il·lustració 4.2 Menú *Wizard*. Està habilitat des de l'inici de l'aplicació**

Podem començar utilitzant l'Assistent per extreure i reconstruir els coeficients d'un senyal.

### 4.1. Funcionament de l'Assistent

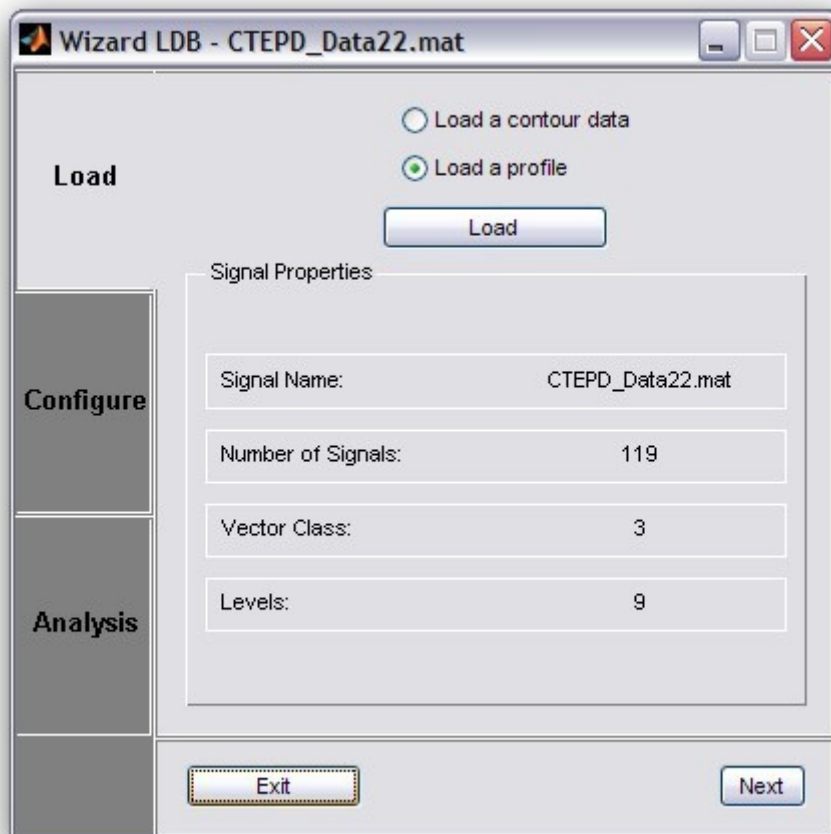
A continuació explicarem el funcionament de la interfície d'Assistent.

Només podrem executar l'Assistent per mitjà de la interfície principal, ja que necessita les seves dades per treballar. La comanda que escriurem és el nom de la funció de creació de la seva interfície, que és *LDB\_Wizard*.

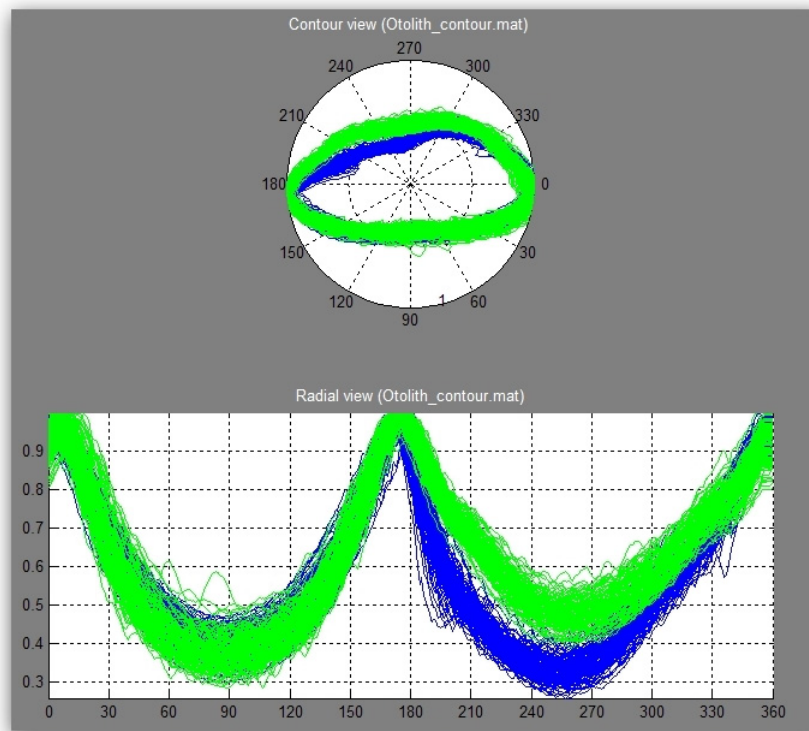
A través del menú *Use Wizard* obrirem l'Assistent. Inicialment només podrem sortir o carregar un fitxer i triar si volem visualitzar-lo en contorn o en perfil o bé sortir de la interfície.

Ens mostrarà una taula amb el nom del senyal, el número de senyals, el número de classes que conté i el nivell màxim de descomposició. Afegirà el nom del senyal al títol de l'Assistent i netejarà la interfície principal.

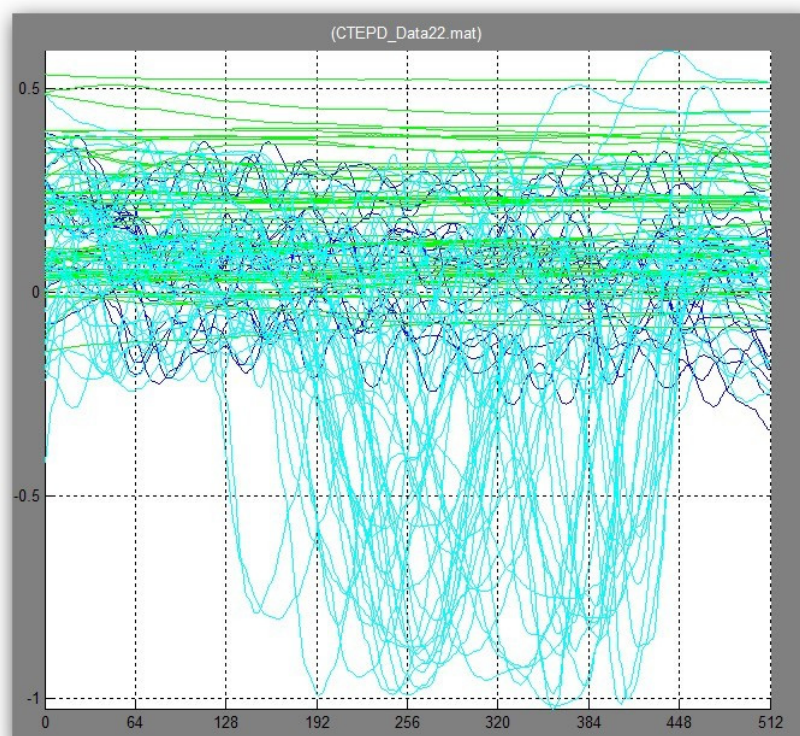
Tenint activat el botó radio "Load a contour data" ens dibuixarà el senyal en contorn i en perfil a la interfície principal. Podem activar el botó radio "Load a profile" per tal de visualitzar-lo en perfil. En aquest moment ens quedarà activat el botó *Next*.



**Il·lustració 4.3** Carreguem un senyal a l'Assistent. Les caixes de text contindran les seves dades



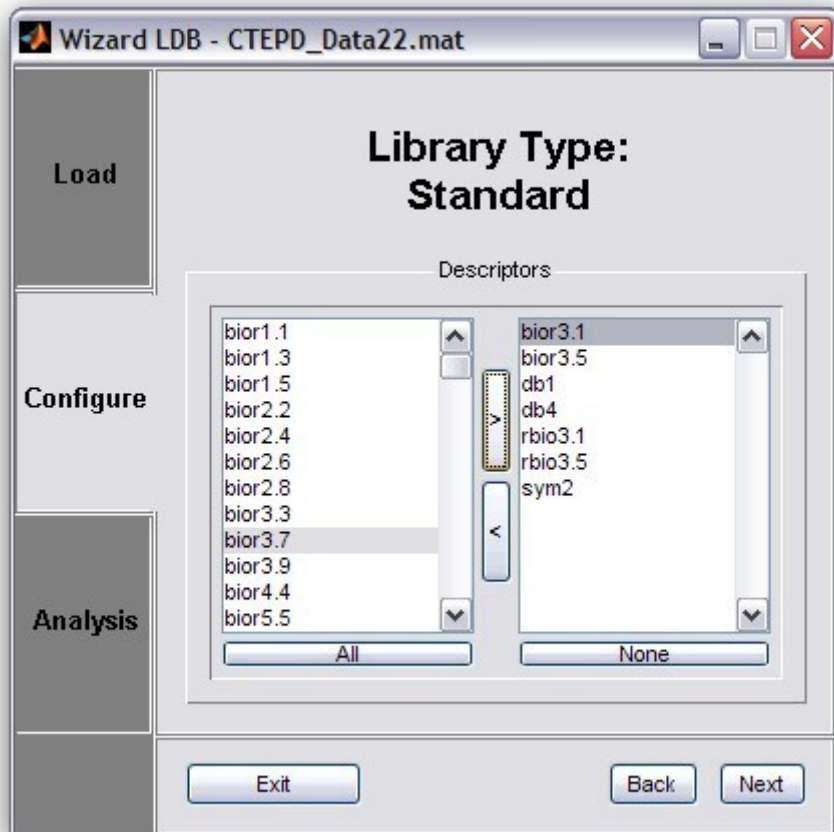
**Il·lustració 4.4 Visualització en contorn i en perfil d'un senyal carregat per mitjà de l'Assistent**



**Il·lustració 4.5 Visualització en perfil d'un senyal carregat per mitjà de l'Assistent**

Premem el botó *Next* per continuar amb l'aplicació.

Ens trobem un panell on se'ns demana els diccionaris amb els que volem fer l'extracció. Els seleccionarem passant-los a la llista del costat. Podem incloure'ls d'un en un o tots a la vegada. En el nostre exemple n'hi afegirem 7: *bior3.1*, *bior3.5*, *db1*, *db4*, *rbio3.1*, *rbio3.5* i *sym2*.



**Il·lustració 4.6 Configuració de l'Assistent. Incloem els diccionaris amb els que volem fer l'extracció**

Premem el botó *Next* per processar-ho. L'aplicació ens inhabilitarà tots els botons per evitar la duplictat de comandes.

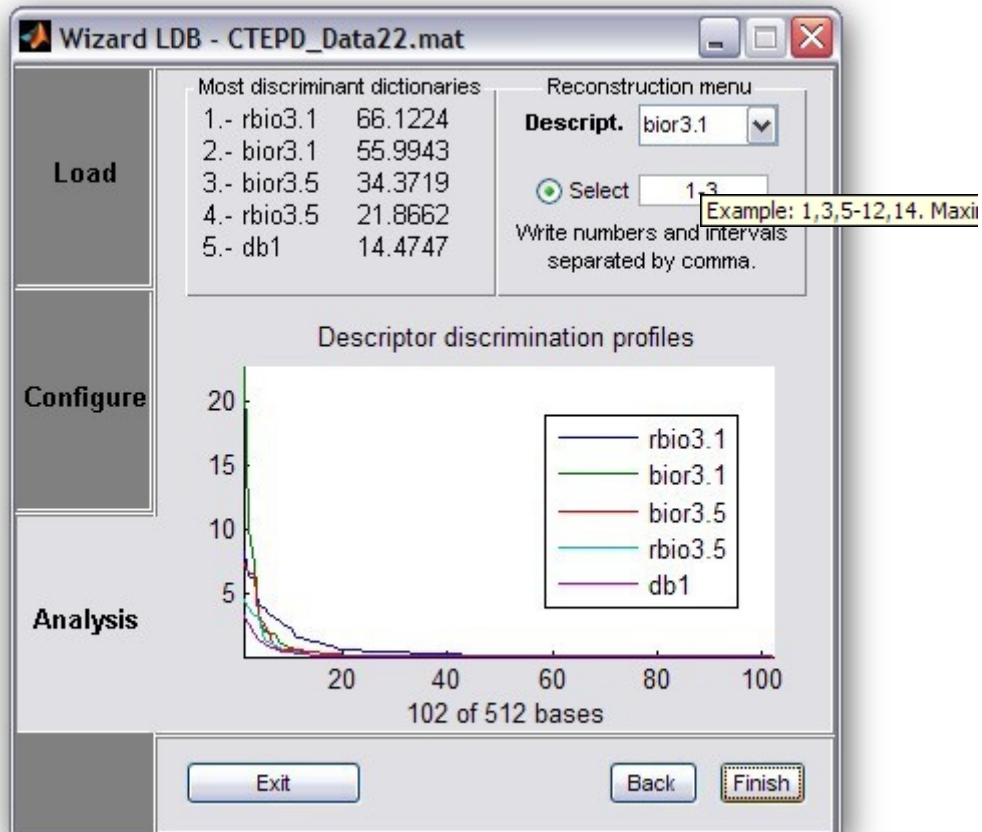
Una vegada processat ens mostrarà el panell d'anàlisi. Aquest panell conté el rànquing dels cinc diccionaris més discriminants, un eix amb la representació de la discriminació d'aquests cinc diccionaris respecte la longitud del senyal (mostrarem el 20% de les mostres) i un panell en el qual hem de triar el diccionari amb el que volem reconstruir i l'interval de reconstrucció.

En aquest punt, el botó *Next* assigna a la propietat *TooltipString* de la caixa de text d'introducció de l'interval un text d'exemple i el rang màxim que podem reconstruir (el nombre de mostres):

```
set(wiz.editSelect, 'ToolTipString', ...
    ['Example: 1,3,5-12,14. Maximum range: 1-' ...
    num2str(size(wiz.file.Bspaces,1))]);
```

En el cas del senyal que estem seguint el text complet que mostrarà la propietat *ToolTipString* que hi passem el ratolí per sobre serà:

“Example: 1,3,5-12,14. Maximum range: 1-512”.

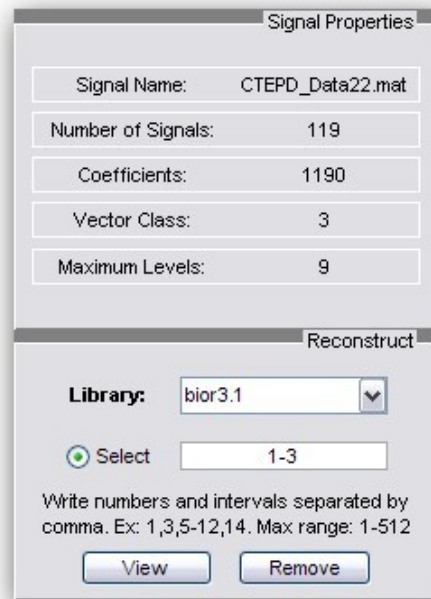


**Il·lustració 4.7 Panell final de l'Assistent. Hem de triar el diccionari amb el qual volem reconstruir i l'interval de reconstrucció**

Un cop haguem clicat el botó *Finish*, l'Assistent reconstruirà el senyal a partir dels seus coeficients en l'interval que li haguem especificat.

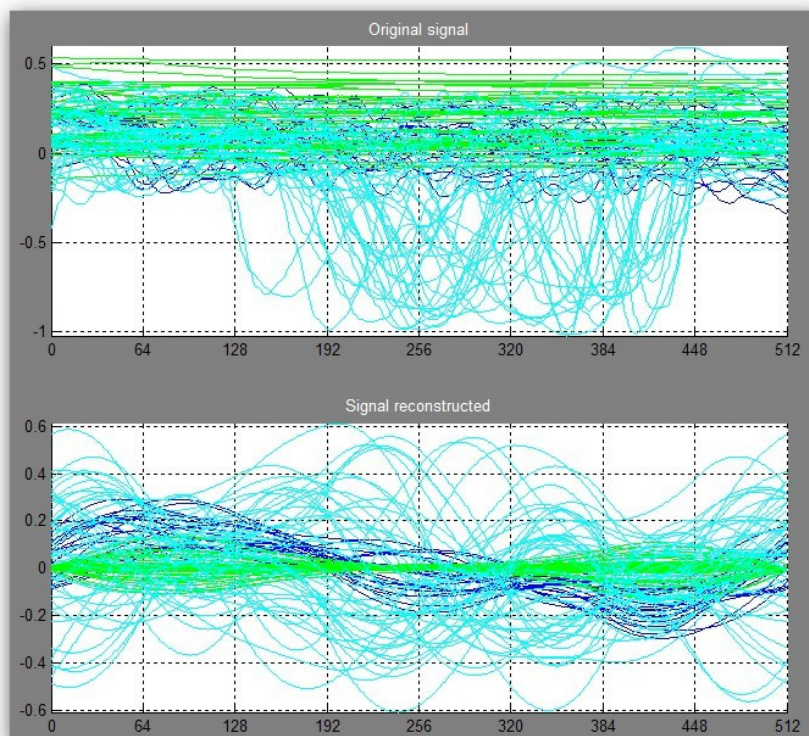
Desarà els resultats obtinguts i el senyal carregat a la variable global *FileGlobal*, ja que l'Assistent els manté en un camp de la seva estructura de handles per no sobre escriure les dades generades anteriorment. D'aquesta manera si sortim de l'Assistent (per mitjà del botó *Exit*, tancant-la amb el menú contextual o amb la creu vermella de tancar) mantenim les variables anteriors.

Posarà les dades del senyal al panell *Signal properties* i ajustarà el panell de reconstrucció d'acord amb els diccionaris seleccionats, l'interval reconstruït i el seu text d'exemple amb l'interval màxim.



**Il·lustració 4.8 Panells de propietats del senyal i de reconstrucció modificats per l'Assistent**

Finalment dibuixarà a la interfície principal el senyal original i el senyal reconstruït. En cas que haguem triat l'opció "Load a contour data" a l'hora de carregar el fitxer també dibuixarà el contorn dels senyals:



**Il·lustració 4.9 Visualització de la reconstrucció d'un senyal de perfils des de l'Assistent**

En aquest punt podem comprovar que els menús modifiquen les seves propietats per indicar que tenim les variables del senyal en memòria. Per tant, el menú *File* ens permetrà sobre escriure les variables (*Save*), desar la sessió (*Save session*) i tancar-la (*Close session*) i exportar les variables generades (*Export*); la resta de menús també activaran les seves opcions.

## 4.2. Funcionament de la interfície principal

A continuació explicarem el funcionament de la interfície principal. Començarem pels menús i la funció de cadascun. Seguidament exposarem el funcionament de la importació de variables. En el següent punt comentarem la representació del senyal dividit en classes. A continuació explicarem les extraccions i reconstruccions de *Wavelet packet transform*, *Discriminant Analysis Standard (LDB)* i *Discriminant Analysis Custom (MLDB)*. Finalment mostrarem la interfície d'exportació de variables.

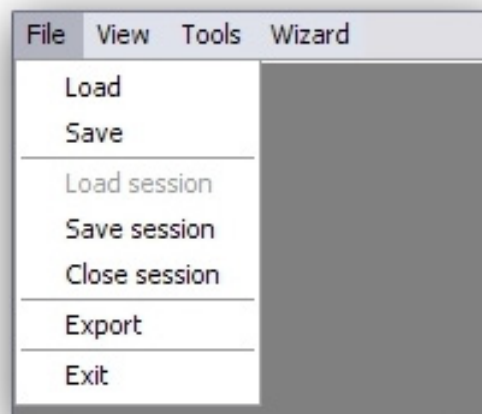
Executarem la interfície principal per mitjà de l'àrea de treball de MATLAB. La comanda que escriurem és el nom de la seva funció de creació, que és *LDB\_GUI*.

### 4.2.1. Menús

El menú *File* de la interfície principal només habilitarà els submenús *Save*, *Close session* i *Export* un cop haguem carregat les variables que contenen la informació del senyal o una sessió. El submenú *Load session* s'habilitarà només si tenim una sessió arxivada al directori d'usuari per defecte de MATLAB. El mateix passa amb les opcions dels menús *View* i *Tools*, només s'activaran quan haguem carregat les variables corresponents o una sessió. El menú *Use Wizard* el tenim activat ja des d'un principi.

#### Menú *File*

Aquest menú conté 7 submenús:



**Il·lustració 4.10** Submenús del menú *File*

El submenú *Load* obrirà la interfície d'importar, que ens permetrà triar les variables que volem importar d'un fitxer o de l'àrea de treball.

*Save* desarà les variables generades durant la sessió al mateix fitxer que hem obert, en cas d'haver carregat un fitxer, o a l'àrea de treball, si hem carregat les variables des de l'àrea de treball.

*Load session* obrirà la sessió que haguem desat prèviament al directori d'usuari de MATLAB.

El submenú *Save session* desarà les característiques actuals de l'aplicació, els panells que hi ha oberts, els menús habilitats, els objectes visibles, el fitxer carregat... en un fitxer .MAT al directori MATLAB de l'usuari per defecte. En cas que ja hi tinguem una sessió desada la sobreescriu.

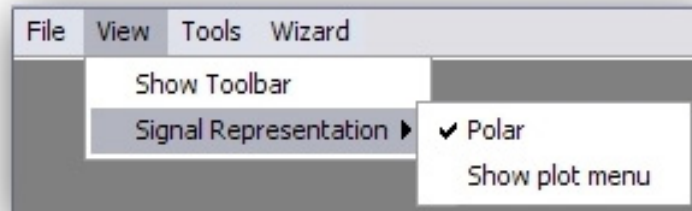
*Close session* inicialitza la interfície, deixant-la completament neta de dades i esborrant les variables referents al senyal i les generades.

*Export* obrirà la interfície d'exportar, que ens permet escollir les variables que volem desar i triar si volem fer-ho en un fitxer .MAT o a l'àrea de treball.

Amb el submenú *Exit* sortirem de l'aplicació i quedaran netes les variables globals.

### Menú *View*

*View* conté dos submenús:



**Il·lustració 4.11 Submenús del menú *View***

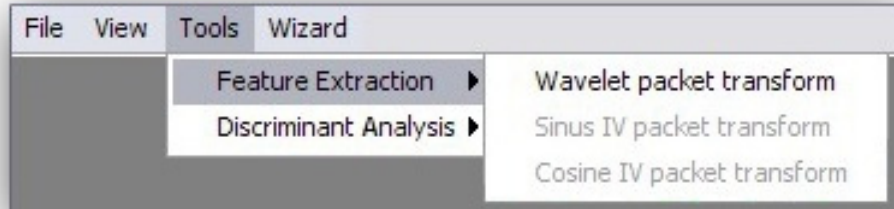
El submenú *Show Toolbar* afegirà una barra d'eines de figura a la interfície. Quan hi tinguem la barra d'eines el nom del submenú canviarà a *Hide Toolbar* i, si hi cliquem, amagarà aquesta barra. En aquesta barra d'eines n'hem amagat algunes opcions no necessàries per la representació de la figura, com rotar en 3D.

El menú *Signal representation* conté dos submenús: *Polar*, que alternarà la visualització dels eixos a polars, quan estigui desmarcada, o a cartesianes, quan estigui marcada, i *Show plot menu*, que ens mostrarà el panell de visualització per classes i canviarà el seu nom a *Hide plot menu*, que l'amagarà.



### Menú *Tools*

Aquest menú conté dos submenús:

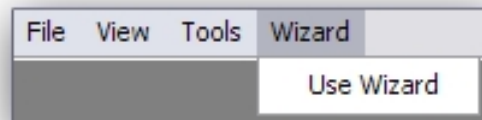


*Feature Extraction*, que conté els menús *Wavelet packet transform*, que obrirà el panell corresponent, *Sinus IV packet transform* i *Cosine IV packet transform*. Aquests dos últims estan inhabilitats, ja que només indiquen la possibilitat d'afegir aquestes transformacions en futures versions de l'aplicació.

I el submenú *Discriminant Analysis* conté el submenú *LDB*, i obre el panell per fer l'extracció amb diccionaris wavelet o amb filtres paraunitaris personalitzats.

### Menú *Wizard*

Aquest menú conté un submenú anomenat *Use Wizard* que obrirà l'Assistent.



**Il·lustració 4.12** Submenú del menú *Wizard*

## 4.2.2. Importació de variables

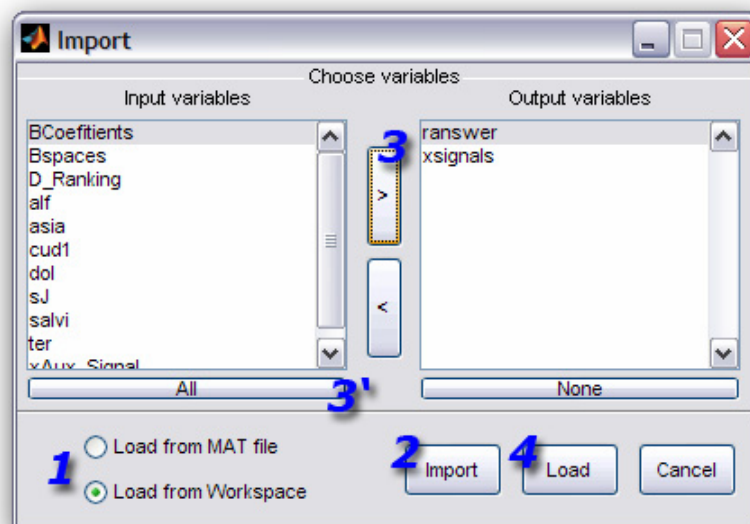
Per mitjà del menú *Load* obrim una nova finestra en la qual podem triar les variables que volem importar i si les volem d'un fitxer o de l'àrea de treball de MATLAB.

El botó *Import* mostrarà les variables contingudes al fitxer seleccionat o de l'àrea de treball, segons el botó radio que haguem marcat: si volem veure les variables que tenim disponibles a l'àrea de treball marcarem el botó radio "Load from Workspace"; si les quines volem veure estan dins d'un fitxer .MAT tindrem marcat el botó radio "Load from MAT file". Finalment clicarem el botó *Import*, que importarà els noms de les variables de l'àrea de treball

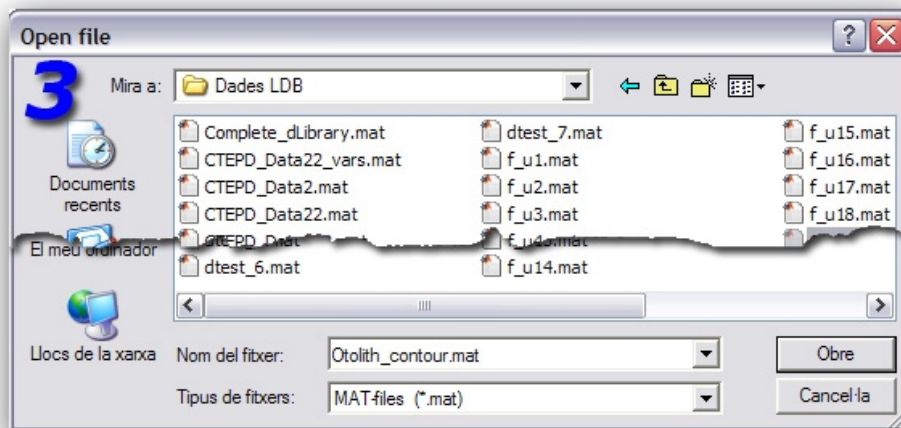
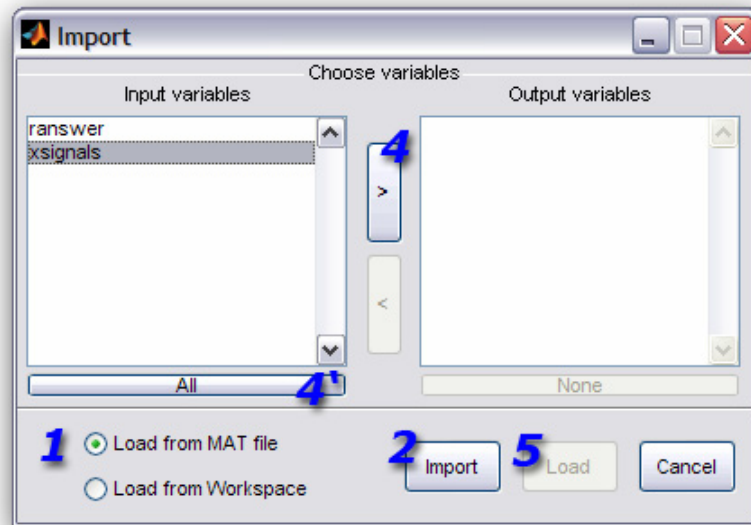
directament (il·lustració 4.13) o obrirà una finestra de selecció de fitxer (il·lustració 4.14).

A continuació escollirem les variables que volem carregar posant-les a la llista de sortida per mitjà dels botons >, que les afegirà individualment, i *All*, que les afegirà totes alhora. Un cop seleccionades podem modificar la selecció que hem fet traient-les d'una en una (<) o totes a la vegada (*None*). Mentre no tinguem cap variable seleccionada els botons <, *None* i *Load* estaran inhabilitats.

Un cop seleccionades les variables que volem importar les carregarem clicant el botó *Load*.



**Il·lustració 4.13 Importem els noms de les variables de l'àrea de treball**



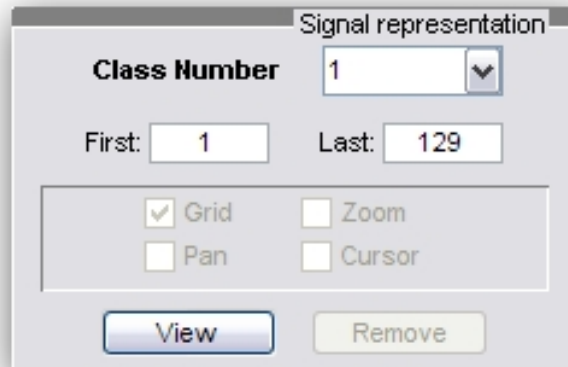
#### Il·lustració 4.14 Importem els noms de les variables contingudes en un fitxer

Amb les variables ja carregades, la interfície modificarà els seus panells, menús i altres elements per adaptar-se a la variable carregada.

### 4.2.3. Representació del senyal

Quan carreguem un fitxer o quan premem el submenú *Show plot menu* del menú *View* l'aplicació obrirà un panell per representar el senyal dividit en classes.

Per mitjà d'aquest panell podrem visualitzar una classe sencera o un nombre determinat de senyals d'aquesta classe. Tots els senyals d'una mateixa cada classe quedaran dibuixats amb un mateix color: blau, verd, cian, magenta, groc o negre. Exclourà el color blanc, ja que el fons dels eixos és blanc i no distingiríem el senyal, i el vermell, ja que en futures versions s'utilitzarà per marcar la part discriminant del senyal reconstruït.



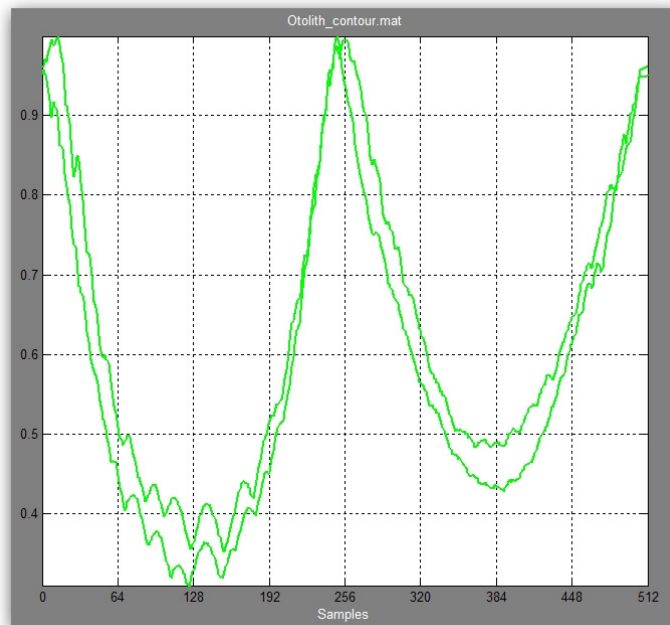
#### Il·lustració 4.15 Panell de representació de senyals agrupats en classes

Amb el menú emergent *Class Number* seleccionarem una de les classes disponibles. Inicialment ens mostra tots els senyals d'una classe a les caixes de text *First* i *Last*, indicant el primer senyal i l'últim pertanyents a la classe. Podem reduir el nombre de senyals que dibuixarem i escollir els quins volem visualitzar.

En aquest panell hi trobem unes caselles de verificació que podem activar un cop visualitzat el fitxer. Amb la casella de verificació "Grid" inclourem una quadrícula als eixos. Aquesta pot estar activada alhora que les altres caselles de verificació. Activant el "Zoom" podrem acostar-los i allunyar-los, amb el "Pan" desplaçarem el senyal representat amb el ratolí a través dels eixos i amb la casella "Cursor" ens hi posarà les coordenades quan cliquem en un punt. D'aquestes tres només podem tenir-ne una d'activada, ja que cadascuna d'elles tindrà el seu menú contextual associat.

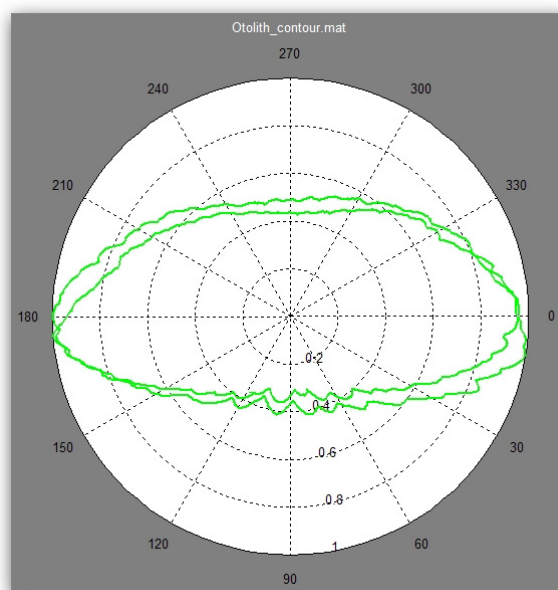
Clicant el botó *View* dibuixarà el conjunt de senyals seleccionat en les caixes de text i habilitarà les caselles de verificació i el botó *Remove*, que esborrarà la imatge i ens mostrarà tots els senyals de totes les classes. Dibuixarà els senyals en forma de contorn o en perfil en funció de la selecció que hi hagi feta al menú *Polar*.

A tall d'exemple, podem visualitzar dos senyals de la segona classe, el setè i el vuitè: escollirem la classe 2 amb el menú desplegable i posarem els valors a les caixes de text corresponents (la segona classe té numerats els senyals del 130 al 258, per tant serà *First: 136* i *Last: 137*) i premerem el botó *View*. Si no tenim marcat el menú *Polar* obtindrem la imatge següent:



**Il·lustració 4.16 Visualització en perfil dels senyals 7 i 8 de la segona classe**

En cas de tenir el menú *Polar* marcat, la imatge que obtindrem serà:



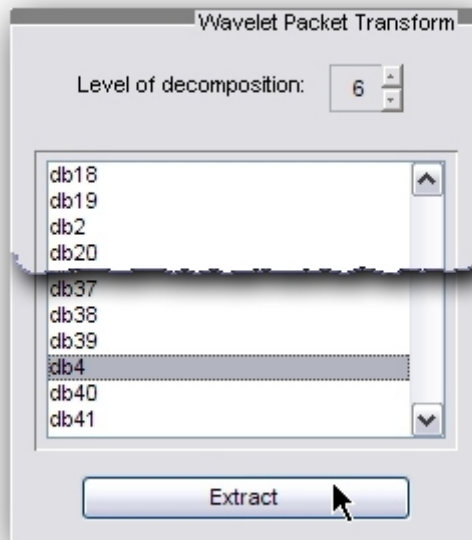
**Il·lustració 4.17 Visualització en contorn dels senyals 7 i 8 de la segona classe**

Podem també activar la barra d'eines per retocar les imatges o desar la finestra.

#### 4.2.4. Wavelet Packet Transform i la seva reconstrucció

Per mitjà del menú *Feature Extraction*, obrirem el panell *Wavelet packet transform*.

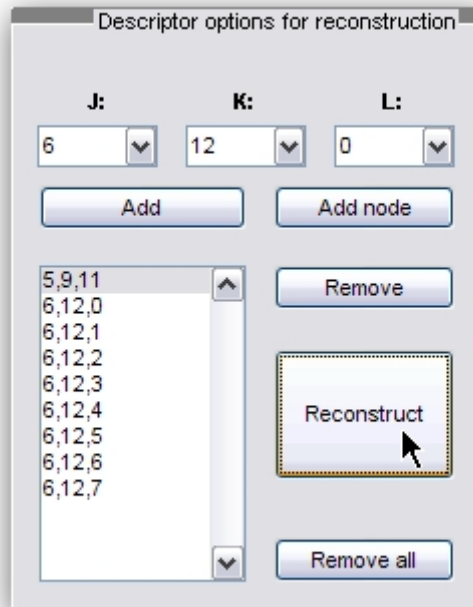
En aquest panell triarem el nivell de descomposició i el diccionari wavelet amb el que volem processar. Un cop fet extraïem els coeficients amb el botó *Export*.



**Il·lustració 4.18 Panell *Wavelet packet transform*. Escollim el nivell de descomposició i el diccionari wavelet**

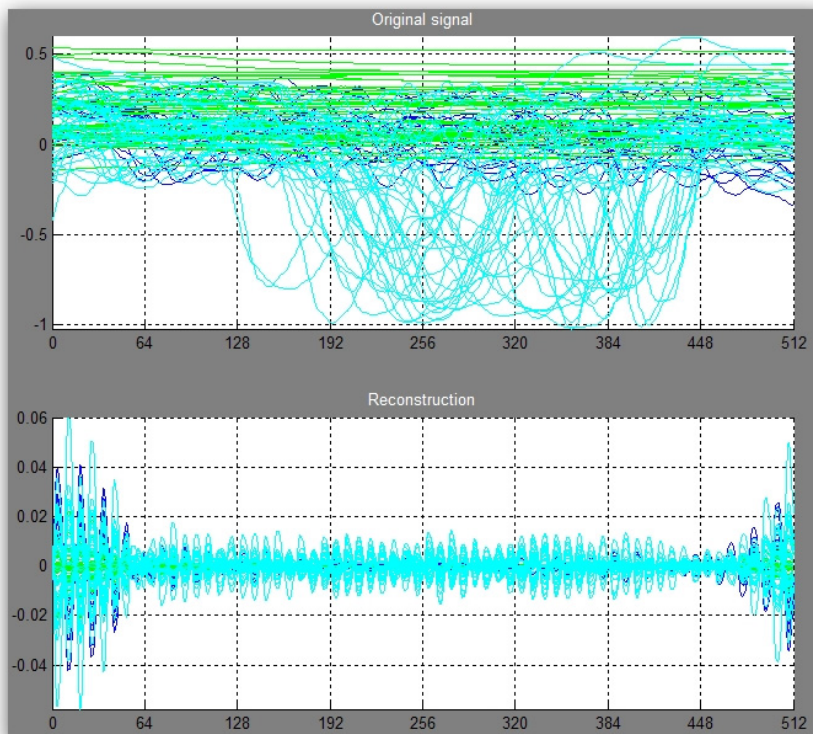
Quan l'aplicació hagi acabat de processar ens mostrarà un panell nou en el que triarem l'espai de reconstrucció. Tindrà un botó per afegir a la llista, d'un en un, els  $j$ ,  $k$ ,  $l$  seleccionats en els menús desplegable i un altre per afegir-hi el node sencer  $(j,k)$ . En té dos més per esborrar un sol node o per esborrar tot el que tenim a la llista.

Un cop escollits premem el botó *Reconstruct*.



**Il·lustració 4.19 Panell de reconstrucció WPT. Reconstrucció de 5,9,11 i del node 6,12**

Finalment dibuixarà el senyal original i el resultat de la reconstrucció. Si tenim la opció *Polar* activada dibuixarà també el senyal en contorns.



**Il·lustració 4.20 Dibuix del senyal original i el reconstruït per WPT. Node 6,12 i 5,9,11**

#### 4.2.5. Discriminant Analysis (LDB) Standard i la seva reconstrucció

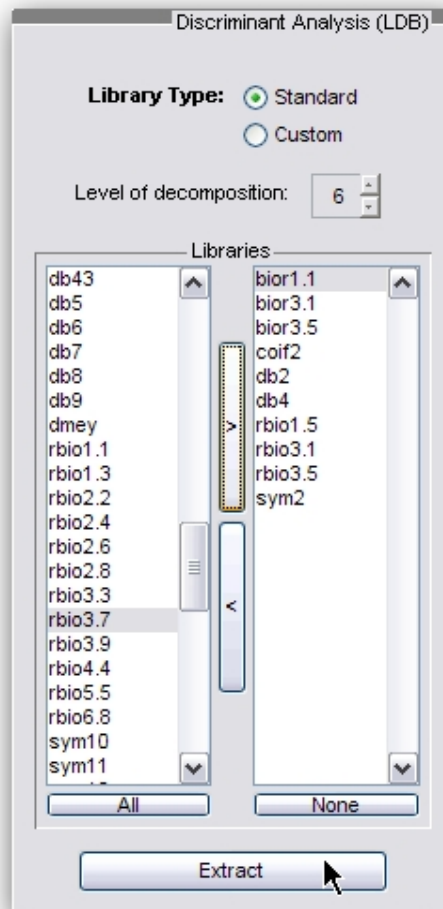
Per mitjà del menú *LDB*, de *Discriminant Analysis*, obrirem el panell *Discriminant analysis (LDB)*.

Per fer una extracció en LDB Standard haurem de tenir activat el botó radio *Library type: Standard*.

En primer lloc triarem el nivell de descomposició que volem en la caixa de text *Level of decomposition*.

A continuació escollirem els diccionaris que volem processar posant-los a la llista de sortida per mitjà dels botons *>*, que les afegirà individualment, i *All*, que afegirà tota la llibreria. Un cop seleccionats podem modificar la selecció que hem fet traient-los d'un en un (*<*) o tots a la vegada (*None*). Mentre no tinguem cap diccionari seleccionat els botons *<*, *None* i *Load* estaran inhabilitats.

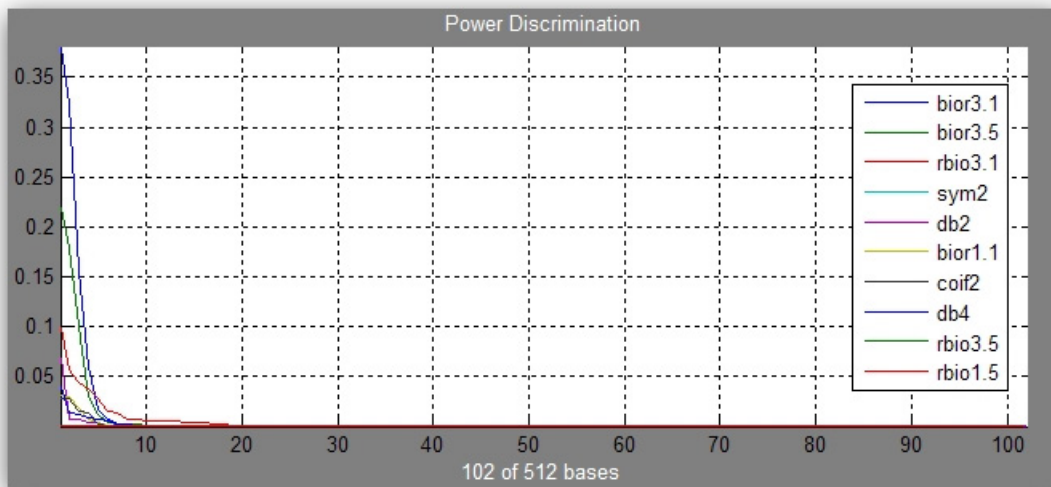
Un cop escollides premem el botó *Extract*.



**Il·lustració 4.21 Panell *Discriminant Analysis (LDB)*. Extracció *Standard*. Escollim el nivell de descomposició i els diccionaris wavelet**



Ens dibuixarà el poder de discriminació de els diccionaris. Si hem processat 10 o menys diccionaris ens mostrarà una llegenda amb el seu nom:



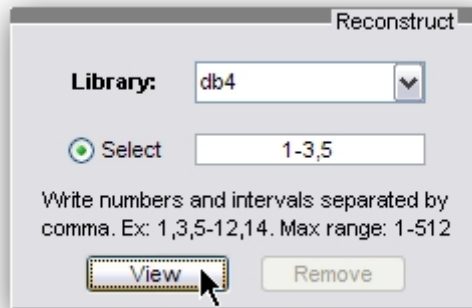
**Il·lustració 4.22** Dibuix amb el poder discriminant de cada diccionari extret amb *WPT*

Ens mostrarà també una taula amb el rànquing de poder discriminant dels diccionaris processats. Aquesta taula està feta a base de caixes de text, enlloc del control *uitable*, per evitar incompatibilitats entre diferents versions de MATLAB. Hi trobem una casella de verificació al costat de cada diccionari per poder comprovar la posició del node òptim:

	Library	Power Discrim.	<input type="checkbox"/>	j,k,l
1	bior3.1	0.954658	<input type="checkbox"/>	
2	bior3.5	0.556774	<input type="checkbox"/>	
3	rbio3.1	0.384148	<input type="checkbox"/>	
4	sym2	0.0964771	<input type="checkbox"/>	
5	db2	0.0964771	<input type="checkbox"/>	
6	bior1.1	0.0958189	<input type="checkbox"/>	
7	coif2	0.0954953	<input type="checkbox"/>	
8	db4	0.0948539	<input checked="" type="checkbox"/>	7,1,0
9	rbio3.5	0.0938327	<input type="checkbox"/>	
10	rbio1.5	0.0937683	<input type="checkbox"/>	

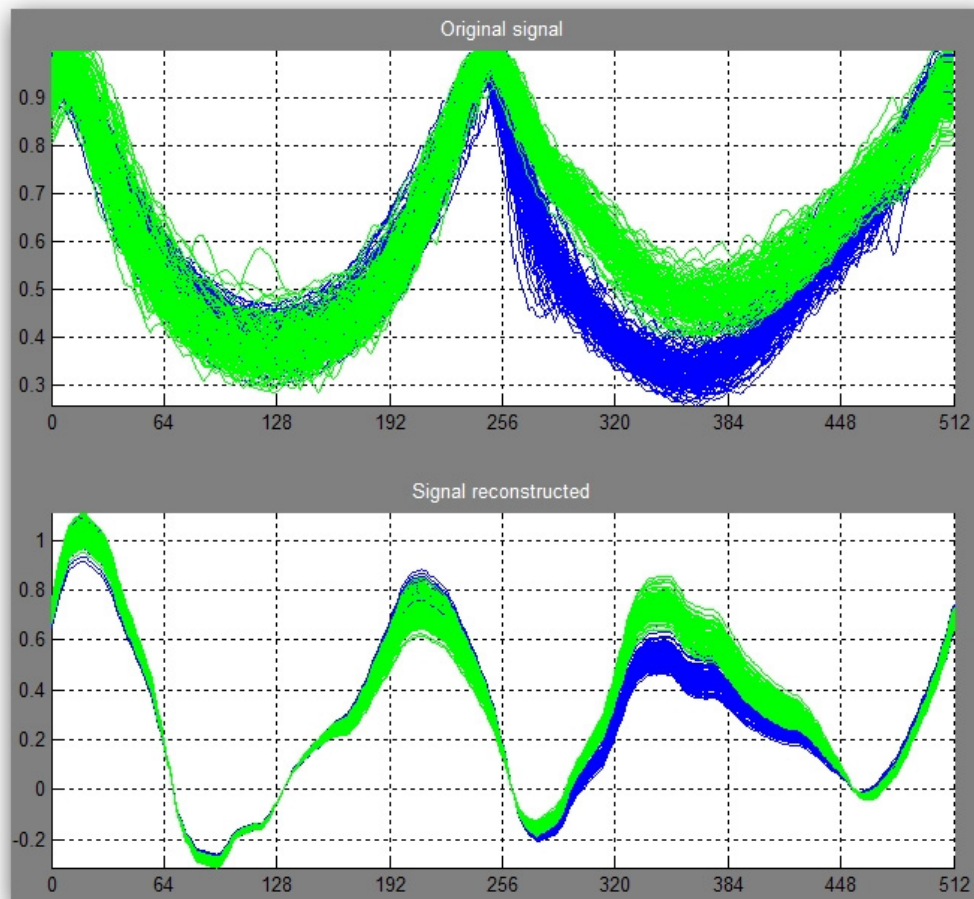
**Il·lustració 4.23** Taula amb el poder discriminant de cada diccionari i posició del node òptim del diccionari *db4*

A continuació ens mostra el panell de reconstrucció. En aquest panell escollirem el diccionari per mitjà del menú desplegable *Library* i seleccionarem l'espai que volem reconstruir a la caixa de text *Select*. En aquesta caixa de text posarem intervals, separats per guió, i números separats per coma.



**Il·lustració 4.24 Panell de reconstrucció *Discriminant analysis*. Reconstrucció de l'interval 1-3,5 amb el diccionari *db4***

Tot seguit premerem el botó *View* per fer la reconstrucció. Finalment dibuixarà el senyal original i el resultat de la reconstrucció. Si tenim la opció *Polar* activada dibuixarà també el senyal en contorns.



**Il·lustració 4.25 Dibuix del senyal original i el reconstruït per *Discriminant Analysis* del diccionari *db4***

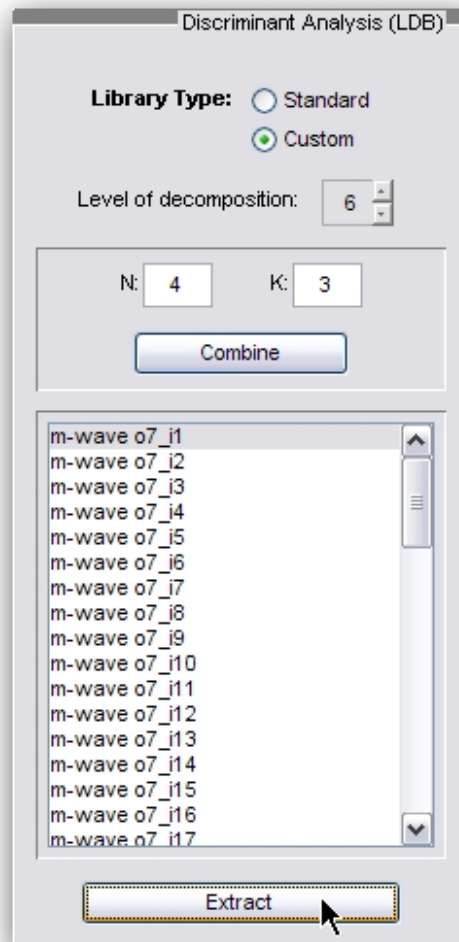
#### 4.2.6. Discriminant Analysis (LDB) Custom i la seva reconstrucció

Per mitjà del menú *LDB*, de *Discriminant Analysis*, obrirem el panell *Discriminant analysis (LDB)*.

Per fer una extracció en LDB Custom haurem de tenir activat el botó radio *Library type: Custom*.

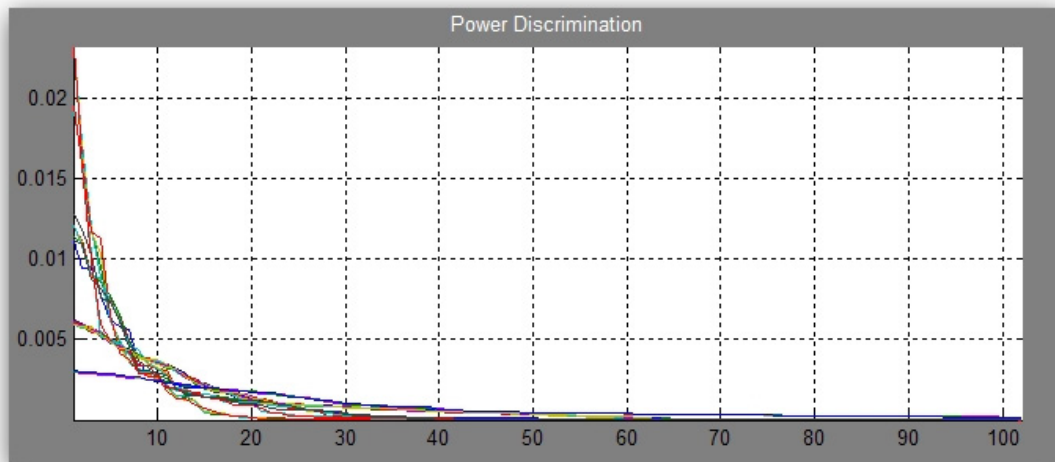
En primer lloc triarem el nivell de descomposició que volem en la caixa de text *Level of decomposition*.

Tot seguit introduïm el mostreig de l'interval  $[0, \pi)$  a la caixa de text *N* i el número de paràmetres del filtre menys  $u$  a la caixa *K*, i premerem el botó *Combine* per generar els filtres paraunitaris. Un cop generats clicarem el botó *Extract*. No el podem clicar si no estan els filtres generats.



**Il·lustració 4.26 Panell discriminant Analysis. Fem l'extracció amb els filtres generats amb  $N=4$  i  $K=3$**

Ens dibuixarà el poder de discriminació dels diccionaris. Si hem processat 10 o menys diccionaris ens mostrarà una llegenda amb el seu nom.



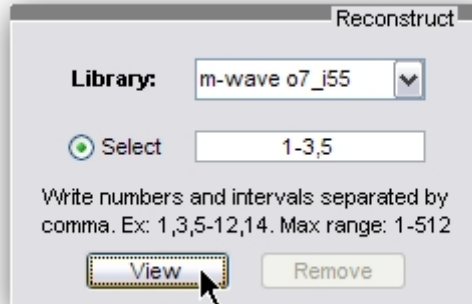
**Il·lustració 4.27** Dibuix amb el poder discriminant de cada diccionari extret amb *Discriminant analysis*

Ens mostrarà també una taula feta a base de caixes de text, enlloc del control *uitable*, per evitar incompatibilitats entre diferents versions de MATLAB. Si la taula conté més de 13 diccionaris hi afegirà una barra de desplaçament vertical. Com que el nom dels filtres poden ser més grans que les caixes de text, la taula es farà més ampla en cas que la mida del nom superi la de la caixa. En aquesta taula hi podrem comprovar el rànquing de poder discriminant dels filtres processats i la posició del node òptim del filtre marcat amb la casella de verificació:

	Library	Power Discrim.		j,k,l
1	m-wave o7_j24	0.0943902	<input type="checkbox"/>	
2	m-wave o7_i4	0.0943817	<input type="checkbox"/>	
3	m-wave o7_j61	0.0943042	<input type="checkbox"/>	
4	m-wave o7_j25	0.0943004	<input type="checkbox"/>	
5	m-wave o7_i10	0.0943004	<input type="checkbox"/>	
6	m-wave o7_j55	0.0943004	<input checked="" type="checkbox"/>	6,1,7
7	m-wave o7_j44	0.0943004	<input type="checkbox"/>	
8	m-wave o7_j59	0.0943004	<input type="checkbox"/>	
9	m-wave o7_j63	0.0943004	<input type="checkbox"/>	
10	m-wave o7_j51	0.0943004	<input type="checkbox"/>	
11	m-wave o7_i1	0.094267	<input type="checkbox"/>	
12	m-wave o7_j53	0.0942662	<input type="checkbox"/>	
13	m-wave o7_j34	0.0942357	<input type="checkbox"/>	

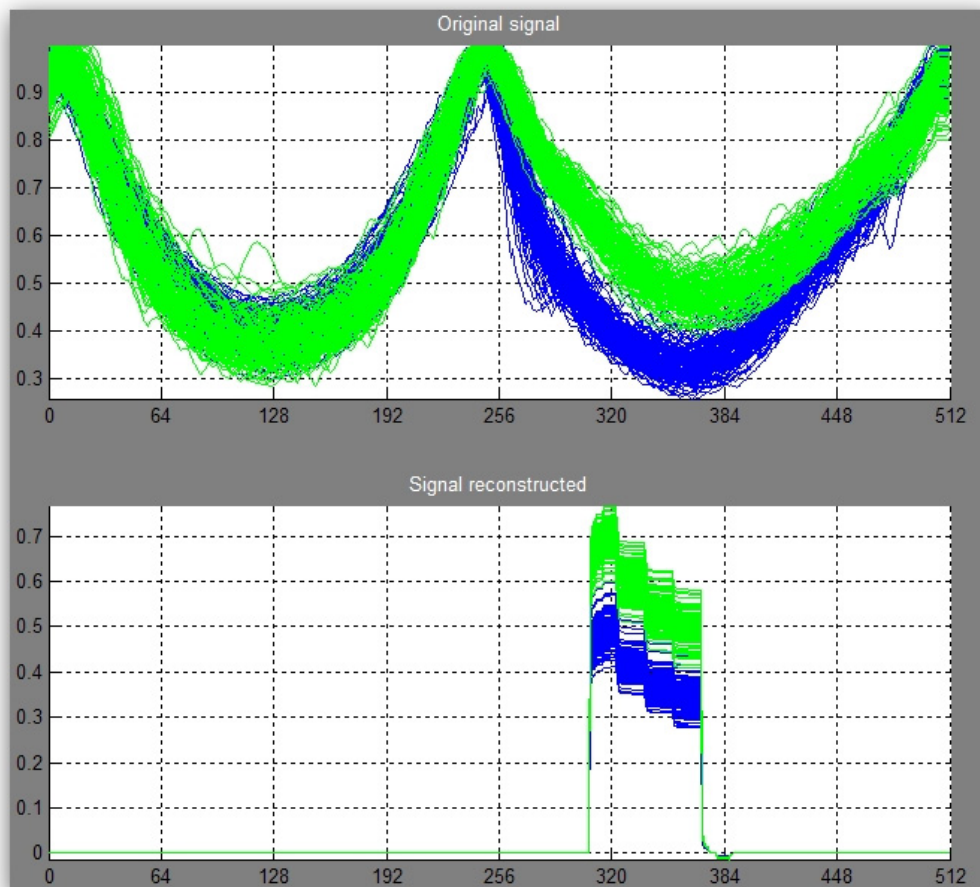
**Il·lustració 4.28** Taula amb el poder discriminant de cada filtre i posició del node òptim del diccionari *m-wave o7\_j55*

A continuació ens mostra el panell de reconstrucció. En aquest panell escollirem el filtre per mitjà del menú desplegable *Library* i seleccionarem l'espai que volem reconstruir a la caixa de text *Select* amb intervals, separats per guió, i números separats per coma.



**Il·lustració 4.29 Panell de reconstrucció WPT. Reconstrucció de l'interval 1-3,5 amb el filtre *m-wave o7\_i55***

Per fer la reconstrucció premerem *View*. Finalment dibuixarà el senyal original i el resultat de la reconstrucció. Si tenim la opció *Polar* activada dibuixarà també el senyal en contorns.



**Il·lustració 4.30 Dibuix del senyal original i el reconstruït per *Discriminant Analysis* del filtre *m\_wave o7\_i55***

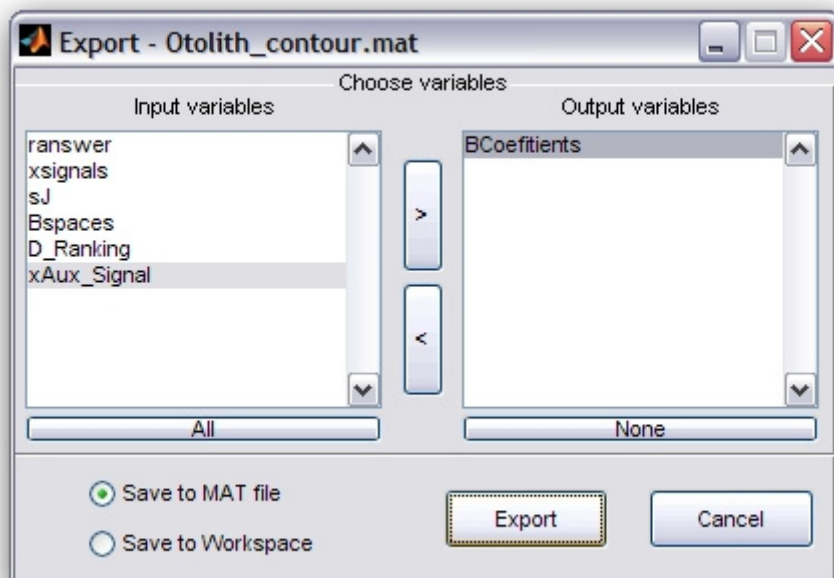
### 4.2.7. Exportació de variables

Per mitjà del menú *Export* obrim una nova finestra en la qual podem triar les variables que volem exportar i si les volem arxivar a un fitxer o desar-les a l'àrea de treball de MATLAB.

Si volem desar les variables que tenim en memòria a l'àrea de treball marcarem el botó radio "Save to Workspace"; si volem desar-les a un fitxer .MAT tindrem marcat el botó radio "Save to MAT file".

A continuació escollirem les variables que volem desar posant-les a la llista de sortida per mitjà dels botons >, que les afegirà individualment, i *All*, que les afegirà totes alhora. Un cop seleccionades podem modificar la selecció que hem fet traient-les d'una en una (<) o totes a la vegada (*None*). Mentre no tinguem cap variable seleccionada els botons <, *None* i *Export* estaran inhabilitats.

Finalment clicarem el botó *Export*, que obrirà una finestra d'explorador de fitxers si hem seleccionat "Save to MAT file" (il·lustració 4.31) o exportarà les variables a l'àrea de treball directament si el que hem seleccionat és "Save to Workspace".



**Il·lustració 4.31** Exportem les variables en memòria a un fitxer

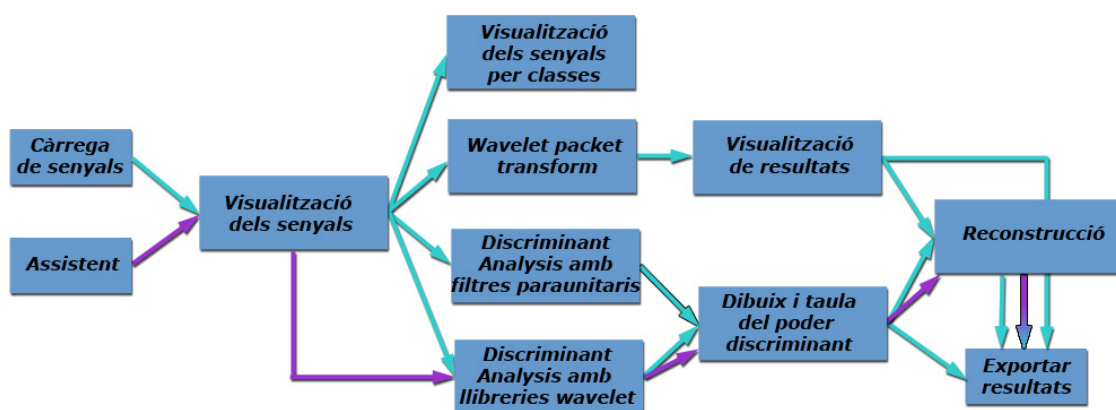
## 5. Funcions desenvolupades en la realització de la interfície gràfica

A continuació exposem la creació de les interfícies i alguns de les seves funcions d'execució. També exposem les funcions externes creades amb anterioritat a la realització d'aquest treball. Aquestes funcions han estat fetes en el marc d'una tesi doctoral en fase de desenvolupament. N'analitzarem els seus arguments i valors de sortida.

Com hem dit anteriorment, el programa no ha estat fet amb el GUIDE (*Graphical User Interface Development Enviroment*) sinó que ha estat programat per complet. El GUIDE és un joc d'eines dissenyades per crear GUIs (*Graphical User Interfaces*) facilitant el disseny i la presentació dels controls de la interfície. Permet seleccionar, arrossegat i canviar la mida dels elements i personalitzar les propietats. Quan aquests controls estan en posició s'editen les funcions d'execució (*Callback*) de cadascun d'ells, que s'executaran cada vegada que s'utilitzi el control. La figura queda desada en un fitxer *.FIG* i les funcions d'execució en un fitxer *.M*, generat automàticament.

En un principi, vam començar a fer aquesta aplicació en GUIDE, creant una figura amb els respectius fitxers *.FIG* (controls) i *.M* (funcions d'execució o *callbacks*). Però ens vam trobar amb incompatibilitats entre versions de MATLAB. Aleshores vam decidir crear els elements de la figura dins del mateix fitxer *.M*. D'aquesta manera només tenim un fitxer, d'extensió *.M*, en el qual hi ha les propietats dels controls, cadascuna de les seves funcions d'execució i altres funcions d'utilitat.

Tot seguit mostrem un esquema simplificat de les principals accions del programa:



Il·lustració 5.1 Diagrama de flux del programa

El handle és un número que identifica qualsevol objecte o element de la jerarquia gràfica d'una aplicació en MATLAB. S'obté com a valor de sortida en el moment de la creació de l'objecte i pot ser desat en una variable per fer referència posteriorment a aquest objecte. No és un valor conegut prèviament,

sinó que es genera i canvia cada vegada que executem el programa. És per això que per fer referència als objectes podem fer-ho per mitjà del nom de la seva variable o pel seu *Tag* (per exemple, la variable *editFirst*). En el nostre cas, el conjunt de variables creades a l'hora de definir cada objecte les desarem en una estructura anomenada *fig* (la crida anterior serà *fig.editFirst*).

Les variables de funcionament de l'aplicació les desarem juntament amb els handles, dins la seva estructura. En canvi, el senyal original, el vector de classes, el nivell de descomposició, els espais escollits i el rànquing de diccionaris creats per *Discriminant Analysis*, els coeficients, i el senyal reconstruït, el subespai seleccionat i els coeficients seleccionats creats durant la reconstrucció els desarem junts en una variable global en forma d'estructura de dades.

Cal dir que MATLAB distingeix entre majúscules i minúscules i, excepte en casos excepcionals, els noms que assignarem a les variables començaran en minúscula i els de les funcions en majúscula.

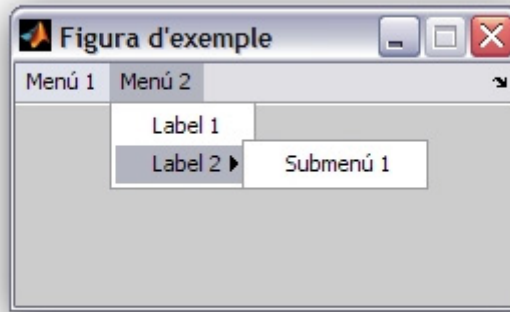
## 5.1. Creació de la interfície principal

En primer lloc crearem la figura. Un exemple senzill de creació de figures és el que segueix:

```
f=figure('MenuBar'           , 'none', ...% sense menú estàndard
        'ToolBar'           , 'none', ...% sense barra d'eines
        'Resize'            , 'off', ... % No maximitzar ni minimitzar
        'NumberTitle'       , 'off',... % sense número de títol
        'Position',[100 100 250 100],... % posició i mides absolutes
        'Name'              , 'Figura d''exemple'); % Nom de la finestra
m1 = uimenu(f,...           % Depèn de f
           'Label','Menú 1'); % Etiqueta
m11= uimenu(m1,...         % Depèn de m1
           'Label','Label &1');
m12= uimenu(m1,...         % Depèn de m1
           'Label','Label &2');
m2 = uimenu(f,...           % Depèn de f
           'Label','Menú 2'); % Etiqueta
m21= uimenu(m2,...         % Depèn de m2
           'Label','Label &1');
m22= uimenu(m2,...         % Depèn de m2
           'Label','Label &2');
m221= uimenu(m22,...      % Depèn de m22
           'Label','Submenú &1');
```

En aquest exemple creem una finestra sense menú estàndard, sense barra d'eines, sense número de títol i que no pugui canviar la mida. I seu nom serà "Figura d'exemple". A continuació creem dos menús, *menú 1* i *menú 2*, que contindran un parell de submenús cadascun (*Label 1* i *Label 2*). El *Label 2* del *menú 2* en tindrà un altre, a la vegada, amb el nom *submenú 1*. El resultat és el següent:





### Il·lustració 5.2 Exemple de creació d'una finestra

Posant la propietat *IntegerHandle* a *off* aconseguim que la figura no obtingui un número enter quan s'obri, de manera que es pugui treballar de forma segura des de la línia de comandes, obrint altres figures amb eixos, sense alterar el funcionament de l'aplicació. És important, doncs, posar la propietat *NumberTitle* a *off* per tal que no mostri aquest número al títol.

La posició que li donarem a la finestra la prendrem de la mida de la pantalla per tal que s'adapti a diferents resolucions. La seva amplada serà la de la pantalla menys 10 (2 marges de 5 píxels) i la seva alçada la de la pantalla menys 88 (2 marges de 44 píxels). La situarem a 5 píxels del marge dret i a 44 de baix, de manera que quedarà centrada. Farem dependre totes les mides dels elements a la mida relativa de la finestra, cap serà absoluta excepte algunes alçades de botons.

Li assignarem també una funció *CloseRequestFcn*, que serà la funció que cridarà l'aplicació quan es tanqui la finestra. Aquesta funció serà l'encarregada de netejar les variables globals i esborrar tota l'estructura de dades.

A continuació definirem tots els elements que formen la interfície principal: *uicontrols* (caselles de verificació o *check boxes*, caixes de text, llistes o *list-boxs*, menús desplegable o *pop-up menus*, botons radio o *radio buttons*, barres de desplaçament o *sliders* i textos estàtics), *uimenu*s (barra de menús), *uipanel*s (panells) i *axes* (eixos). Ho desarem tot en l'estructura de dades *fig*.

Tot seguit mostrem, a tall d'exemple, la creació d'un botó radio:

```
fig.radioStd=uicontrol(fig.uipanelDiscrimAnalysis, ...
    'Style','radiobutton', ...
    'String','Standard', ...
    'Enable','off', ...
    'Tag','radioStd', ...
    'TooltipString','Choose the library type', ...
    'Position',[fig.panelsAmplada/2+m ...
                fig.panelDiscrimAnalysisMenuAlçada*8/9 ...
                mtl ...
                aR ...
                ], ...
    'Callback',@RadioStd);
```

Propietats de l'exemple citat:

- *fig.uipanelDiscrimAnalysis*: pare de l'objecte.
- *'Style'*: quin tipus d'*uicontrol* és (*'radiobutton'*)
- *'String'*: text que ha de mostrar (*'Standard'*)
- *'Enable'*: indiquem si l'*uicontrol* és o no operatiu (*'off'*: no ho és, per tant la seva imatge apareix en gris).
- *'Tag'*: l'hem indexat per mitjà d'un *Tag* (*'radioStd'*)
- *'TooltipString'*: text que ha de mostrar quan passem el ratolí per sobre de l'element (*'Choose the library type'*)
- *'Position'*: posició de l'element dins del pare, en aquest cas amb unes variables calculades amb anterioritat
- *'Callback'*: rutina que ha d'executar quan hi hagi una acció sobre l'objecte (*RadioStd*)

L'ordre de creació dels controls indicarà l'ordre en què canviarà d'element cada vegada que polsem la barra tabuladora del teclat.

Un cop creats tots els objectes inicialitzarem algunes variables dins de la mateixa funció de creació de la interfície principal:

- *Complete\_dLibrary*: És una variable que es carrega des d'un fitxer *.MAT* amb el mateix nom, que conté tots els noms dels diccionaris wavelet. Són els següents:
  - *Daubechies*: db1 a db43
  - *Symlets*: sym2 a sym37
  - *Coiflets*: coif1 a coif5
  - *Biorthogonal*: bior1.1, bior1.3, bior1.5, bior2.2, bior2.4, bior2.6, bior2.8, bior3.1, bior3.5, bior3.7, bior3.9, bior4.4, bior5.5, bior6.8
  - *Reverse Biorthogonal*: rbio1.1, rbio1.3, rbio1.5, rbio2.2, rbio2.4, rbio2.6, rbio2.8, rbio3.1, rbio3.3, rbio3.5, rbio3.7, rbio3.9, rbio4.4, rbio5.5, rbio6.8
  - *Aproximació FIR Meyer*: dmey
  - *LCos, LSin*

Hem de suprimir *LCos* i *LSin*. Ho fem amb les instruccions

```
fig.Complete_dLibrary(...
    (ismember(fig.Complete_dLibrary, 'LCos')==1))=[];
fig.Complete_dLibrary(...
    (ismember(fig.Complete_dLibrary, 'LSin')==1))=[];
```

- *fig.esquerra*, *fig.dreta*, *fig.ultimaposicioEsquerra*, *fig.ultimaposicioDreta*:

Són variables que controlen les llistes del panell *Discriminant Analysis (LDB)*.

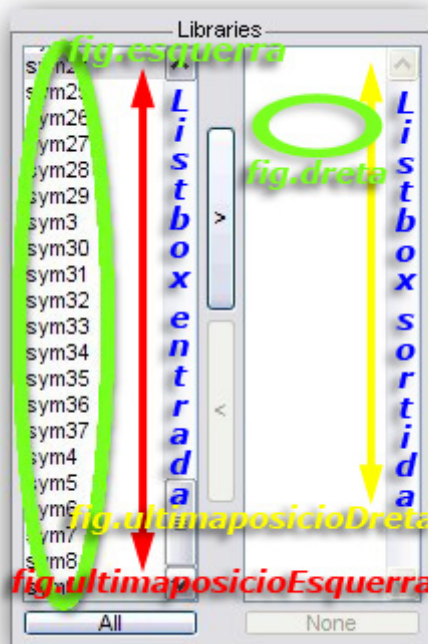
La variable *fig.esquerra* tindrà tots els valors de *Complete\_dLibrary*. Inicialment seran iguals ja que la cadena que posarem a la llista d'entrada serà *Complete\_dLibrary*.

La variable *fig.dreta* estarà buida. Serà la cadena que posarem a la llista de sortida, de la quina prendrem les dades.

La variable *fig.ultimaposicioEsquerra* serà l'última posició de la llista d'entrada (la longitud de *Complete\_dLibrary*).

I la variable *fig.ultimaposicioDreta* tindrà la posició inicial de la llista de sortida, que inicialment serà 0.

A mida que anem col·locant els noms dels diccionaris a la llista de sortida els anirem suprimint de *fig.esquerra* i afegint-los a *fig.dreta*, a la vegada que decrementem *fig.ultimaposicioEsquerra* i incrementem *fig.ultimaposicioDreta*. Quan els traiem de la llista de sortida per col·locar-los al d'entrada fem el procés invers: l'annexem a *fig.esquerra*, el suprimim de *fig.dreta*, incrementem *fig.ultimaposicioEsquerra* i decrementem *fig.ultimaposicioDreta*.



**II-lustració 5.3 Inicialització de variables que controlen les llistes del panell Discriminant Analysis (LDB)**

- *fig.StdOrCust*: Desarem a aquesta variable el valor per defecte 'Std'. Serà l'encarregada d'indicar a l'aplicació quin tipus d'anàlisi discriminant estem fent. Sempre començarem amb un anàlisi de diccionaris wavelet

(*Standard*) i, en cas que l'usuari canviï a filtres paraunitaris personalitzats (*Custom*) la variable prendrà el valor '*Cust*'.

Tot seguit amagarem la visibilitat de totes les funcions d'execució des de la línia de comandes per tal que no sigui modificable. Durant les rutines de les funcions d'execució els handles de la figura hi són visibles (podrem fer *gca*, *gcf*, *gco*, *findobj* i *newplot*). Amb aquesta instrucció protegim els handles i la figura, no permetent dibuixar des de la línia de comandes sobre els eixos de la figura i donant prioritat a les funcions d'accés als handles.

Finalment desarem l'estructura de handles de tota la figura (*gcf*) per mitjà de la instrucció

```
guidata(gcf, fig);
```

## 5.2. Creació de la interfície Assistent (Wizard)

L'Assistent facilita l'extracció i reconstrucció de senyals amb l'algorisme LDB.

La manera en què comunicarem les dues GUIs, la principal i l'Assistent, serà per mitjà d'una variable global creada abans de cridar l'Assistent, en la funció d'execució del menú *Use Wizard*. En cas que finalitzem amb èxit l'Assistent la posarem a 1 i la interfície principal posarà les mides que corresponen a l'extracció *Standard* del panell *Discriminant analysis (LDB)*, esborrant tot seguit la variable global. Enviarem també a l'Assistent l'estructura de handles *fig* de la interfície principal, que desarà dins de la seva estructura de handles.

Un cop haguem cridat la funció de creació de l'Assistent haurem de posar la instrucció *uiwait* a la interfície principal per tal que no continuï amb el programa. Per tornar a donar-li el control haurem de posar la instrucció *uiresume* en cadascuna de les funcions que tanquen la finestra l'Assistent:

Principal	Assistent
<code>function MenuOpenWizard</code>	<code>function LDB_Wizard (varargin)</code>
<code>...</code>	<code>...</code>
<code>LDB_Wizard(fig);</code>	<code>end</code>
<code>uiwait(gcf)</code>	<code>function ButtonExitWiz (varargin)</code>
<code>if loadOk==1</code>	<code>...</code>
<code>...</code>	<code>loadOk=0; % Càrrega no</code>
<code>end</code>	<code>    % satisfactòria</code>
<code>clear loadOk</code>	<code>uiresume(wiz.wizardLDB);</code>
<code>guidata(gcbo, fig);</code>	<code>delete(wiz.wizardLDB);</code>
<code>end</code>	<code>end</code>

En la funció de creació de components de l'Assistent crearem una finestra centrada. Prendrem la mida de la pantalla per donar la posició i les mides a la nova finestra per tal que s'adapti a diferents resolucions. La seva amplada serà el 40% de l'amplada de la pantalla i la seva alçada la meitat de l'alçada de la pantalla. D'aquesta manera farem dependre les mides dels elements a la mida relativa de la finestra en lloc de ser absolutes, excepte algunes alçades de botons. No serà visible fins que tots els seus components estiguin creats.

La definirem com a finestra modal, una finestra que no permet canviar el focus d'entrada a altres finestres de MATLAB i resta apilada sobre de les figures normals i de la finestra d'ordres.

A continuació definirem tots els elements que formen la finestra de l'Assistent: *uicontrols* (caixa de text, llistes o *list-boxs*, menús desplegable o *pop-up menu*, botons, botó radio o *radio button* i textos estàtics), *uipanel*s (panells) i *axes* (l'eix). Ho desarem tot en l'estructura de dades *wiz*.

Hi inicialitzarem un comptador per saber en quin panell ens trobem, ja que utilitzarem un sol botó per avançar i un per retrocedir en els seus panells durant l'extracció i la reconstrucció. Ens basarem en aquest comptador per inhabilitar el botó *Next* en el primer panell o fer-lo invisible en l'últim. Només en aquest últim panell haurem de fer visible el botó *Finish*, i en el primer panell haurem de fer invisible el botó *Back*.

Carregarem els noms dels diccionaris a una variable de l'estructura de l'Assistent i inicialitzarem un camp a l'estructura de handles per desar-hi el senyal original, el vector de classes, el nivell de descomposició, els espais escollits i el rànquing de diccionaris creats per *Discriminant Analysis*, els coeficients, i el senyal reconstruït. En principi desarem aquestes variables en un camp de l'estructura de components i, si finalitzem l'Assistent amb èxit, les desarem a la variable global *fileGlobal* en forma d'estructura de dades (en cas que aquesta variable global ja hi tingui dades la substituirà).

Finalment desarem l'estructura de handles de tota la figura de l'Assistent (*gcf*) per mitjà de la instrucció

```
guidata(gcf,wiz);
```

### 5.3. Creació de la interfície Exportar/Importar

Crearem aquesta finestra a l'hora de carregar un fitxer o exportar variables des de la interfície principal. És una finestra que conté els mateixos components per ambdós casos, però amb diferent comportament.

Comunicarem la finestra principal i la d'Importar/Exportar per mitjà d'una variable global creada abans de cridar la segona finestra. En aquest cas només ens caldrà la variable global en la funció d'execució del menú *Load*, ja que si exportem no hem de modificar res de la finestra principal. En cas que carreguem amb èxit algun fitxer la posarem a 1 i la interfície principal modificarà les característiques dels seus components per adaptar-s'hi, esborrant tot seguit la variable global.

Per mitjà del pas d'arguments li donarem el títol de la finestra, el text que han de mostrar els seus botons radio, els botons que estaran habilitats i els que no i un factor de multiplicació per la mida del botó *Cancel*, comú en ambdues situacions, per tal que s'adapti a la mida de la finestra:

```
ExportImport (nomFigura, strRadioExportMAT, ...
              strRadioExportWorkspace, ...
              positionCancel, lbEntrada, ...
              lbSortida, bAfegTot, bTreuCap);
```

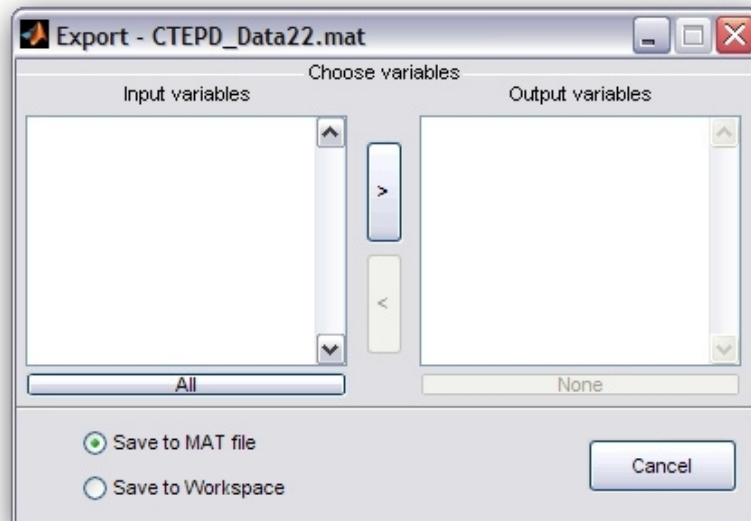
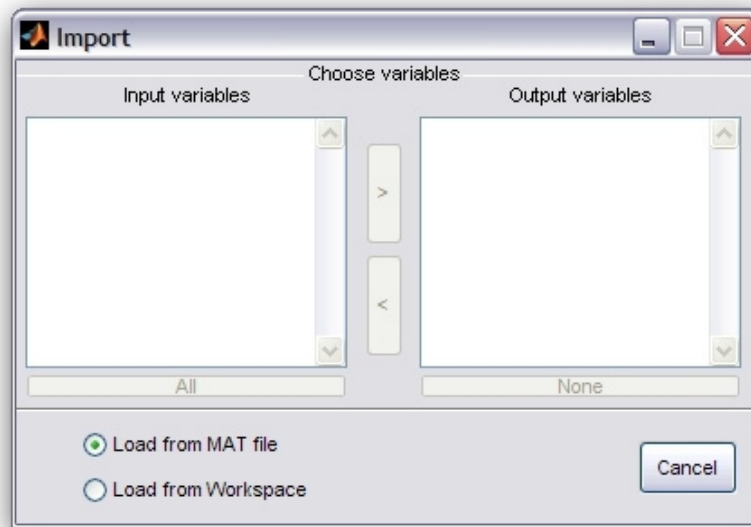
En el cas que vulguem exportar algunes variables, el títol que li posarem serà *Export* seguit del nom del fitxer que exportarem, separat amb guió.

La finestra serà modal i per posicionar-la ho farem a partir de la mida i la posició de la finestra principal i de la pantalla. Estarà a 25 píxels del seu marge esquerre i a 2/3 parts del de baix. La seva amplada serà del 40% de l'amplada de la pantalla i la seva alçada del 33.3% de l'alçada de la pantalla.

Crearem els objectes que són comuns a ambdós casos, posant-los les variables dels arguments en els llocs corresponents.

```
function ExportImport (...)
...
exp.radioExportMAT=icontrol (exp.uipanelRadiosContenedor, ...
    'Style', 'radiobutton', ...
    'String', strRadioExportMAT, ...
    'Enable', 'on', ...
    'Tag', 'radioExportMAT', ...
    'Value', 1, ...
    'Position', [p(3)/2-mrE-mr ...
                 p(4)/2+m/2 ...
                 mrE ...
                 15 ...
                ], ...
    'Callback', @RadioExportMAT);
...
ampB=(p(3)/4-2*m)*positionCancel;
exp.botoCancel=icontrol (exp.uipanelRadiosContenedor, ...
    'Style', 'pushbutton', ...
    'String', 'Cancel', ...
    'Enable', 'on', ...
    'Tag', 'botoCancel', ...
    'Position', [p(3)-m-ampB ...
                 p(4)/2-mbE/2 ...
                 ampB ...
                 mbE ...
                ], ...
    'Callback', @BotoCancelExport);
...
end
```

Obtindrem una de les següents figures:



**Il·lustració 5.4 Objectes comuns pel menú *Load* i pel menú *Export*  
Només canvia el nom dels objectes i la mida del botó *Cancel***

Un cop creats aquests objectes comuns definirem els botons únics per cada cas i la seva posició.

Sabrem si exportem o importem per mitjà de la variable *menu*, creada anteriorment, que conté el handle de l'objecte que ha fet la crida. Aleshores farem una comparació amb el handle del menú *Load* (variable *valorLoad*) i el del menú *Export* (variable *valorExport*) i podrem fer la tasca corresponent a cada cas.

És a dir:

```

function ExportImport (...)
...
menu=gcbob;    % Element que ha fet la crida
valorLoad=fig.menuLoad;    % Handle del menú Load
valorExport=fig.menuExport; % Handle del menú Export
...
switch menu
    case valorLoad
        ...
    case valorExport
        ...
end
end
end

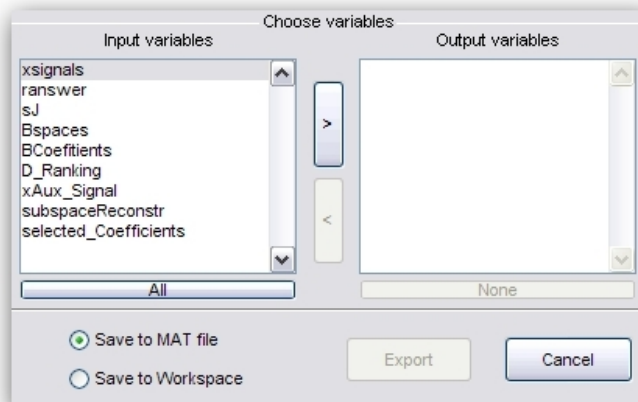
```

En cas que vulguem carregar un fitxer crearem un botó *Import* i un botó *Load*. El botó *Import* ens servirà per veure les variables que tenim dins del fitxer o en l'àrea de treball de MATLAB, segons el que haguem escollit, i el botó *Load* per carregar les variables que haguem seleccionat.

En cas que el que vulguem fer sigui exportar crearem un botó *Export* i, per mitjà de la instrucció *fieldnames*, obtindrem els noms de les variables de resultats que tenim en memòria. A continuació posarem, com a string de la llista esquerra aquests noms i inicialitzarem les variables que controlen aquestes llistes d'entrada i sortida de variables. El botó *Export* exportarà les variables seleccionades a l'àrea de treball de MATLAB o a un fitxer, segons el que haguem escollit.



(a)



(b)

**II-lustració 5.5 Objectes únics del menú *Load* (a) i del menú *Export* (b)**



## 5.4. Funcions d'execució dels elements

En total hem fet 73 funcions en la interfície principal (71 d'execució i les dues de creació de la figura inicial i la d'exportar i importar). D'aquestes, 9 pertanyen a la finestra d'importar i exportar. En la de l'Assistent n'hem fet 23, i la de la creació de la figura. En els punts que segueixen exposem només 6 funcions d'execució per no fer-lo excessivament extens. Les 3 funcions de creació les hem explicades anteriorment i, les funcions d'execució que explicarem a continuació són les més importants, ja sigui perquè s'utilitzen sovint, perquè, segons la complexitat McCabe, són les més complicades o perquè són necessàries pel processament del senyal.

Els noms de les funcions de la interfície principal són els següents:

*LDB\_GUI* (creació de la interfície)

*LDB\_CloseRequestFcn, MenuExit, PopupGroupNumber, EditFirst, EditLast, RadioStd, RadioCust, ButtonView, ButtonRemove, EntraK, EntraN, ButtonCreateComb, BotoAfegeix, BotoTreu, BotoTots, BotoCap, ButtonProcess, MenuPolar, MenuToolbar, MenuSaveSession, MenuLoadSession, MenuCloseSession, MenuShowHidePlot, MenuWaveletPT, MenuLDB, CheckGrid, CheckPan, CheckZoom, CheckCursor, MenuOpenWizard, RadioSelect, ButtonViewRec, ButtonRemoveRec, MenuSave, SliderNivellDescomp, EntraNivellDescomp, SliderNivellDescompWPT, EntraNivellDescompWPT, ButtonProcessWPT, MenuTools, MenuView, MenuFile, PopupJ, ButtonAddToReconstr, ButtonAddNodeToReconstr, ButtonRemoveToReconstr, ButtonRemoveAllToReconstr, ButtonReconsCustDict, MenuLoad, MenuExport.*

Les funcions d'utilitat per la interfície principal són:

*Modif\_interf\_inicial, EnableDisable, ComparaCadena, FetElProcess, FuncPlot, FuncPolar, CheckMostraJKL, InicialitzaTaula, ExtentTaula, SlideTaula, DibuixaBspaces, AxesResetIInvisible*

La interfície que importa i exporta en comparteix algunes amb la principal, però les pròpies són les funcions següents:

*ExportImport* (creació de la interfície)

*BotoImport, BotoLoad, RadioExportMAT, RadioExportWorkspace, BotoExport, SaveToWorkspace, SaveToMATFile, Export\_CloseRequestFcn, BotoCancelExport*

Finalment, les funcions de l'Assistent són:

*LDB\_Wizard* (creació de la interfície)

*LDB\_Wizard\_CloseRequestFcn, ButtonFinishWiz, ButtonNextWiz, ButtonPrevWiz, ButtonLoadWiz, ButtonExitWiz, BotoTotsWiz, BotoCapWiz, BotoAfegeixWiz, BotoTreuWiz, RadioContourWiz, RadioProfileWiz, RadioSelectWiz,*

I les funcions d'utilitat per l'Assistent:

*CarregaLlibreries, Modif\_interf\_inicial\_Wiz, EnableDisableWiz, FinishedWiz, ComparaCadenaWiz, FuncPlotWiz, FuncPolarWiz, RadioStdFig, RadioCustFig*

Les funcions d'execució, o *Callbacks*, defineixen les rutines que executarem cada vegada que hi hagi una acció sobre un element de la figura. Hi ha elements que no en necessitaran, com els nostres textos estàtics, d'altres que seran molt complexes, com el menú *Load session* i d'altres que no en seran tant, com la del botó radio *RadioSelect*.

Per gestionar les dades en cadascuna de les funcions d'execució començarem obtenint una còpia de les dades desades amb la instrucció

```
fig=guidata(gcbo);
```

Aquesta instrucció retorna a l'estructura *fig* les dades emmagatzemades prèviament.

A continuació realitzarem els canvis desitjats i finalment desarem la versió modificada de les dades per mitjà de la instrucció

```
guidata(gcbo, fig);
```

Si dins d'una funció en cridem una altra i hi modifiquem variables també haurem d'utilitzar la instrucció *guidata* per desar els canvis fets abans de cridar la segona funció i recarregar les variables quan en retornem. Tot seguit en mostrem un exemple.

```
function ButtonProcessWPT (varargin)
fig=guidata(gcbo); % llegeim guidata.
global fileGlobal
EnableDisable(gcbo, 'off', 0) % No cal que desem
% l'estructura perquè
% en aquesta funció no
% farem canvis

set(... % Fem canvis en les
fig.nivellDescomposatsJ=str2double(... % variables

guidata(gcbo, fig) % Desem els canvis fets
EnableDisable(gcbo, 'on', 0) % Cridem la funció
fig=guidata(gcbo); % Actualitzem

set(... % Fem canvis

guidata(gcbo, fig) % Desem els canvis fets
FetElProcess(fig) % Cridem la funció
fig=guidata(gcbo); % Actualitzem
```

```

ButtonRemove ();                                % Els canvis estan
                                                % acabats de desar i la
                                                % funció que cridem
                                                % desarà els seus

set (...                                         % Fem canvis

guidata(gcbo,fig)                               % Desem els canvis fets
end

```

En la instrucció `fig=guidata(gcbo)` dessem a una estructura anomenada *fig* els handles relacionats amb l'objecte (*gcbo*) que ha fet la crida. Així, podem utilitzar una mateixa funció per diverses interfícies, de manera que la funció agafarà l'estructura de dades de l'objecte que faci la crida. En el programa hem definit, per exemple, un *Callback* a la funció *BotoAfegeix* pel botó *fig.botoAfegeix*, de la interfície principal, i el mateix pel botó *exp.botoAfegeix*, de la interfície d'importar i exportar. Quan fem la crida des del botó *fig.botoAfegeix* agafarà les dades de l'estructura de handles de la interfície principal, i quan fem la crida des del botó *exp.botoAfegeix* agafarà les dades de la interfície d'importar/exportar, l'estructura *exp*.

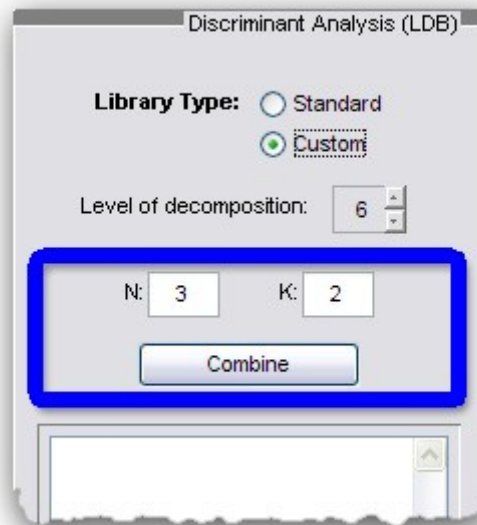
A continuació analitzarem algunes funcions creades per realitzar l'aplicació gràfica que ens permeten adaptar dades entrants a les demanades per les funcions externes, desar-les o analitzar-les.

#### 5.4.1. Creació de $N^K$ combinacions. Funció *ButtonCreateComb*

Utilitzem aquesta funció quan creem les combinacions per la matriu d'angles dels filtres paraunitaris personalitzats.

Aquesta funció genera totes les combinacions possibles d'un conjunt de 1 a N. En total hi haurà  $N^K$  combinacions, on K és número d'elements.

Pren les dades de les caixes de text *N* i *K* (la discretització de l'interval  $[0, \pi)$  i la longitud de la resposta impulsional respectivament) i actua quan premem *Combine*. El seu argument serà el handle de l'objecte que ha fet la crida, en aquest cas el botó *Combine*.



**Il·lustració 5.6 Panell de creació de la matriu d'angles**

Els passos que segueix aquesta funció són els següents:

- Desa les variables  $N$  i  $K$  a l'estructura de handles *fig* (*fig.N* i *fig.K*) i comprova que les dades entrades no siguin caràcters. La comprovació d'enters i de zeros l'hem feta anteriorment.

- Crea un vector fila  $N_1$  amb números enters de 1 a *fig.N*:

$$N_1 = [1 \ 2 \ 3]$$

- Genera un conjunt de cel·les amb el vector  $N_1$  d'una fila i  $K$  columnes.

$$ord = [1 \ 2 \ 3] [1 \ 2 \ 3]_{1 \times K, cell}$$

- Executa la instrucció *ndgrid*. És una instrucció per generar series de  $K$  dimensions i interpolacions.

$$A = \begin{bmatrix} 1 & 1 & 1 \\ 2 & 2 & 2 \\ 3 & 3 & 3 \end{bmatrix} \begin{bmatrix} 1 & 2 & 3 \\ 1 & 2 & 3 \\ 1 & 2 & 3 \end{bmatrix}_{1 \times K, cell}$$

- Refà la matriu amb cadascuna de les cel·les de  $A$  en la dimensió  $k+1$ .

$$A = \begin{bmatrix} 1 & 1 & 1 \\ 2 & 2 & 2 \\ 3 & 3 & 3 \end{bmatrix}_{3 \times 3 \times 1 (N \times N \times 1)} \begin{bmatrix} 1 & 2 & 3 \\ 1 & 2 & 3 \\ 1 & 2 & 3 \end{bmatrix}_{3 \times 3 \times 2 (N \times N \times K)}$$

- Per mitjà de la instrucció *reshape* reordena la matriu i li canvia la mida per ajustar-la a les dimensions desitjades. Concatena les columnes de la

primera pàgina de  $A$  i les posa a la primera columna, les de la segona pàgina a la segona columna, i així successivament. Per tal que els elements que canvien amb més rapidesa estiguin a l'última columna invertirem l'ordre d'aquestes.

$$A = \begin{bmatrix} 1 & 1 \\ 2 & 1 \\ 3 & 1 \\ 1 & 2 \\ 2 & 3 \\ 3 & 2 \\ 1 & 3 \\ 2 & 3 \\ 3 & 3 \end{bmatrix}_{9 \times 2(N^K \times K)} \quad ; \quad A_{\text{ordre\_invertit}} = \begin{bmatrix} 1 & 1 \\ 1 & 2 \\ 1 & 3 \\ 2 & 1 \\ 2 & 2 \\ 2 & 3 \\ 3 & 1 \\ 3 & 2 \\ 3 & 3 \end{bmatrix}_{9 \times 2(N^K \times K)}$$

- Crea una variable  $B$  que conté els noms que utilitzarem. La plantilla que seguim per anomenar-los és  
 'm-wave o',ordre,'\_i',NK<sub>i</sub> ,  
 on  $\text{ordre} = 2 * \text{size}(A,2) + 1$  i  $NK_i$  anirà de 1 a  $N^K$
- Centra ara la matriu  $A$  en l'interval  $[0, \pi)$ . Com que MATLAB ens permet fer operacions element per element en les matrius, podem centrar-la en l'interval amb una sola línia de codi.

$$A = \frac{A - N \cdot (\pi - \pi/N)}{N - 1} + (\pi - \pi/N)$$

$$A = \begin{bmatrix} 0 & 0 \\ 0 & 1,04719755 \ 119660 \\ 0 & 2,09439510 \ 239320 \\ 1,04719755 \ 119660 & 0 \\ 1,04719755 \ 119660 & 1,04719755 \ 119660 \\ 1,04719755 \ 119660 & 2,09439510 \ 239320 \\ 2,09439510 \ 239320 & 0 \\ 2,09439510 \ 239320 & 1,04719755 \ 119660 \\ 2,09439510 \ 239320 & 2,09439510 \ 239320 \end{bmatrix}_{9 \times 2(N^K \times K)}$$

- Desa els resultats de  $A$  i els noms generats a  $B$  en variables als handles  $fig$ .

### 5.4.2. Creació d'un vector a partir d'una cadena. Funció *ComparaCadena*

La funció *ComparaCadena* s'executa durant la reconstrucció, en la interfície principal i en l'Assistent. És l'encarregada de retornar en forma de vector la cadena introduïda a la caixa de text del panell de reconstrucció. Aquesta cadena pot estar formada per números i intervals separats per coma:

```
1,3-8,17-7,2,83,3963
```

#### II-lustració 5.7 Cadena introduïda a la funció *ComparaCadena*

La funció es basa en la instrucció *strtok*, que cerca un delimitador dins d'una cadena de caràcters i ens retorna la cadena fins al delimitador en una variable i la resta fins al final en una altra. És a dir, fent

```
[a remain]=strtok('17-7', '-')
```

ens retornaria les variables  $a = 17$  i  $remain = -7$ .

La funció *eval* executa l'expressió que té entre parèntesis com a cadena de caràcters i *sprintf* li dona el format. D'aquesta manera podem crear variables i treballar-hi.

Aquesta funció fa comparacions per saber si la dada que està tractant és escalar i entera. En cas que no ho sigui ens adverteix i surt de la funció sense tractar el vector.

Per mostrar els passos que segueix la funció *ComparaCadena* introduïrem la cadena *1,3-8,17-7,80,2,83,3963*. És una cadena en la qual es repeteixen números, un interval està invertit, i supera la mida del senyal (serà un senyal de 512 mostres).

- Inicialitzem les variables *vect* (variable on anirem desant el vector) i *pM* (ens indicarà si el vector té elements majors que la matriu). Inicialitzem també la variable *k* a 1, que donarà el nom a les variables que crearem, *P\_%d*, per desar els intervals i números temporals. En cada bucle incrementarem *k* per obtenir les variables *P\_2*, *P\_3*...
- Fem bucles cercant a la cadena d'entrada fins a la primera coma i ho desem en una variable temporal, *P\_1* deixant la resta a la variable *remain*. En el nostre cas obtenim:

```
P_1 = 1
```

```
remain = ,3-8,17-7,2,83,3963
```

- Cerquem dins la variable *P\_1* acabada de crear la posició d'algun guió, per saber si es tracta d'un interval. En aquesta cadena no n'hi tenim cap.
- Si no és un interval desem la variable *P\_1* a una nova variable *a* i la converteix a format numèric. Si és escalar, enter i més gran que 1 el

concatena a la variable *vect*. Si no es compleix la condició ens advertirà i sortirà de la funció. Tenim el vector:

```
vect = 1
```

- En el següent bucle detectem que tenim un guió: desarem a la variable *a* els caràcters fins al guió i a la variable *b* la resta, transformant-los a números. Si són escalars, enters i més grans que 1, i *b* és major que *a*, adjuntem l'interval al vector creat.

```
vect=eval(sprintf('[vect a:b];')); 
```

Obtenim:

```
vect = 1 3 4 5 6 7 8
```

- Detectem que el proper interval és invertit ja que  $b > a$ , així que farem:

```
vect=eval(sprintf('[vect b:a];')); 
```

I obtindrem:

```
vect = 1 3 4 5 6 7 8 7 8
      9 10 11 12 13 14 15 16 17
```

- Mentre la variable *remain* encara tingui alguna cadena farà el mateix procés fins a obtenir el vector:

```
vect = 1 3 4 5 6 7 8 7 8 9 10
      11 12 13 14 15 16 17 2 83 3963
```

- El posem en ordre de menor a major i si hi ha algun camp que sobrepassa el número de senyals els desarem a *pM*, en el nostre cas  $pM = 3963$ , i la resta a *vect*. Si *pM* conté algun valor avisarem que hem adaptat el vector.
- Finalment traurem els valors repetits del vector *vect*. El valor de retorn serà el vector *vect*, que en el nostre exemple és

```
vect = 1 2 3 4 5 6 7 8 9 10
      11 12 13 14 15 16 17 83
```

### 5.4.3. Habilitació o inhabilitació dels components. Funció *EnableDisable*

Aquesta funció la utilitzarem per netejar tota la interfície principal i inicialitzar algunes variables cada vegada que obrim un fitxer, tanquem la sessió o n'obrim una, i cada vegada que vulguem que quedin inhabilitats o habilitats tots els camps de la interfície.

Els seus arguments són *gcbo*, *able*, *inici* i *varargin*. L'argument *gcbo* indica l'objecte que ha fet la crida. L'argument *able* és una cadena de caràcters que pot prendre el valor *'off'* o *'on'* i ens indicarà si habilitem o inhabitem els components de la interfície. L'argument *inici* prendrà el valor *0* o *1* i ens indicarà si estem obrint un fitxer (*1*) o no (*0*).

Una de les instruccions bàsiques és la instrucció *fieldnames*, que retorna tots els noms d'una estructura. Una altra instrucció important és *findstr*, que cerca una cadena de caràcters dins d'una altra, així, compararem cada camp de l'estructura *fig* amb una cadena de caràcters definida.

### Arguments (*gcbo,'off',0*) o bé (*gcbo,'on',0*)

En cas que no estiguem obrint un fitxer només passarem amb un punter per tota l'estructura *fig* per comparar els seus noms amb cadenes de caràcters que haurem definit, per habilitar-los o no segons sigui l'argument *able*. És a dir, un cop fet

```
it=fieldnames(fig);
```

i haguem obtingut tots els camps de l'estructura, passarem amb un punter per tots els seus camps, fent:

```
...
if (~isempty(findstr(char(it(i,:)), 'edit')))
    set(fig.(char(it(i,:))), 'Enable', able);
end
...
```

D'aquesta manera inhabilitarem tots els camps que continguin el text *edit* al seu nom, com ara *editFirst*, *editLast* i *editSelect*.

### Arguments (*gcbo,'off',1*)

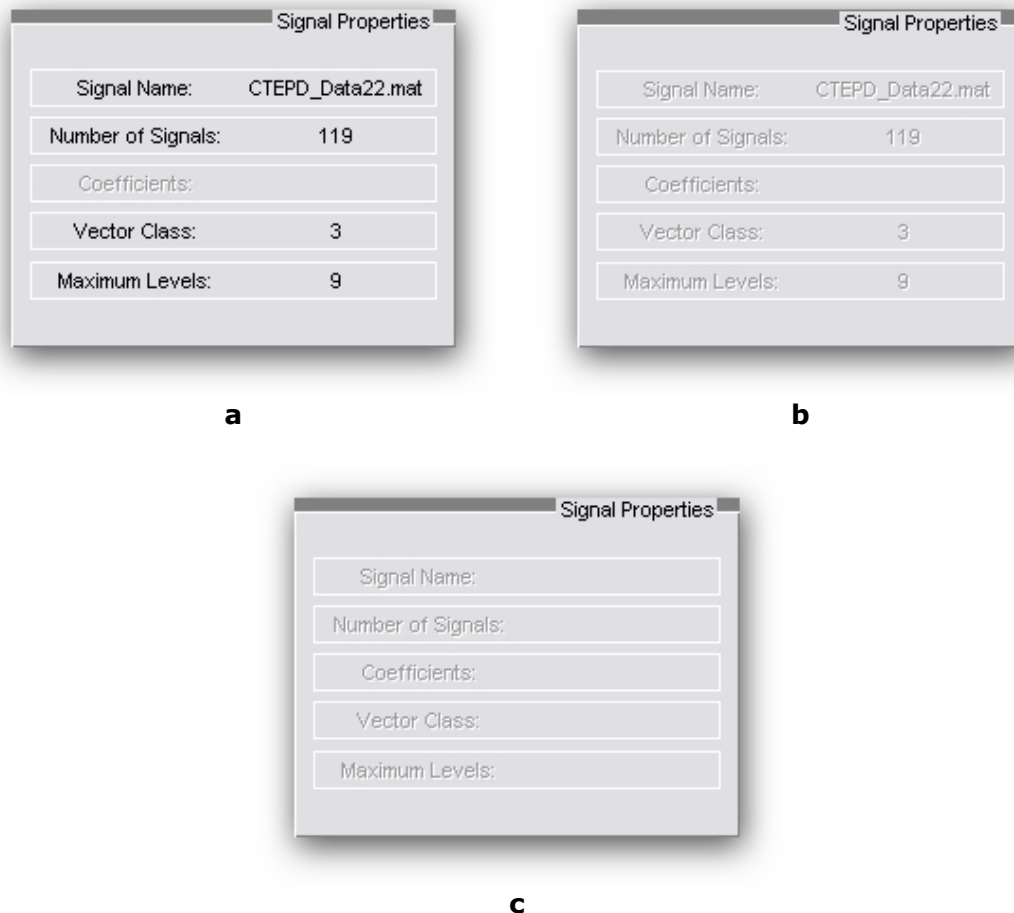
En cas que estiguem obrint un fitxer o obrint o tancant sessió (*inici=1*) també en posarem alguns d'invisibles, o en valor *0*, o inhabilitat per defecte, o en buidarem la seva cadena de caràcters. Per exemple,

```
...
if (~isempty(findstr(char(it(i,:)), 'var')))
    if inici==1
        set(fig.(char(it(i,:))), ...
            'Enable', 'off', 'String', '');
    end
end
...
```

En aquest cas cercarem tots els camps que continguin la cadena de caràcters *var* i, com que *inici=1*, els inhabilitarem i en buidarem els seus *strings*.

Tot seguit netejarà la interfície de panells, dels eixos i de la taula de rànquing. Per últim inicialitzarà les variables que controlen les llistes del panell *Discriminant analysis (LDB)*.





**Il·lustració 5.8 Resultat de la funció *EnableDisable*.**  
**Arguments: a) *gcbo,'on',0*. b) *gcbo,'off',0*. c) *gcbo,'off',1***

En l'Assistent també l'utilitzem, però amb alguns canvis: cerca cadenes en la interfície principal i les inhabilita o les habilita, i cadenes de l'Assistent. Les cadenes de text de l'Assistent que cercarà seran *text*, *var*, i *button*. Abans de fer-ho posarà la finestra a invisible, ja que és modal i si no ho fes emetria un so cada vegada que modifiqués la interfície principal. Si *inici* és 1, és a dir, si carreguem un fitxer, neteja les variables de senyal de l'Assistent (*wiz.file.xsignals*, *wiz.file.ranswer*, *wiz.file.sJ*), torna a centrar la finestra, habilita o inhabilita alguns dels seus components, i inicialitza les variables que controlen les llistes dels diccionaris. Finalment la torna a fer visible.

#### 5.4.4. Modificació de la interfície inicial. Funció *Modif\_interf\_inicial*

La funció *Modif\_interf\_inicial* és l'encarregada de modificar la interfície principal o la de l'Assistent per tal que mostri les dades principals del fitxer importat o de la sessió carregada o nova al panell *Signal Properties* i dibuixi el senyal original. No funcionarà igual en la interfície principal i en l'Assistent.

En primer lloc habilitarem el text que ens mostra el nom del senyal i l'hi posa.

En ambdues interfícies comprova el nom del senyal carregat i, si és *xdata* el canvia a *xsignals*. Ho fa també amb el vector de classes, si hem carregat un vector anomenat *rlabel*, canviant-lo a *answer*. Si no carreguem cap variable que contingui el valor de nivell de descomposició la crea, per mitjà de la fórmula:

$$sJ = \log_2(\text{num\_mostres}) , \quad \text{Equació 5.1}$$

on *num\_mostres* són el número de mostres del senyal.

En la interfície principal podem carregar coeficients, per tant, si carreguem coeficients i vector de classes calcularíem el nombre de coeficients per mitjà de la fórmula

$$\text{num\_mostres} \times (\text{nivell\_descomposició} + \Omega_{(0,0)}) , \quad \text{Equació 5.2}$$

on *num\_mostres* són el número de mostres del senyal. Però per tal que no depengui de si carreguem un senyal, el calcula en funció de *BCoefficients*, la variable de coeficients. Un cop carregats obrirà el panell de reconstrucció.

Aleshores, tant en la interfície principal com en l'Assistent, comprova les variables carregades i habilita les caixes de text que corresponen a cadascuna.

#### Si hem carregat un senyal original i el seu vector de classes

- Interfície principal. Si **no és una nova sessió**.

En aquest cas posa a punt el panell *Signal representation*, mostrant-nos en un menú desplegable el nombre de classes que hi ha i assenyalant-nos-en el primer i l'últim element que podem representar, en principi, per la primera classe.

Posarà a punt també el panell *Discriminant Analysis (LDB)* i el *Wavelet Packet transform*, habilitant alguns dels seus elements.

Inicialitza variables com *ContourOrProfile='Profile'*, ja que inicialment dibuixarem perfils d'intensitat, i els valors màxim i inicial i el pas de les barres de desplaçament (*sliders*) del nivell de descomposició. Aquestes barres de desplaçament han de tenir increments de números enters, per tant, quedaran de la següent manera:

Valor màxim i inicial:  $'Max' = 'Value' = fileGlobal.sJ$

Pas:  $'SliderStep' = 1/(fileGlobal.sJ-1)$

Finalment, si hem carregat un senyal i el seu vector de classes, dibuixem el senyal en forma radial.

- Assistent.

Com que la finestra de l'Assistent és modal, és a dir, resta apilada sobre de les figures normals i de la finestra d'ordres de MATLAB, cada vegada que dibuixem en una altra finestra (només es pot fer per mitjà del programa) emetrà un so. Per tal d'evitar-ho la transformarem en finestra normal.

Si hem marcat l'opció de carregar un fitxer de perfils dibuixarà el senyal en vista radial. Si hem marcat l'opció de carregar un senyal de contorns dibuixarà el contorn i el perfil en dos eixos diferents.

Aleshores habilitarà tots els botons i posarà la finestra de l'Assistent davant amb la instrucció *uistack*. Finalment la tornarà a fer modal.

#### 5.4.5. Arxivament de la sessió actual. Funció *MenuSaveSession*

Aquesta funció desa les característiques actuals de l'aplicació, els panells que hi ha oberts, els menús habilitats, els objectes visibles, el fitxer carregat... en un fitxer *.MAT* al directori MATLAB de l'usuari per defecte. Només estarà disponible en cas que haguem carregat algun senyal.

En primer lloc comprova si tenim alguna llegenda a la interfície, ja que podria causar problemes i la desactiva.

Gravarà variables en forma de vector columna tots els *Tags* (els noms dels objectes), la visibilitat de tots els objectes, el *Tag* de totes les caixes de text editables i els seus valors, el *Tag* de tots els textos fixes i els seus valors, la barra d'eines, si hi és, els menús que estan marcats, els *Tags* i les etiquetes dels menús (n'hi ha que varien, com *Show/Hide Toolbar*), els *Tags* i els valors dels botons radio i la visibilitat de tots els eixos. Ho farem per mitjà de les instruccions *findall* i *findobj*.

Desem també els valors de *fig.esquerra* i *fig.dreta*, que són els quins controlen les llistes del panell d'extracció.

Ens quedarà, per exemple,

```
axes = 'on'      radioTag = 'radioSelect'   radioVal = 0
      'off'      'radioCust'              0
      'off'      'radioStd'               1
      'off'
      'off'
      'off'
```

Gravarà en variables el senyal original, el vector de classes i el nivell de descomposició màxim.

Si hem creat una variable *Bspaces*, que conté el poder de discriminació de cada node i la seva posició per cadascun dels diccionaris processats, també la desarem, tot i que si la seva mida supera els dotze diccionaris desarem els dotze primers. Ho farem amb totes les variables creades durant la sessió, ja que la mida del fitxer de sessió podria ser excessiva.

En aquest punt preparem una cel·la anomenada *stringPelWarndlg* amb el text "We will save only first 12" per tal d'anar afegint-hi opcions a mida que anem suprimint pàgines o files de les matrius quan sigui necessari.

Desarem els coeficients *BCoefitients* i, en cas d'haver de suprimir pàgines, afegirem a la cel·la *stringPelWarndlg* el text "descriptors".

Desarem tota la matriu *D\_Ranking*, ja que no influeix en el desenvolupament del programa i no és una matriu excessivament extensa.

Si hem creat una matriu d'angles de filtres personalitzats, *cust\_matr\_V*, desarem aquesta matriu i la dels noms dels filtres, *cust\_V*. En cas que sobrepassin les dotze files en suprimirem i afegirem a *stringPelWarndlg* el text "dictionaries".

Desarem també el senyal reconstruït. En aquest cas s'ha de desar tot el senyal.

També desarem la matriu d'espais seleccionats per reconstruir, *subspaceReconstr*, en cas que l'haguem generada. Només agafarem les 12 primeres files.

En cas que tinguem els coeficients seleccionats per una reconstrucció n'agafarem les 12 primeres files i afegirem a *stringPelWarndlg* "selected coefficients".

Aleshores, si hem afegit algun camp a *stringPelWarndlg*, farem un avís per mitjà d'una caixa de diàleg modal que mostri el primer text creat i concateni comes entre cadascun dels textos que hem anat afegint, excepte en l'últim, que hi posarà "and", just abans, i un punt al final. És a dir, que ens pot dir:

*We will save only first 12 decriptors.*

O bé:

*We will save only first 12 dictionaries, subspaces and selected coefficients.*

Si inicialment hi teníem una llegenda la tornem a fer visible amb els seus textos corresponents.

Finalment ho dessem en una estructura de dades i aquesta l'arxivem en un fitxer anomenat *savedSession.mat*.

### 5.4.6. Càrrega de la sessió desada. Funció *MenuLoadSession*

Aquesta funció actualitza les característiques de l'aplicació desades en un fitxer .MAT, els panells oberts, els menús habilitats, els objectes visibles, el fitxer carregat... La funció *MenuLoadSession* només estarà disponible en cas que hi hagi una sessió desada prèviament al directori d'usuari de MATLAB amb el nom *savedSession.mat*.

Comparant els *Tags* carregats amb els camps corresponents canviem les etiquetes dels menús, les cadenes de text de les caixes editables i dels textos fixes, posem els valors dels botons radio corresponents i fem visibles o invisibles els components segons els valors de les variables desades.

Posarem a punt el panell d'extracció *Discriminant Analysis (LDB)* per mitjà de les variables desades *fig.esquerra* i *fig.dreta* i els valors dels botons radio d'aquest panell.

Cridarà la funció de la barra d'eines, *MenuToolbar*, per tal que l'hi posi si en la sessió antiga l'hi teníem o la tregui i centri la finestra si no l'hi teníem.



**Il·lustració 5.9 Barra d'eines personalitzada**

Si a la sessió antiga teníem la taula de rànquing de poder discriminant de diccionaris dibuixarà *Bspaces* i inicialitzarà la taula.

A continuació comprovarà quins eixos han d'estar visibles per saber a quina funció de l'aplicació remetre's: si ha de dibuixar el senyal original, o dibuixar el senyal reconstruït, o reconstruir diccionaris, o veure la representació del senyal segons la classe...

Per últim marcarà els menús que pertoquin i es dirigirà a les funcions que els corresponguin per tal que facin la seva tasca.

## 6. Funcions externes a la interfície

Com ja hem comentat, la interfície gràfica utilitza funcions externes creades amb anterioritat a la realització d'aquest treball. S'han extret dels treballs realitzats per Naoki Saito [3], que s'han agafat com a punt de partida per la realització dels algorismes d'una tesi doctoral en fase de desenvolupament.

En el següent subapartat presentem, com a fonament teòric, un extracte d'aquesta base matemàtica a tall d'informació, procedent d'uns informes entregats periòdicament com a seguiment del projecte europeu AFISA [1][2].

I en el subapartats posteriors analitzarem els arguments i valors de sortida d'aquestes funcions externes a la interfície.

Una funció inclou en la primera línia del fitxer una capçalera on s'especifica quants **valors** retorna, el seu **nom** i quins són els seus **arguments**. Aquesta capçalera s'identifica per la paraula *function*:

```
function
    a=myfun(x,y,varargin)
    ...
end
```

on *a* és el valor de sortida, *myfun* és el nom de la funció i *x*, *y* i *varargin* són els arguments (índexs, matrius, seqüències... d'ús intern que necessita per processar la funció). El cas de l'argument *varargin* és especial, ja que contempla poder rebre un número variable d'arguments.

Les variables contingudes en aquestes funcions són locals a la funció si no s'especifica el contrari, és a dir, no prenen valors de les variables des de l'espai de treball de MATLAB ni d'altres funcions ni els modifiquen.

A l'hora de fer la crida a aquestes funcions els haurem d'enviar els arguments que han d'utilitzar i, si la funció està preparada per retornar valors de sortida, el nom d'una variable on desar-los. Els seus noms no tenen per què coincidir amb els noms definits a la funció. Per exemple, per fer la crida a *myfun* podem fer-ho de la següent manera:

```
b=myfun(x_definida,y_definida)
```

## 6.1. Algorisme de discriminació de trets característics

A continuació presentem un extracte d'uns informes que s'entreguen periòdicament com a seguiment del projecte europeu AFISA [1][2]. Ho incloem en aquest treball com a fonament teòric.

### 6.1.1. Algorisme de discriminació de trets característics – LDB

#### 6.1.1.1. Introducció

Els mètodes *Feature extraction* normalment fan servir tècniques de processament de senyal tradicionals per aconseguir nous senyals amb la mida adequada amb la finalitat de formar un classificador i millorar el seu rendiment.

#### 6.1.1.2. Procediment

Donat un conjunt de senyals i la informació de la classe a la qual pertanyen, usem l'LDB. Aquesta aproximació combina *Wavelet Packet Transform* (WPT) i *Discriminant Analysis* per reduir la dimensió que seleccionen les funcions base (o filtres) que ressaltaran les diferències entre classes.

Primer, l'LDB descompon l'espai  $x_i$  del senyal original utilitzant WPT per obtenir la seva projecció en espais  $\Omega_{j,k,l}$ , calculant els coeficients  $w_{j,k,l}x_i$ , on  $w_{j,k,l}$  és la l-èssima base (mostra) de l'espai denotada per  $j$  i  $k$  ( $j=0, \dots, J$ ;  $k=0, \dots, 2^j-1$ ), essent  $J$  el nivell màxim de descomposició. Aquests espais es poden simbolitzar en forma d'arbre com mostra la il·lustració 6.1. Els mapes d'energia de classe es calculen per mitjà de l'equació

$$\Gamma_{j,k,l}^{(c)} = \frac{\sum_{i=1}^{N_c} (w_{j,k,l}x_i^{(c)})^2}{\sum_{i=1}^{N_c} \|x_i^{(c)}\|^2}, \quad \text{Equació 6.1}$$

on  $\Gamma_c$  denota el mapa d'energia de la classe  $c$ ,  $N_c$  és el número de senyal de cada classe i la notació  $x_i^{(c)}$  s'utilitza per especificar la classe. Després s'acumulen els parells de combinacions de l'entropia simètrica relativa per avaluar discrepàncies entre classes. Aquesta mesura ve donada per la següent equació:

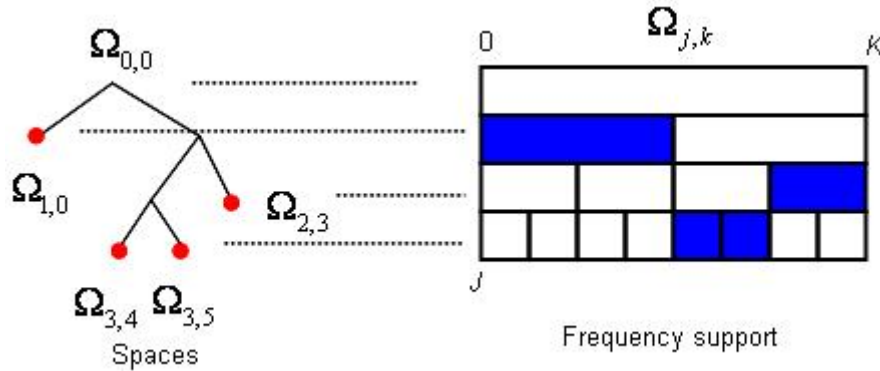
$$D_{j,k,l} = \sum_{i=1}^{C-1} \sum_{j=i+1}^C D(p^{(i)}, p^{(j)}), \quad \text{Equació 6.2}$$

on  $D(\cdot, \cdot)$  és l'entropia simètrica relativa entre dues mostres d'arbre equivalents dels mapes d'energia en dues classes diferents ( $P^{(c)} = \Gamma_{j,k,l}^{(c)}$ )

(equació 6.1)) i  $C$  és el número de classes. En aquest punt, el poder de discriminació del node es calcula per mitjà de

$$D_{j,k} = \sum_{l=0}^{2^{n_0}-1} D_{j,k,l} , \quad \text{Equació 6.3}$$

on  $2^{n_0}$  és la longitud del senyal.



**Il·lustració 6.1 Relació entre la posició en l'arbre WPT i la seva freqüència exemplificada en una selecció de l'espai**

Amb aquesta mesura l'algorisme LDB selecciona els nodes més discriminants de l'arbre (nodes puntejats en vermell de la il·lustració 6.1). Comparem cada node pare amb els seus fills en tots els nivells per obtenir l'espai més discriminant  $A_{0,0}$ .

Els passos de l'algorisme LDB són els següents:

1. Escollim un diccionari (una descomposició wavelet amb els corresponents filtres d'anàlisi passa alt i passa baix).
2. Per cada classe  $c = 1, \dots, C$  construeix mapes d'energia escalats en el temps  $\Gamma_{j,k,l}^{(c)}$  (equació 6.1).
3. Calcula  $D_{j,k}$  en tots els valors de  $l$  (equació 6.3).
4. A continuació estableix, per  $k=0, \dots, 2^J-1$ :
  - a.  $A_{j,k} = B_{j,k}$
  - b.  $\Delta_{j,k} = D_{j,k}$

on  $B_{j,k}$  representa els coeficients escalats en el temps de l'espai  $\Omega_{j,k}$  com a resultat de l'aplicació de WPT (*Wavelet packet transform*)).



5. Finalment, per  $j=J-1, \dots, 0$  i  $k=0, \dots, 2^j-1$ , determina el "millor subconjunt"  $A_{j,k}$ , segons la regla següent:

Si  $\Delta_{j,k} \geq \Delta_{j+1,2k} + \Delta_{j+1,2k+1}$ , aleshores

$$A_{j,k} = B_{j,k}$$

sinó  $A_{j,k} = A_{j+1,2k} \cup A_{j+1,2k+1}$  i  $\Delta_{j,k} = \Delta_{j+1,2k} + \Delta_{j+1,2k+1}$

Les sortides principals de l'algorisme són els coeficients seleccionats  $A_{0,0}$  i el valor de discriminació  $\Delta_{0,0}$  que pot servir per avaluar el diccionari. El senyal d'entrada pot ser reconstruït amb aquests coeficients sense que es perdi informació. No obstant, estem només interessats en la reconstrucció de trets discriminants, per això hi ha un pas addicional en el qual tots els espais d' $A_{0,0}$  s'ordenen d'acord amb els seus valors  $D_{j,k,l}$ .

## 6.1.2. Algorisme de discriminació de trets característics — MLDB

Tot seguit explicarem el procediment de l'MLDB (*Modified LDB*) o LDB (*Local discriminant bases*) amb bancs de filtres paraunitaris personalitzats.

### 6.1.2.1. Procediment

Encara que el mètode WPT contingui una biblioteca rica en diccionaris wavelet que puguem utilitzar per descriure trets característics del senyal, una altra opció és considerar la inclusió de bancs de filtres paraunitaris com a diccionaris wavelet nous "personalitzats" en l'LDB.

Els LDB que fan servir aquests diccionaris s'anomenen MLDB (*Modified Local Discriminant Bases*) ja que la forma dels paquets wavelet s'ajusta de l'arbre. Utilitzem aquesta tècnica per obtenir filtres personalitzats tal com es descriu a continuació.

Els coeficients reals de bancs paraunitaris de dos canals es poden obtenir a través de

$$H_i(z) = H_{i0}(z^2) + z^{-1}H_{i1}(z^2) \quad (i = 0,1), \quad \text{Equació 6.4}$$

on  $H_i(z)$  indica la funció de transferència discreta associada dels filtres *passa-baixes* ( $i=0$ ) i *passa-altes* ( $i=1$ ), i  $H_{i0}(z^2)$ ,  $H_{i1}(z^2)$  són entrades obtingudes de la *matriu polifase* descrita a continuació:

$$H_{pol}(z) = \begin{pmatrix} H_{00}(z) & H_{01}(z) \\ H_{10}(z) & H_{11}(z) \end{pmatrix} = \left( \prod_{k=0}^{K-1} \begin{pmatrix} \cos v_k & \sin v_k \\ -\sin v_k & \cos v_k \end{pmatrix} \begin{pmatrix} 1 & 0 \\ 0 & z^{-1} \end{pmatrix} \right) \begin{pmatrix} \cos v_K & \sin v_K \\ -\sin v_K & \cos v_K \end{pmatrix},$$

**Equació 6.5**

on  $v_k \in [0, \pi)$  ( $k = 0, \dots, K-1$ ), essent  $k$  l'índex del coeficient del filtre FIR d'ordre  $2K+1$  i  $v_k$  obtingut calculant el residu de  $\pi/4 - \sum_{k=0}^{K-1} v_k$  mòdul  $2\pi$  en  $[0, 2\pi)$ .

Assumint això, l'espai

$$P^K = \{v = (v_0, \dots, v_{K-1}) : v_k \in [0, \pi)\}$$

serveix per parametritzar bancs de filtres de dos canals on cada vector d'angle  $v \in P^K$  defineix un diccionari diferent. El poder discriminant del diccionari ve donat per  $\Delta_{0,0}(v)$ , calculat per l'LDB. Així, l'MLDB s'obté estimant  $v$  per mitjà de l'equació

$$\hat{v} = \arg \max_{v \in P^K} \Delta_{0,0}(v)$$

**Equació 6.6**

Per cercar  $\hat{v}$  utilitzem un paràmetre discretitzat en l'espai  $N$  de la forma

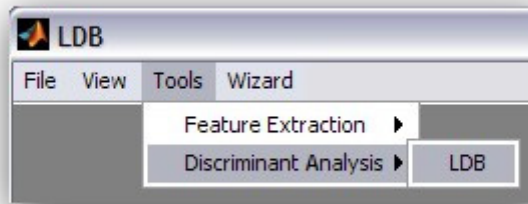
$$P_K^N = \left\{ \left( \frac{\pi}{N} (i_0, \dots, i_{K-1}) : i_k \in (0, \dots, N-1) \right) \right\}$$

Si els paràmetres  $K$  i  $N$  són de mida moderada (per exemple,  $K = 2$  i  $N = 10$ ) aquest mètode és convenient per dades d'una dimensió, com els senyals d'otòlits considerats en aquest projecte.

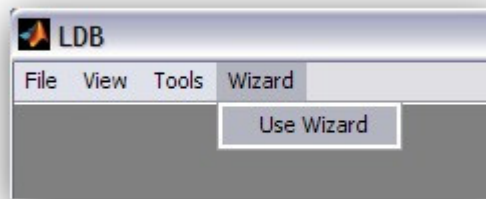
**6.2. Extracció LDB. Funció `f_mlldb2`**

```
[Bspaces, BCoeficients, D_Ranking] = ...
    f_mlldb2(xsignals, ranswer, sJ, varargin)
```

Cridem aquesta funció en dues ocasions: quan utilitzem els diccionaris wavelet en *Discriminant Analysis* i quan els que fem servir són els bancs de filtres paraunitaris personalitzats. Podem utilitzar els diccionaris wavelet per mitjà de la interfície principal o bé per mitjà de l'Assistent (*Wizard*).



**II·lustració 6.2** Crida del panell Discriminant Analysis per mitjà de la interfície principal



**II·lustració 6.3** Crida de l'Assistent (Wizard)

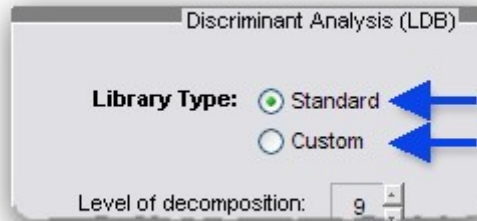
Cal enviar-li, com a primer i segon argument, el senyal ( $xsignals$ ) i el vector de classes ( $ranswer$ ) respectivament, i a continuació el nivell de descomposició que volem ( $sJ$ ).

$$\begin{array}{c}
 \begin{array}{cc|cccc}
 \text{Senyal 1} & \text{Senyal 2} & \text{Senyal 1} & \text{Senyal 2} & \text{Senyal 3} & \text{Senyal 4} \\
 \hline
 x_{11} & x_{12} & y_{11} & y_{12} & y_{13} & y_{14} \\
 \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
 x_{m1} & x_{m2} & y_{m1} & y_{m2} & y_{m3} & y_{m4}
 \end{array} \\
 xsignals = \left[ \begin{array}{cc|cccc}
 \hline
 \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
 \hline
 \end{array} \right]_{m \times n} \\
 \begin{array}{cc|cccc}
 \text{Classe 1} & & \text{Classe 2} & & & \\
 \hline
 1 & 1 & 2 & 2 & 2 & 2
 \end{array} \\
 ranswer = \left[ \begin{array}{cc|cccc}
 \hline
 \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
 \hline
 \end{array} \right]_{1 \times n}
 \end{array}$$

**II·lustració 6.4** Matriu de senyals ( $xsignals$ ) i vector de classes ( $ranswer$ )



la cadena de caràcters '*Cust*', en cas que vulguem *Custom*, o '*Std*', en cas que vulguem *Standard* en el primer argument de *varargin*. Aquesta cadena de caràcters està desada en la variable de handles *fig.StdOrCust*, i s'actualitza cada vegada que canviem de *Standard* a *Custom* per mitjà dels botons radio amb handles *fig.radioStd* i *fig.radioCust*.

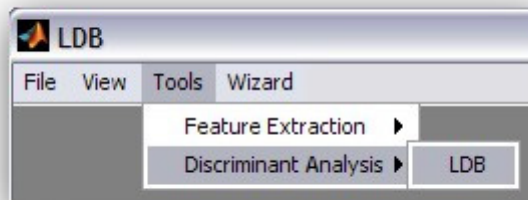


II-lustració 6.7 Botond radio per escollir el tipus d'anàlisi

### 6.2.1. Arguments de Standard Discriminant Analysis

Carregarem un fitxer amb 258 senyals de 512 mostres cadascun repartits en 2 classes. Per tant tindrà 9 nivells de descomposició màxims ( $\log_2(\text{num.mostres})$ ).

Anant al menú LDB, de *Discriminant Analysis*, podem marcar el botó radio *Standard* del panell que apareix i escollir el nivell de descomposició (en l'exemple que seguirem, 6) i els diccionaris que volem utilitzar.



II-lustració 6.8 Menú Discriminant Analysis

També utilitzarà aquesta funció triant els diccionaris per mitjà de l'Assistent.

La funció **f\_mllldb2** contempla d'opció de processar amb tota la llibreria wavelet o només alguns diccionaris. Tot seguit veurem les dues opcions.

**a)** Si escollim tota la llibreria

Podem passar-li, a continuació de l'argument '*Std*', la cadena de caràcters '*All*', per indicar-li que les processis totes.

En el cas que vulguem processar tota la llibreria, i fent-ho des del Workspace, quedaria de la següent manera:

```
[Bspaces, BCoefitients, D_Ranking]=f_mllldb2(...
    xsignals, ...
    ranswer, ...
    6, ...
    'Std', ...
    'All');
```

Però en el cas de l'aplicació ho tenim tot desat en handles i variables globals, per tant, ens quedaria

```
[fileGlobal.Bspaces, ...
    fileGlobal.BCoefitients, ...
    fileGlobal.D_Ranking]=f_mllldb2(
    fileGlobal.xsignals, ...
    fileGlobal.ranswer, ...
    fig.nivellDescomposatsJ, ...
    fig.StdOrCust, ...
    'All');
```

**b)** Si volem processar només amb alguns diccionaris.

Podem entregar-li, a continuació de l'argument **'Std'**, els noms dels diccionaris que hem triat en forma de cadena de caràcters, separats per coma.

Escollint, per exemple, els diccionaris *bior3.1*, *bior3.5*, *db1*, *db4*, *rbio3.1* i *rbio3.5* i el nivell de descomposició 6, tindríem, si ho féssim des del l'àrea de treball de MATLAB, la següent instrucció:

```
[Bspaces, BCoefitients, D_Ranking]=f_mllldb2(...
    xsignals, ...
    ranswer, ...
    6, ...
    'Std', ...
    'bior3.1', ...
    'bior3.5', ...
    'db1', ...
    'db4', ...
    'rbio3.1', ...
    'rbio3.5');
```

Però en el nostre cas, hem hagut de crear 116 variables buides per, tot seguit, omplir-les amb els noms dels diccionaris que hem escollit. Només un número igual de variables al nombre de diccionaris que volem processar tindran una cadena de caràcters, i la resta quedaran buides:

```
X_1 = 'bior3.1'
X_2 = 'bior3.5'
X_3 = 'db1'
X_4 = 'db4'
X_5 = 'rbio3.1'
```

```
X_6 = 'rbio3.5'
X_7 = [] fins a X_116 = []
```

El fitxer *Complete\_dLibrary* conté 117 diccionaris, però només cal crear 116 variables buides perquè en cas que els hi entreguem tots l'aplicació escolliria la instrucció explicada en el punt anterior.

Hem escollit aquest mètode perquè internament la funció compara cadena per cadena, i no accepta cel·les ni estructures.

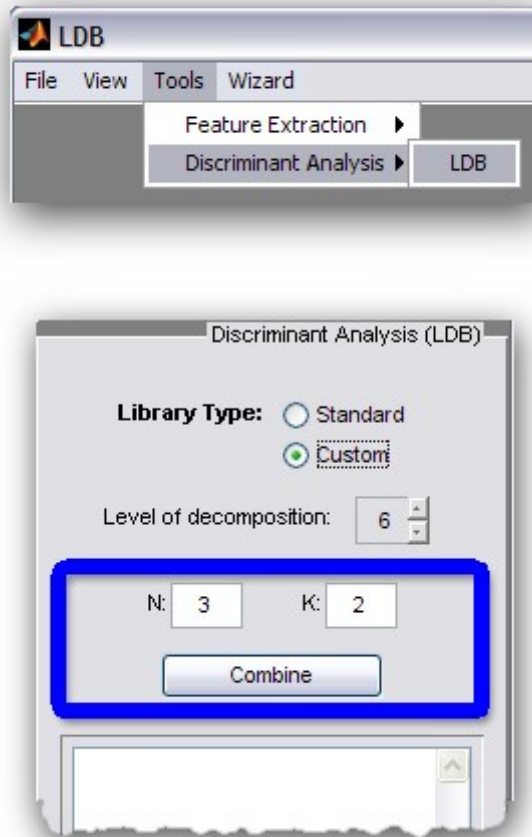
D'aquesta manera, la crida a la funció serà

```
[fileGlobal.Bspaces, ...
 fileGlobal.BCoefitients, ...
 fileGlobal.D_Ranking]=f_mlldb2( ...
 fileGlobal.xsignals, ...
 fileGlobal.ranswer, ...
 fig.nivellDescomposatsJ, ...
 fig.StdOrCust, ...
 X_1 ,X_2 ,X_3 ,X_4 ,X_5 ,X_6 ,X_7 ,X_8 ,X_9 ,X_10, ...
 X_11,X_12,X_13,X_14,X_15,X_16,X_17,X_18,X_19,X_20, ...
 X_21,X_22,X_23,X_24,X_25,X_26,X_27,X_28,X_29,X_30, ...
 X_31,X_32,X_33,X_34,X_35,X_36,X_37,X_38,X_39,X_40, ...
 X_41,X_42,X_43,X_44,X_45,X_46,X_47,X_48,X_49,X_50, ...
 X_51,X_52,X_53,X_54,X_55,X_56,X_57,X_58,X_59,X_60, ...
 X_61,X_62,X_63,X_64,X_65,X_66,X_67,X_68,X_69,X_70, ...
 X_71,X_72,X_73,X_74,X_75,X_76,X_77,X_78,X_79,X_80, ...
 X_81,X_82,X_83,X_84,X_85,X_86,X_87,X_88,X_89,X_90, ...
 X_91,X_92,X_93,X_94,X_95,X_96,X_97,X_98,X_99,X_100, ...
 X_101,X_102,X_103,X_104,X_105,X_106,X_107,X_108,X_109,X_110,...
 X_111,X_112,X_113,X_114,X_115,X_116);
```

### 6.2.2. Arguments de Custom Discriminant Analysis

Carregarem un fitxer amb 258 senyals de 512 mostres cadascun repartits en 2 classes. Per tant tindrà 9 nivells de descomposició màxims ( $\log_2(\text{num.mostres})$ ).

Anant al menú LDB, de *Discriminant Analysis*, podem marcar el botó radio *Custom* del panell que apareix i escollir el nivell de descomposició (en l'exemple que seguirem, 6). Tot seguit introduïm la discretització de l'interval  $[0, \pi)$  a la caixa de text *N*, la longitud de la resposta impulsional (número de paràmetres del filtre -1) a la caixa de text *K* i premem *Combine*.

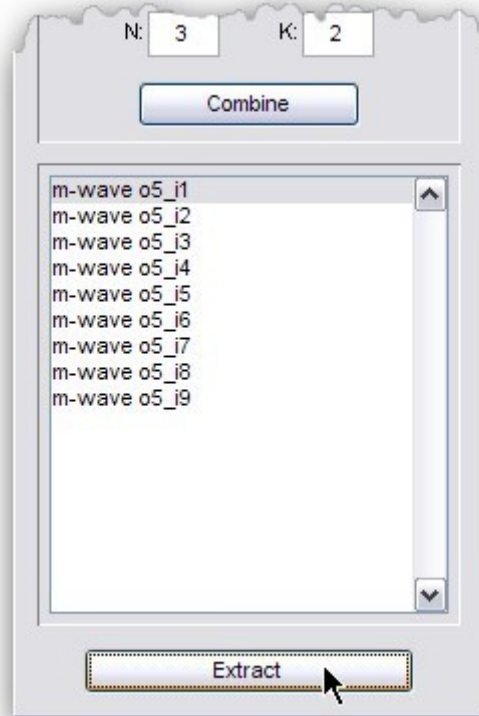


**Il·lustració 6.9** Panell d'entrada de dades per la creació de la matriu d'angles

Amb això generarem la matriu d'angles ***fig.cust\_Matr\_V*** d'ordre  $N^K$  centrada en l'interval  $[0, \pi)$ , que serà el cinquè argument que entregarem a la funció. En l'exemple:

$$\begin{bmatrix} 0 & 0 \\ 0 & 1,04719755 \ 119660 \\ 0 & 2,09439510 \ 239320 \\ 1,04719755 \ 119660 & 0 \\ 1,04719755 \ 119660 & 1,04719755 \ 119660 \\ 1,04719755 \ 119660 & 2,09439510 \ 239320 \\ 2,09439510 \ 239320 & 0 \\ 2,09439510 \ 239320 & 1,04719755 \ 119660 \\ 2,09439510 \ 239320 & 2,09439510 \ 239320 \end{bmatrix}_{9 \times 2}$$





**Il·lustració 6.10 Panell Discriminant Analysis per bancs de filtres paraunitaris personalitzats**

Farem la crida a la funció entregant-li, de la mateixa manera que en *Standard*, la matriu de senyals, el vector de classes i el nivell de descomposició. Seguidament hi posem la cadena '*Cust*', desada a *fig.StdOrCust*, i la matriu d'angles, desada a *fig.cust\_Matr\_V*.

```
[fileGlobal.Bspaces,
 fileGlobal.BCoefitients,
 fileGlobal.D_Ranking]=f_mlldb2(
     fileGlobal.xsignals,
     fileGlobal.ranswer,
     fig.nivellDescomposatsJ,
     fig.StdOrCust,
     fig.cust_Matr_V);
```

### 6.2.3. Valors de sortida

Tots els valors de sortida queden ordenats segons el poder de discriminació de la (mostra). *D\_Ranking* queda ordenat per files i *Bspaces* i *BCoefitients* per pàgines (ordena la tercera dimensió).

La primera columna de *D\_Ranking* conté els noms de les bases escollides i la segona el poder de discriminació de la base. Les seves dimensions seran *número\_de\_llibreriaes\_processades* × 2.

En l'exemple,

D_Ranking	
Nom	PD
llibreria	llibreria
bior3.1	0,1831
bior3.5	0,1572
rbio3.1	0,1586
db4	0,0948
db1	0,0945
rbio3.5	0,0938

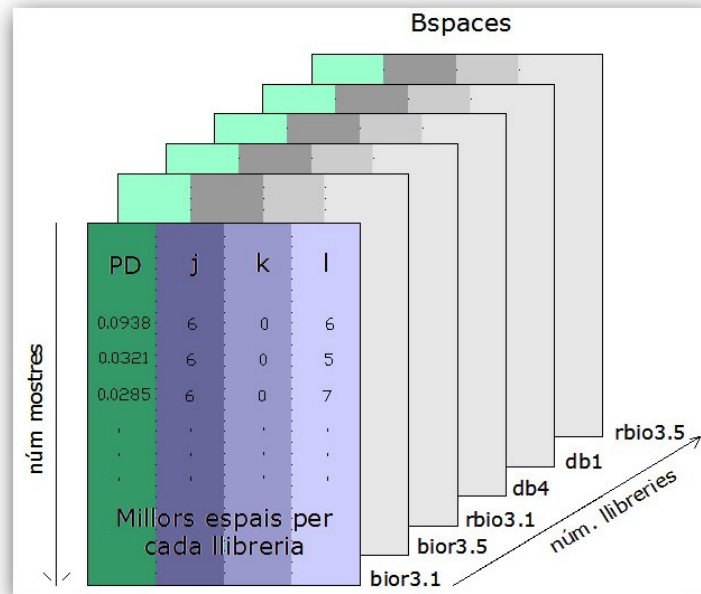
núm. llibreriaes processades ↓

**Il·lustració 6.11 Exemple del valor de sortida D\_Ranking pels diccionaris wavelet**

La primera columna de *Bspaces* ens indica el poder de discriminació de cada node. La resta, de la segona a la quarta columna, ens indica la posició de les bases (mostres). A mida que avancem en les pàgines (la tercera dimensió) d'aquesta variable anem canviant de diccionari segons l'ordre que ens indica la variable *D\_Ranking*, així, cada pàgina conté els nodes d'un diccionari diferent. Les seves dimensions seran

$$\text{número\_de\_mostres} \times 4 \times \text{número\_de\_llibreriaes\_escollides}$$

En l'exemple,



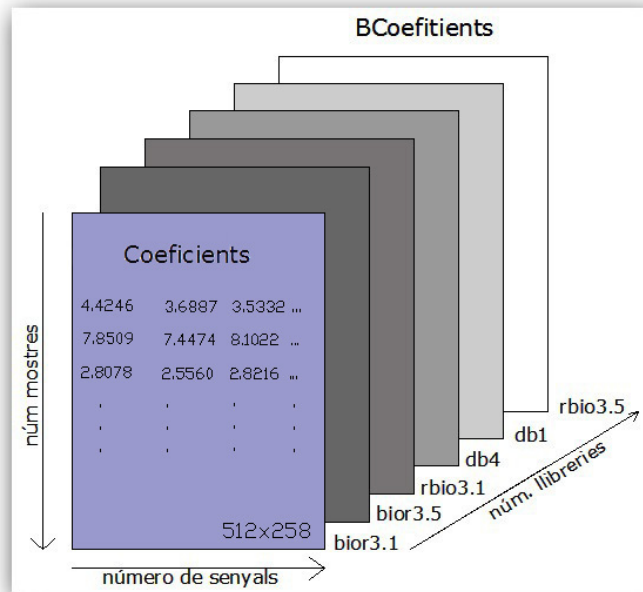
**Il·lustració 6.12 Exemple del valor de sortida Bspaces pels diccionaris wavelet**

La suma del poder discriminant de tots els nodes (d'una pàgina de *Bspaces*) serà el poder discriminant total de la base (segona columna de *D\_Ranking* respecte la fila indicada per la pàgina que haguem agafat en la suma). En l'exemple, la primera pàgina fa referència a *bior3.1*, aleshores,

$$\begin{aligned} \text{sum}(Bspaces(:,1,1)) &= D\_Ranking(1,2) \\ 0.0938 + 0.0321 + 0.0285 + \dots + 0.0000 &= 0.1831 \end{aligned}$$

La matriu *BCoefitients* conté els coeficients creats per aquesta funció. De la mateixa manera que *Bspaces*, cada pàgina conté els coeficients d'una base diferent, amb l'ordre indicat en la variable *D\_Ranking*. Les seves dimensions seran

$$\text{número\_de\_mostres} \times \text{número\_de\_senyals} \times \text{número\_de\_llibreries\_escollides}$$



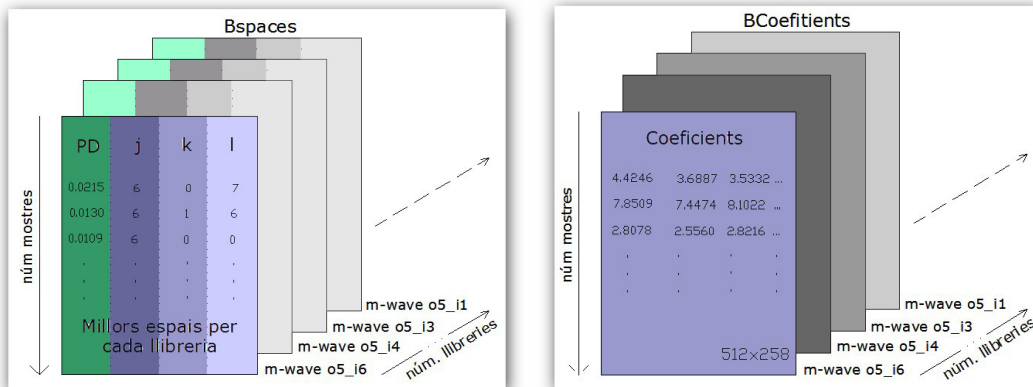
**II-lustració 6.13 Exemple del valor de sortida BCoeficients pels diccionaris wavelet**

En cas que haguem processat un banc de filtres paraunitaris personalitzats les propietats d'aquestes variables seran les mateixes tot i que *D\_Ranking* mostrarà els noms d'aquests filtres, en ordre segons el poder de discriminació de la base, i les pàgines de *Bspaces* i *BCoeficients* correspondran a aquests filtres en l'ordre marcat per *D\_Ranking*.

En l'exemple,

D_Ranking		
	Nom llibreria	PD llibreria
núm. llibreries processades	m-wave o5_i6	0.0950
	m-wave o5_i4	0.0947
	m-wave o5_i3	0.0946
	m-wave o5_i1	0.0945
	m-wave o5_i8	0.0943
	m-wave o5_i7	0.0943
	m-wave o5_i9	0.0943
	m-wave o5_i2	0.0942
	m-wave o5_i5	0.0941

**II-lustració 6.14 Exemple de la sortida D\_Ranking pel banc de filtres paraunitaris personalitzats**



**Il·lustració 6.15 Exemple del valor de sortida Bspaces i BCoefitients pels bancs de filtres paraunitaris personalitzats**

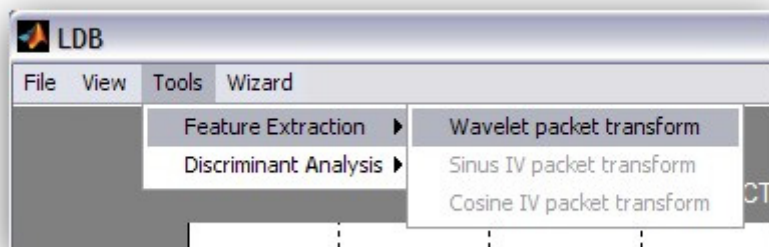
En el programa, aquests tres valors de sortida, queden desades en l'estructura de dades *fileGlobal*: *fileGlobal.D\_Ranking*, *fileGlobal.Bspaces* i *fileGlobal.BCoefitients*, juntament amb el senyal, el vector de classes i el nivell de descomposició màxim.

### 6.3. Extracció Wavelet packet transform. Funció *fWP\_DecTree*

```
fileGlobal.BCoefitients=fWP_DecTree( ...
    fileGlobal.xsignals,           ...
    fig.nivellDescomposatsJ,       ...
    LO_D,HI_D);
```

Cridem aquesta funció quan fem l'extracció *Wavelet packet transform*.

Anant a l'opció *Wavelet packet transform*, del menú *Feature Extraction*, podem escollir el nivell de descomposició (en l'exemple, 6).



**Il·lustració 6.16 Menú Wavelet packet transform**

### 6.3.1. Funció prèvia al pas d'arguments. *wfilters*

```
[LO_D,HI_D]=wfilters(str,'d');
```

Per fer la crida a la funció **fWP\_DecTree** necessitarem els filtres de descomposició passa-altes i passa-baixes (*HI\_D* i *LO\_D*). Els crearem amb la Wavelet Toolbox de MATLAB *wfilters*.

Tindrà la següent sintaxi:

```
[LO_D,HI_D]=wfilters(str,'d');
```

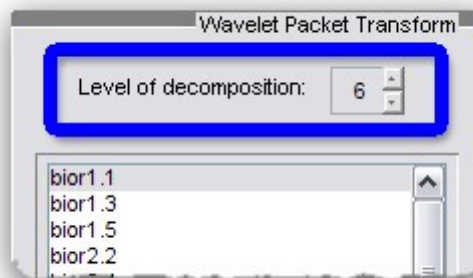
on *str* és una cadena de caràcters que conté el nom del diccionari del qual volem obtenir aquests filtres i *'d'* indica que són filtres de descomposició.

### 6.3.2. Arguments

El primer argument que entrarem a la funció **fWP\_DecTree** és el senyal original (desat a la variable **fileGlobal.xsignals**).

La funció necessita, com a segon argument, el nivell de descomposició al qual volem fer l'extracció. Aquest nivell de descomposició l'entra l'usuari per mitjà de la interfície i queda desat a la variable de handles **fig.nivellDescomposatsJ**.

Com a tercer i quart argument hi enviarem els filtres de descomposició creats prèviament, **LO\_D** i **HI\_D**.



**Il·lustració 6.17** Entrada del nivell de descomposició en l'extracció Wavelet packet transform

Així, la crida a la funció la farem de la següent manera:

```
fileGlobal.BCoefficients=fWP_DecTree( ...
    fileGlobal.xsignals, ...
    fig.nivellDescomposatsJ, ...
    LO_D,HI_D);
```

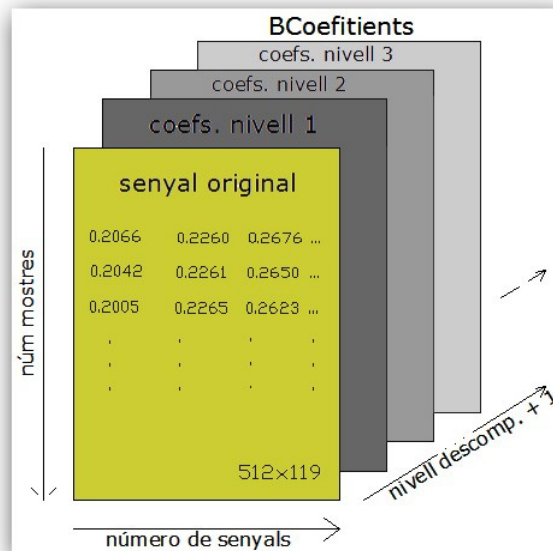
### 6.3.3. Valor de sortida

Desarem el valor de sortida de la funció **fWP\_DecTree** a la variable *fileGlobal.BCoefitients*.

Les dimensions d'aquesta variable seran

$$\text{número\_de\_mostres} \times \text{número\_de\_senyals} \times (\text{nivell\_de\_descomposició} + 1)$$

En la il·lustració que segueix mostrem lles dimensions de la variable *BCoefitients* per un senyal original de 512 mostres i 119 senyals.

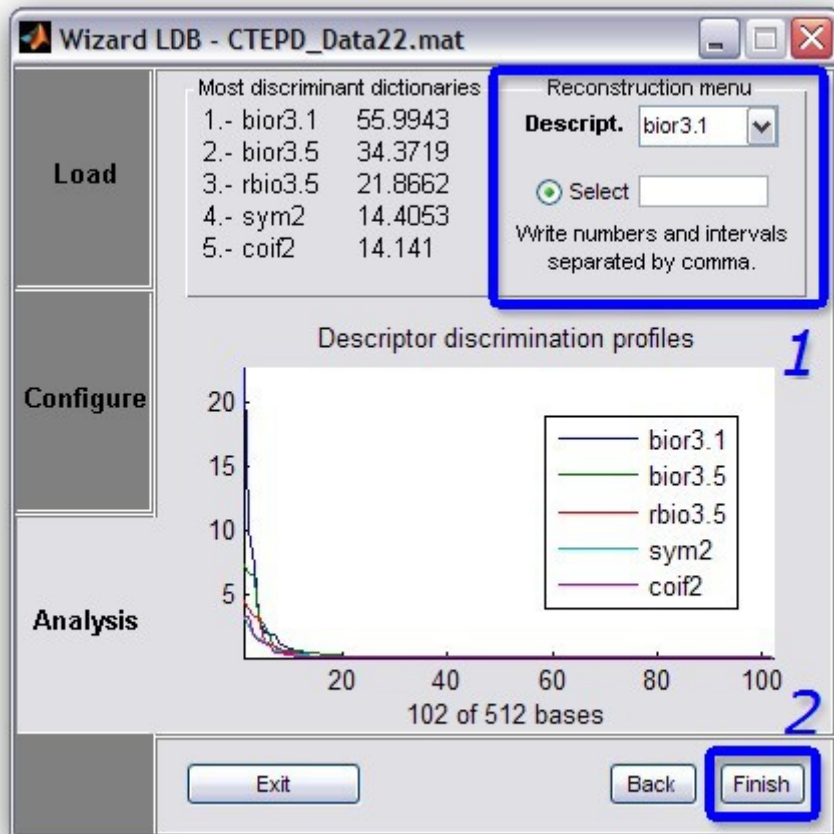


**Il·lustració 6.18 Exemple del valor de sortida BCoefitients de la funció fWP\_DecTree suposant un senyal original amb 119 senyals**

## 6.4. Reconstrucció. Funció fIWPT\_RecTree

```
xAux_Signal=fIWPT_RecTree( ...
    xBestCoefitients,      ...
    xBestspaces,          ...
    nsamples,              ...
    LO_R, HI_R);
```

Cridem la funció **fIWPT\_RecTree** quan volem reconstruir per mitjà dels coeficients. Podem fer-ho a través de la interfície principal, per mitjà d'un panell que apareix un cop feta l'extracció, tan per *Wavelet packet transform* com per *Discriminant analysis*, o bé en finalitzar l'Assistent.



Il·lustració 6.19 Finalització de l'Assistent i crida a la funció `fiWPT_RecTree`

## 6.4.1. Reconstrucció de senyals provinents d'extracció LDB

### 6.4.1.1. Funcions prèvies al pas d'arguments. `wfilters` i `Paraunitary_m`

Per fer la crida a la funció `fiWPT_RecTree` necessitarem els filtres de reconstrucció passa-altes i passa-baixes (`HI_R` i `LO_R`). Els crearem amb les funcions `wfilters` o `Paraunitary_m`.

- En cas que la reconstrucció que vulguem fer sigui d'una extracció feta amb diccionaris wavelet haurem d'aplicar la Wavelet Toolbox de MATLAB **wfilters**. Tindrà la següent sintaxi:

```
[LO_R, HI_R]=wfilters(str, 'r');
```

on `str` és una cadena de caràcters que conté el nom del diccionari de la qual volem obtenir aquests filtres i `'r'` indica que són filtres de reconstrucció.

- En cas que l'extracció prèvia a la reconstrucció hagi estat amb filtres paraunitaris personalitzats cridarem la funció externa `Paraunitary_m`.

```
[H0, H1]=Paraunitary_m(angle, op)
```



La manera com la cridarem serà:

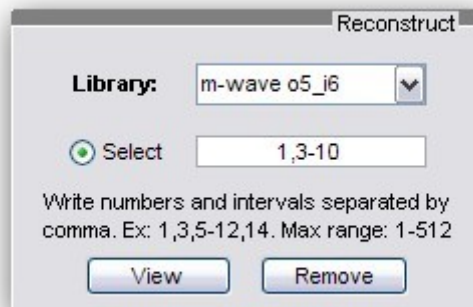
```
[LO_R, HI_R]= ...
Paraunitary_m(strmatch(str,fig.cust_V),'r');
```

- o `str` conté una cadena de caràcters amb el diccionari amb el qual volem reconstruir i a `fig.cust_V` hi ha els noms dels diccionaris generats anteriorment per fer l'extracció per ordre d'obtenció.

És a dir, que fent `strmatch(str,fig.cust_V)` ens donarà la posició del nom del diccionari dins de `fig.cust_V`, és a dir, l'angle que hem d'enviar a la funció `Paraunitary_m`

- o `'r'` indica a la funció que es tracta d'una reconstrucció.

Tot seguit prendrem la cadena de caràcters introduïda a la caixa de text del panell de reconstrucció i hi farem un tractament per mitjà de la funció interna `ComparaCadena` per tal que la retorni en forma de vector.



$$vect = [1 \ 3 \ 4 \ 5 \ 6 \ 7 \ 8 \ 9 \ 10]$$

**Il·lustració 6.20** Panell de reconstrucció i vector que retornarà la funció `ComparaCadena`

#### 6.4.1.2. Arguments per la reconstrucció de Discriminant Analysis

A l'hora d'entregar els arguments `BCoeficients` i `Bspaces` a la funció **fIWPT\_Rectree** li indicarem les files que ha de reconstruir per mitjà del vector `vect` creat amb la funció citada anteriorment, `ComparaCadena`.

Hem de tenir en compte que la primera columna de la variable `Bspaces` és el poder de discriminació de cada node i no ens interessa per reconstruir. Li hem de dir només la posició del node, i això ens ho mostren les columnes 2 ( $j$ ), 3 ( $k$ ) i 4 ( $l$ ) de `Bspaces`.

Li direm quina pàgina (tercera dimensió) ha d'agafar de `BCoeficients` i `Bspaces` per mitjà del valor del menú desplegable on triem el diccionari, ja que estan ordenats de la mateixa manera, de més a menys discriminants. Aquest valor el desarem a la variable `val`.

Per últim li enviarem el nombre de mostres del senyal i els filtres passa-altes i passa-baixes creats anteriorment (*LO\_R* i *HI\_R*).

Com ja hem explicat amb anterioritat, totes aquestes variables estan desades en l'estructura de dades *fileGlobal*, per tant, els noms de les variables seran *fileGlobal.BCoefitients*, *fileGlobal.Bspaces* i *fileGlobal.xsignals*. La crida a la funció la realitzarem de la següent manera:

```
fileGlobal.xAux_Signal=fIWPT_RecTree( ...
    fileGlobal.BCoefitients(vect,:,val)      , ...
    fileGlobal.Bspaces(vect,2:4,val)        , ...
    size(fileGlobal.xsignals,1)             , ...
    LO_R,HI_R);
```

## 6.4.2. Reconstrucció de senyals provinents d'extracció WPT

### 6.4.2.1. Funcions prèvies al pas d'arguments. *fgetCoefitient* i *wfilters*

Per fer la crida a la funció **fIWPT\_RecTree** necessitarem els filtres de reconstrucció passa-altes i passa-baixes (***HI\_R*** i ***LO\_R***). Els crearem amb la Wavelet Toolbox de MATLAB *wfilters*. Tindrà la següent sintaxi:

```
[LO_R,HI_R]=wfilters(str,'r');
```

on *str* és una cadena de caràcters que conté el nom del diccionari del qual volem obtenir aquests filtres i *'r'* indica que són filtres de reconstrucció.

Per agafar els subespais que volem reconstruir s'ha fet dues funcions internes (*ButtonAddToReconstr*, que adjunta un sol espai, i *ButtonAddNodeToReconst*, que adjunta tot el node) que els posen en una matriu anomenada ***fileGlobal.subspaceReconstr***, de 3 columnes i un número de files igual al número d'espais afegits.

A diagram showing a selection matrix with three columns labeled 'j', 'k', and 'l'. The rows are labeled 'núm d'espais seleccionats' on the left. The values in the matrix are:

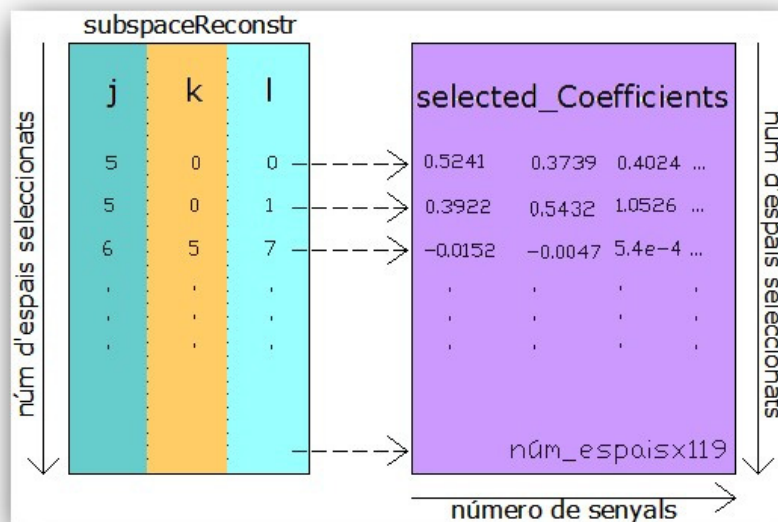
	j	k	l
5	0	0	0
5	0	0	1
6	5	5	7
.	.	.	.
.	.	.	.
.	.	.	.

**II-lustració 6.21 Matriu de subespais seleccionats**

Els arguments de la funció **fgetCoefitient** seran la matriu d'espais seleccionats i la matriu de coeficients, i ens retornarà els coeficients seleccionats:

```
fileGlobal.selected_Coefficients=fgetCoefitient(...
    fileGlobal.BCoefitients,fileGlobal.subspaceReonstr);
```

La matriu d'espais i la de coeficients seleccionats tindran el mateix nombre de files, tot i que la matriu de coeficients seleccionats tindrà tantes columnes com senyals té el senyal original.



**II-lustració 6.22 Matriu d'espais seleccionats i de coeficients seleccionats suposant un senyal original amb 119 senyals**

### 6.4.2.2. Arguments per la reconstrucció de Wavelet packet transform

Si l'extracció feta prèviament l'hem realitzada per mitjà de *Wavelet packet transform*, haurem d'enviar a **fIWPT\_RecTree** els coeficients seleccionats per la funció *fgetCoefitient* (variable *fileGlobal.selected\_Coefficients*), la matriu d'espais seleccionats (variable *fileGlobal.subspaceReconstr*), la mida del senyal original (prendrà la mida de la variable *fileGlobal.BCoefitients*, els coeficients del senyal) i els filtres passa-altes i passa-baixes corresponents al diccionari wavelet segons el qual volem reconstruir (variables *LO\_R* i *HI\_R* els filtres passa-baixes i passa-altes respectivament).

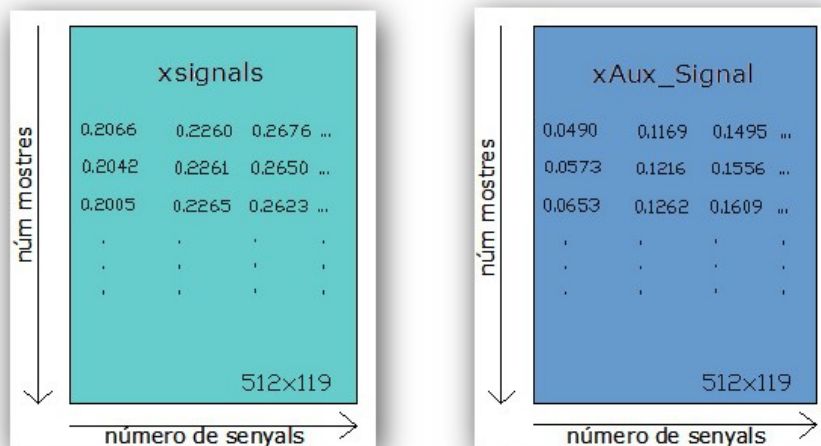
De manera que la crida que haurem de fer serà

```
fileGlobal.xAux_Signal=fIWPT_RecTree( ...
    fileGlobal.selected_Coefficients, ...
    fileGlobal.subspaceReconstr, ...
    size(fileGlobal.BCoefitients,1), ...
    LO_R,HI_R);
```

### 6.4.3. Valor de sortida

El valor que retorna la funció **fIWPT\_RecTree** el desarem a la variable *xAux\_Signal*, dins de l'estructura de dades *fileGlobal*.

Les seves dimensions són *número\_de\_mostres* × *número\_de\_senyals*. Coincideixen amb les dimensions del senyal original.



**Il·lustració 6.23** Dimensions la matriu de senyals i la de reconstrucció

Aquesta matriu està ordenada per mostres de més a menys discriminant.

## 7. Conclusions

Finalitzat aquest treball podem dir que hem arribat als objectius marcats inicialment.

Podem agregar nous paquets a aquesta aplicació per tal d'ampliar-la, però haurem de respectar l'estructura de dades global que conté les variables referents al senyal, com el del propi senyal, el vector de classes i el nivell màxim de descomposició, així com totes les variables associades a aquests (els espais escollits i el rànquing de diccionaris creats per *Discriminant Analysis*, els coeficients, i el senyal reconstruït, el subespai seleccionat i els coeficients seleccionats creats durant la reconstrucció).

Encara que l'aplicació tingui els elements col·locats en la posició presentada podem retocar-los per tal de fer les interfícies més amigables a l'usuari.

Gràcies a aquesta aplicació podem tenir una idea intuïtiva de com actuen les transformades wavelet amb diferents tipus de senyals. És a dir, observem que la descomposició en els nodes superiors, els més propers a l'espai original, recull molta informació local (temporal, en cas dels senyals d'acceleròmetre, o espacial, en cas de senyals d'otòlits). En els nodes posteriors, els més allunyats de l'espai original, la descomposició porta més informació en el domini freqüencial, similar a una transformada de Fourier.

Això mostra el potencial de l'anàlisi wavelet, on es combina la informació en freqüència i en temps o espai.

A nivell més personal, un dels punts que han costat més de dur a terme ha estat la taula amb la barra de desplaçament, ja que quan deixem una casella de verificació activada i movem la barra, la casella ha de seguir la caixa de text corresponent encara que aquesta desaparegui. A més, hi hem creat 13 caixes de text que s'actualitzen amb cada moviment de la barra de desplaçament.

Un altre punt conflictiu ha estat a l'hora de comunicar les diferents finestres per tal que es transmetessin la informació de la interfície. Per passar com a argument la informació de la finestra que fa la crida no hi teníem problemes, però a l'hora de retornar la informació de la finestra cridada no s'actualitzaven els handles. Ho hem resolt per mitjà de variables globals.

Un punt bàsic que s'hauria de retocar en futures ampliacions de la interfície és, precisament, treure les variables globals i comunicar les interfícies per mitjà dels handles, ja que aquest tipus de variables no són adients en una aplicació MATLAB.

També es pot integrar l'Assistent amb la interfície principal per tal de no repetir funcions i poder-les reutilitzar. En total en reutilitzaríem 4 de control de les llistes i 1 per comparar la cadena entrada a la reconstrucció, les dues que dibuixen els senyals i les que controlen el panell de *Discriminant Analysis*.

## 8. Bibliografia

- [1] IFREMER. "Automated FISH Ageing". Projecte núm 044132. (31/3/2007). Deliverable 2.4.a.
- [2] IFREMER. "Automated FISH Ageing". Projecte núm 044132. (31/3/2007). Deliverable 2.4.b.
- [3] SAITO, NAOKI, *et al.* "Local discriminant bases". *Wavelet Applications in Signal and Image Processing II*. San Diego, 1994. Pàgs: 2-14
- [4] TUSSET, VICTOR M. "Otolith atlas for the western mediterranean, north and central eastern atlantic". *Scientia Marina* 72S1 (2008).

### Adreces d'Internet

- [5] PARISI-BARADAD, V. *et al* (2005). "Otolith shape contour analysis using affine transformation invariant wavelet transforms and curvature scale space representation". *Marine and Freshwater Research* 56 (2005): 795-804.  
<http://www.icm.csic.es/aforo/publications/Parisi-et-al-2005-MFR.pdf>.  
(Consulta: 23 de juliol de 2008)
- [6] MATLAB CENTRAL. *MATLAB newsgroup*.  
<http://www.mathworks.com/matlabcentral/newsreader/> (Consulta: 6 de juny de 2008)
- [7] THE MATHWORKS. *MATLAB*. 02 de febrer de 2008.  
<http://www.mathworks.com/access/helpdesk/help/techdoc/matlab.html>  
(Consulta: 17 de juliol de 2008)
- [8] THE MATHWORKS. *Product Support - Techniques for Debugging MATLAB M-files*.  
<http://www.mathworks.com/support/tech-notes/1200/1207.html>  
(Consulta: 23 de juliol de 2008)

# **ANNEX**

## **A**

**Interfície principal,  
funció LDB\_GUI**

```

1  function LDB_GUI (varargin)
2  % LDB_GUI Utilitat per LDB
3
4  %-----
5  % Tasques d'inicialització i
6  %-----
7
8  % Variables globals per desar les dades processades
9  global fileGlobal fileNameGlob filePathNameGlob ContourOrProfile
10
11 fileGlobal=struct; % Inicialitzem variables
12 filePathNameGlob=char;
13 fileNameGlob=char;
14 ContourOrProfile=char;
15 % Generarem una estructura 'fig' per passar als Callbacks i
16 % emmagatzemar-ho.
17 dadesFig=struct(...
18     'IntegerHandle'    , 'off', ... % Figure No. no és un enter
19     'MenuBar'         , 'none', ...
20     'ToolBar'         , 'none', ...
21     'Resize'          , 'off', ... % No maximitzar ni minimitzar
22     'NumberTitle'     , 'off',...
23     'Tag'              , 'figLDB',...
24     'Interruptible'   , 'off',...
25     'BusyAction'      , 'cancel',...
26     'Color', [0.5 0.5 0.5]);
27
28
29 %-----
30 % Construcció de components
31 %-----
32 % Creem la primera figura i els seus components principals
33 % Les propietats de la figura no inclouen els marges, per tant usem 5
34 % pixels als costats i 22 a dalt i a baix
35 fig.bdwidth = 5;
36 topbdwidth = 22;
37
38 set(0, 'Units', 'pixels') % ens assegurem que està en pixels
39 fig.scnsize = get(0, 'Screenize'); % agafem la mida de la pantalla
40
41 finestra = [fig.bdwidth, ... % [X
42     0*fig.scnsize(4) + 2*topbdwidth, ... % Y
43     fig.scnsize(3) - 2*fig.bdwidth, ... % Amplada
44     fig.scnsize(4) - 4*(topbdwidth + fig.bdwidth)]; % Alçada]
45 clear topbdwidth
46
47 fig.figLDB = figure(dadesFig, ...
48     'MenuBar', 'none', ... % amaga el menú estàndard
49     'Name', 'LDB', ... % el nom de la figura serà LDB
50     'NumberTitle', 'off', ... % treu el número de figura (figure: 1)
51     'Position', finestra, ... % la posiciona a 'finestra'
52     'Resize', 'off', ... % no ens deixarà canviar-li la mida
53     'Units', 'Pixels', ...
54     'CloseRequestFcn', @LDB_CloseRequestFcn); % la mesurarà en píxels
55
56 % Mesures dels panels
57 % Tots mesuren la mateixa amplada: amplada de la finestra menys

```



```

58 % amplada del panel, i tots estaran a la mateixa posició X, només
59 % variarà la Y.
60 fig.panelsAmplada=finestra(3)/4.3149; % Amplada
61
62 fig.panelsX=(finestra(3)-fig.panelsAmplada); % Posició X
63
64 % Alçada panels de la dreta
65 % L'alçada estarà en proporció de la finestra. Per la posició Y de
66 % cadascun es resta la seva alçada de la mida de la finestra:
67 % inicialment estaran tots a dalt de tot.
68 fig.panelSignalAlçada=finestra(4)/3.7288; % Alçada
69 fig.panelPlotAlçada=finestra(4)/4.4;
70 fig.panelDiscrimAnalysisMenuAlçada=finestra(4)-fig.panelSignalAlçada
71
72 fig.panelSignalY=(finestra(4)-fig.panelSignalAlçada); % Posició Y
73 fig.panelPlotY=(finestra(4)-fig.panelPlotAlçada);
74 fig.panelDiscrimAnalysisMenuY=(finestra(4)- ...
75     fig.panelDiscrimAnalysisMenuAlçada);
76
77 % ---Panel Signal
78 midaPanelSignal=[fig.panelsX      fig.panelSignalY ...
79                  fig.panelsAmplada fig.panelSignalAlçada];
80 fig.uipanelSignal=uipanel('Parent',fig.figLDB,... % Pare del panel
81     'Title','Signal Properties',...           % Títol del panel
82     'TitlePosition','righttop',...
83     'Units','Pixels',...
84     'Position',[fig.panelsX      fig.panelSignalY      ...
85                 fig.panelsAmplada fig.panelSignalAlçada],...
86     'Tag','uipanelSignal',...
87     'BorderType','beveledout');
88
89 % Panels dins del PanelSignal
90 % Per deixar un espai al marge superior i inferior del panel igual a la
91 % mida dels panels farem:
92 % NumPanels=5:
93 % uipanel1: [marge (alçadaPanel/(NumPanels+2)*(NumPanels-0) ...]
94 % uipanel2: [marge (alçadaPanel/(NumPanels+2)*(NumPanels-1) ...]
95 % uipanel3: [marge (alçadaPanel/(NumPanels+2)*(NumPanels-2) ...]
96 % uipanel4: [marge (alçadaPanel/(NumPanels+2)*(NumPanels-3) ...]
97 % uipanel5: [marge (alçadaPanel/(NumPanels+2)*(NumPanels-4) ...]
98
99 np=5; % Número de panels
100 for i=1:5
101     uip=strcat('uipanel',num2str(i));
102     fig.(char(ui))=uipanel('Parent',fig.uipanelSignal,...
103         'Units','Pixels',...
104         'Position',[10      ...
105                     midaPanelSignal(4)/7*np ...
106                     (midaPanelSignal(3)-20) ...
107                     20      ],...
108         'Tag',uiip,...
109         'BorderType','line');
110     np=np-1;
111 end
112 % Elements dins d'aquets panels
113 midaPanelsPropietats=[10      midaPanelSignal(4)/7*5 ...
114     (midaPanelSignal(3)-20) 20      ];

```

```

115 % Amplada: serà la meitat del panel i se li ha de restar la X inicial
116 % per posicionar-lo correctament
117 texts=struct('Style','text',...
118     'Position',[7 2 midaPanelsPropietats(3)/2-7 14],...
119     'Enable','off');
120 fig.textSignalName=icontrol(fig.uipanel1,...
121     texts,'String','Signal Name: ','Tag','textSignalName');
122 fig.textNumberSignals=icontrol(fig.uipanel2,...
123     texts,'String','Number of Signals: ','Tag','textNumberSignals');
124 fig.textCoefficients=icontrol(fig.uipanel3,...
125     texts,'String','Coefficients: ','Tag','textCoefficients');
126 fig.textVectorClass=icontrol(fig.uipanel4,...
127     texts,'String','Vector Class: ','Tag','textVectorClass');
128 fig.textLevels=icontrol(fig.uipanel5,...
129     texts,'String','Maximum Levels: ','Tag','textLevels');
130 % En els textos que prendran les dades del fitxer cal fer:
131 % X: Hi sumarem 2 pixels per separar-lo de l'altre text.
132 % Amplada: serà la meitat del panel i se li ha de restar la X inicial
133 % més un marge perquè no se sol·lapi amb el panel.
134 vars=struct(...
135     'Style','text',...
136     'Position',[midaPanelsPropietats(3)/2+2 2 ...
137         midaPanelsPropietats(3)/2-4 14],...
138     'Enable','off');
139 fig.varSignalName=icontrol(fig.uipanel1,vars,'Tag','varSignalName');
140 fig.varNumberSignals=icontrol(fig.uipanel2,vars, ],...
141     'Tag','varNumberSignals');
142 fig.varCoefficients=icontrol(fig.uipanel3,vars,'Tag','varCoefficients');
143 fig.varVectorClass=icontrol(fig.uipanel4,vars,'Tag','varVectorClass');
144 fig.varLevels=icontrol(fig.uipanel5,vars,'Tag','varLevels');
145
146 % Netegem les variables que ja no necessitarem:
147 clear midaPanelsPropietats midaPanelSignal vars uip
148
149 % ---Panel Plot
150 fig.uipanelPlot=uipanel('Parent',fig.figLDB,...
151     'Title','Signal representation',...
152     'TitlePosition','righttop',...
153     'Units','Pixels',...
154     'Position',[fig.panelsX ...
155         (fig.panelPlotY-fig.panelSignalAlcada) ...
156         fig.panelsAmplada ...
157         fig.panelPlotAlcada ],...
158     'BorderType','beveledout',...
159     'Tag','uipanelPlot',...
160     'Visible','off');
161
162 % Elements del panelPlot
163 % Primera fila
164 nf=5; % Numero files que volem
165 atGN=85; % Amplada del textGroupNumber
166 apGN= 75; % Amplada del popupGroupNumber
167 mee=14; % Marge entre elements
168
169 % 'mitjana': Resta entre l'amplada del panel i tots els elements que
170 %posarem a la fila
171 mitjana=fig.panelsAmplada-(atGN+apGN+mee);

```

```

172
173 fig.textGroupName=icontrol(fig.uipanelPlot,...
174     'Style','text',...
175     'String','Class Number',...
176     'FontWeight','bold',...
177     'Enable','off',...
178     'Tag','textGroupName',...
179     'Position',[mitjana/2 ...
180                 (fig.panelPlotAlcada/(nf+2)*(nf-0)+16/2) ...
181                 atGN ...
182                 16 ]);
183 fig.popupGroupName=icontrol(fig.uipanelPlot,...
184     'Style','popupmenu',...
185     'String','1',...
186     'Tag','popupGroupName',...
187     'Position',[(mitjana/2+atGN+mee) ...
188                 (fig.panelPlotAlcada/(nf+2)*(nf-0)+22/4) ...
189                 apGN ...
190                 22 ], ...
191     'BackgroundColor','w',...
192     'enable','off',...
193     'Callback',@PopupGroupName);
194 % Netegem les variables que ja no necessitarem:
195 clear atGN apGN mee mitjana
196
197 % Segona fila
198 mtc=2; % Marge entre text i caixa
199 mat=25; % Amplada text
200 mac=50; % Amplada caixa editable
201 m=20; % Marge entre parells de text i caixa (text-caixa- marge+marge -
202     % text-caixa)
203 fig.textFirst=icontrol(fig.uipanelPlot,...
204     'Style','text',...
205     'String','First:',...
206     'Enable','off',...
207     'Tag','textFirst',...
208     'Position',[((fig.panelsAmplada-m)/2-mac-mat-mtc) ...
209                 (fig.panelPlotAlcada/(nf+2)*(nf-1)) ...
210                 mat ...
211                 15 ]);
212 fig.textLast=icontrol(fig.uipanelPlot,...
213     'Style','text',...
214     'String','Last:',...
215     'Enable','off',...
216     'Tag','textLast',...
217     'Position',[((fig.panelsAmplada+m)/2) ...
218                 (fig.panelPlotAlcada/(nf+2)*(nf-1)) ...
219                 mat ...
220                 15 ]);
221 fig.editFirst=icontrol(fig.uipanelPlot,...
222     'Style','edit',...
223     'Enable','off',...
224     'Tag','editFirst',...

```

```

225     'Position', [(fig.panelsAmplada-m)/2-mac)      ...
226                (fig.panelPlotAlcada/(nf+2)*(nf-1)) ...
227                mac                                ...
228                17                                ], ...
229     'BackgroundColor','w',...
230     'Callback',@EditFirst);
231 fig.editLast=uicontrol(fig.uipanelPlot,...
232     'Style','edit',...
233     'Enable','off',...
234     'Tag','editLast',...
235     'Position', [(fig.panelsAmplada+m)/2+mtc+mat)    ...
236                (fig.panelPlotAlcada/(nf+2)*(nf-1)) ...
237                mac                                ...
238                17], ...
239     'BackgroundColor','w',...
240     'Callback',@EditLast);
241 % Netegem les variables que ja no necessitarem
242 clear mtc mat mac m nf
243
244 % Tercera fila: uipanelChecks
245 fig.margeP=12; % Marge dels panels interiors a l'exterior
246 fig.midaAmpPanInt=fig.panelsAmplada-12*2; % amplada panels interiors
247
248 % Farem que la posició de 'uipanelChecks' depengui de la posició
249 % de 'editLast'
250 p=get(fig.editLast,'Position');
251
252 fig.uipanelChecks=uipanel('Parent',fig.uipanelPlot,...
253     'Units','Pixels',...
254     'Tag','uipanelChecks',...
255     'Position',[fig.margeP ((p(2)-20)/2) fig.midaAmpPanInt 44],...
256     'BorderType','beveledin');
257
258 % Components de uipanelChecks
259 mac=61; % amplada checks
260 mec=2; % Marge entre checks
261 mal=15; % Alçada dels checks
262
263 p=get(fig.uipanelChecks,'Position');
264
265 fig.checkGrid=uicontrol(fig.uipanelChecks,...
266     'Style','checkbox',...
267     'String','Grid',...
268     'Enable','off',...
269     'Tag','checkGrid',...
270     'Position',[(p(3)/2-mac-mec) (p(4)/2+(p(4)/2-mal)/2) mac mal],...
271     'Callback',@CheckGrid);
272 fig.checkZoom=uicontrol(fig.uipanelChecks,...
273     'Style','checkbox',...
274     'String','Zoom',...
275     'Enable','off',...
276     'Tag','checkZoom',...
277     'Position',[(p(3)/2+mec) (p(4)/2+(p(4)/2-mal)/2) mac mal],...
278     'Callback',@CheckZoom);
279 fig.checkPan=uicontrol(fig.uipanelChecks,...

```

```

280     'Style','checkbox',...
281     'String','Pan',...
282     'Enable','off',...
283     'Tag','checkPan',...
284     'Position',[(p(3)/2-mac-mec) ((p(4)/2-mal/2)/2) mac mal],...
285     'Callback',@CheckPan);
286 fig.checkCursor=uicontrol(fig.uipanelChecks,...
287     'Style','checkbox',...
288     'String','Cursor',...
289     'Enable','off',...
290     'Tag','checkCursor',...
291     'Position',[(p(3)/2+mec) ((p(4)/2-mal/2)/2) mac mal],...
292     'Callback',@CheckCursor);
293
294 % Netegem les variables que ja no necessitarem
295 clear mac mec mal
296
297 % Quarta fila: botons View i Remove
298 abV=fig.panelsAmplada*5/16; % Amplada dels botons View i Remove
299 mee=14; % Marge entre elements
300
301 % 'mitjana': Resta entre l'amplada del panel i tots els elements que
302 % posarem a la fila
303 mitjana=fig.panelsAmplada-(abV*2+mee);
304
305 fig.buttonView=uicontrol(fig.uipanelPlot,...
306     'Style','pushbutton',...
307     'String','View',...
308     'Enable','off',...
309     'Tag','buttonView',...
310     'Position',[mitjana/2                                ...
311                 (p(2)/2-(20/2))                        ...
312                 abV                                     ...
313                 20                                     ],...
314     'Callback',@ButtonView);
315
316 fig.buttonRemove=uicontrol(fig.uipanelPlot,...
317     'Style','pushbutton',...
318     'String','Remove',...
319     'Enable','off',...
320     'Tag','buttonRemove',...
321     'Position',[(mitjana/2+abV+mee)                    ...
322                 (p(2)/2-(20/2))                        ...
323                 abV                                     ...
324                 20                                     ],...
325     'Callback',@ButtonRemove);
326
327 % Netegem les variables que ja no necessitarem
328 clear p
329
330 % ---panelDiscrimAnalysisMenu
331 fig.uipanelDiscrimAnalysis=uipanel('Parent',fig.figLDB,...
332     'Title','Discriminant Analysis (LDB)',...
333     'TitlePosition','righttop',...

```

```

334     'Units','Pixels',...
335     'Tag','uipanelDiscrimAnalysis',...
336     'Position',[fig.panelsX ...
337                 (fig.panelDiscrimAnalysisMenuY-fig.panelSignalAlcada) ...
338                 fig.panelsAmplada ...
339                 fig.panelDiscrimAnalysisMenuAlcada ], ...
340     'BorderType','beveledout',...
341     'Visible','off');
342
343     % Elements del uipanelDiscrimAnalysis
344     m=5; % Marge entre text i radios
345     mtl=72; % Amplada text LibType i radios
346     aR=15; % Alçada radioCust
347     fig.textLibType=icontrol(fig.uipanelDiscrimAnalysis,...
348         'Style','text',...
349         'String','Library Type:',...
350         'FontWeight','bold',...
351         'Enable','off',...
352         'Tag','textLibType',...
353         'Position',[fig.panelsAmplada/2-mtl-m ...
354                     fig.panelDiscrimAnalysisMenuAlcada*8/9 ...
355                     mtl ...
356                     aR ...
357                     ]);
358     fig.radioStd=icontrol(fig.uipanelDiscrimAnalysis,...
359         'Style','radiobutton',...
360         'String','Standard',...
361         'Enable','off',...
362         'Tag','radioStd',...
363         'TooltipString','Choose the library type', ...
364         'Position',[fig.panelsAmplada/2+m ...
365                     fig.panelDiscrimAnalysisMenuAlcada*8/9 ...
366                     mtl ...
367                     aR ...
368                     ],...
369         'Callback',@RadioStd);
370     fig.radioCust=icontrol(fig.uipanelDiscrimAnalysis,...
371         'Style','radiobutton',...
372         'String','Custom',...
373         'Enable','off',...
374         'Tag','radioCust',...
375         'TooltipString','Choose the library type',...
376         'Position',[fig.panelsAmplada/2+m ...
377                     fig.panelDiscrimAnalysisMenuAlcada*8/9-aR-aR/3 ...
378                     mtl ...
379                     aR ...
380                     ],...
381         'Callback',@RadioCust);
382     % Netegem les variables que ja no necessitarem
383     clear m mtl aR
384
385     % Panel Custom i elements
386     fig.uipanelCustNK=uipanel('Parent',fig.uipanelDiscrimAnalysis,...
387         'Units','Pixels',...
388         'Tag','uipanelCustNK',...

```

```

387     'Position', [fig.margeP                                     ...
388                 fig.panelDiscrimAnalysisMenuAlcada*22/37 ...
389                 fig.midaAmpPanInt                             ...
390                 fig.panelDiscrimAnalysisMenuAlcada/6.5 ], ...
391     'BorderType', 'beveledin', ...
392     'Visible', 'off');
393
394     % Obtenim les mides i posició de uipanelCustNK
395     p=get(fig.uipanelCustNK, 'Position');
396
397     mat=p(3)*0.047;      % amplada text
398     mac=p(3)*0.17;      % Amplada caixa editable
399     m=p(3)*0.07;        % Marge entre parells de text i caixa (text-caixa-
400                        % marge+marge -text-caixa)
401     mtc=p(3)*0.01;      % Marge entre text i caixa
402     maltx=p(4)*0.225;   % Alçada text
403     maled=p(4)*0.315;  % Alçada edit
404     mAlB=p(4)-maled-m*1.5; % Alçada boto
405
406     fig.txtN=icontrol(fig.uipanelCustNK, ...
407         'Style', 'text', ...
408         'String', 'N:', ...
409         'Tag', 'txtN', ...
410         'TooltipString', 'Sampling of interval [0,pi]', ... % Discretització
411         'Position', [(fig.midaAmpPanInt/2-mac-mat-mtc-m) ...
412                     (p(4)/2+maltx/2) ...
413                     mat ...
414                     maltx ...
415                     ]);
416     fig.entraN=icontrol(fig.uipanelCustNK, ...
417         'Style', 'edit', ...
418         'Tag', 'entraN', ...
419         'Position', [(fig.midaAmpPanInt/2-mac-m) ...
420                     (p(4)/2+maled/4) ...
421                     mac ...
422                     maled ...
423                     ], ...
424         'BackgroundColor', 'w', ...
425         'TooltipString', 'Sampling of interval [0,pi]', ...
426         'Callback', @EntraN);
427
428     fig.txtK=icontrol(fig.uipanelCustNK, ...
429         'Style', 'text', ...
430         'String', 'K:', ...
431         'Tag', 'txtK', ...
432         'TooltipString', 'Number of parameters of filter - 1', ...
433         'Position', [(fig.midaAmpPanInt/2+m) ...
434                     (p(4)/2+maltx/2) ...
435                     mat ...
436                     maltx ...
437                     ]);
438     fig.entraK=icontrol(fig.uipanelCustNK, ...
439         'Style', 'edit', ...
440         'Tag', 'entraK', ...
441         'Position', [(fig.midaAmpPanInt/2+m+mtc+mat) ...
442                     (p(4)/2+maled/4) ...
443                     mac ...

```

```

441         maled                                     ],...
442     'BackgroundColor','w',...
443     'TooltipString','Number of parameters of filter - 1',...
444     'Callback',@EntraK);
445
446 mAb=(p(3)-100); % Amplada botó
447 fig.buttonCreateComb=icontrol(fig.uipanelCustNK,...
448     'Style','pushbutton',...
449     'String','Combine',...
450     'Tag','buttonCreateComb',...
451     'Position',[((fig.midaAmpPanInt-mAb)/2) ...
452                 ((p(4)/2-mAlB) ...
453                 mAb ...
454                 23 ...
455                 ],...
456     'Callback',@ButtonCreateComb);
457 % Netegem les variables que ja no necessitem
458 clear mat mac m mtc maltx maled mAlB mAb
459 % Panel Listboxes i elements
460 % Panel pels listboxes del Discriminant (LDB)
461 fig.uipanelListboxes=uipanel('Parent',fig.uipanelDiscrimAnalysis,...
462     'Units','Pixels',...
463     'Tag','uipanelListboxes',...
464     'Position',[fig.margeP ...
465                 (p(2)-fig.panelDiscrimAnalysisMenuAlcada/2) ...
466                 fig.midaAmpPanInt ...
467                 fig.panelDiscrimAnalysisMenuAlcada/2-10 ...
468                 ],...
469     'BorderType','beveledin',...
470     'Title','Libraries',...
471     'TitlePosition','centertop ',...
472     'Visible','off');
473 % Obtenim les mides i posició de uipanelListboxes
474 p=get(fig.uipanelListboxes,'Position');
475
476 mee=3; % Marge entre elements interns: ListBoxes-mee-botons
477 mde=6; % Marge dret, esquerre, dalt i baix entre listboxes i panel.
478 mbp=16; % Amplada del botó petit, botoAfegeix i botoTreu
479
480 % Fem una petita equació per trobar l'amplada dels ListBoxes:
481 % midaPanelAmplada=mde+midaListBoxes+mee+mbp+mee+midaListBoxes+mde
482 % midaPanelAmplada=2*mde+2*midaListBoxes+2*mee+mbp (...)
483 % midaListBoxes=(midaPanelAmplada-mbp-2*(mde+mee))/2
484 ampLB=(p(3)-mbp-2*(mde+mee))/2; % Amplada de ListBoxes
485
486 abT=14; % Alçada dels botons botoTots i botoCap
487 alcLB=(p(4)-2*mde-abT-mee*4); % Alçada dels Listboxs
488
489 fig.listBoxEntrada=icontrol(fig.uipanelListboxes,...
490     'Style','listbox',...
491     'Tag','listboxEntrada',...
492     'Position',[mde (mde+abT+mee) ampLB alcLB],...
493     'BackgroundColor','w');
494
495 fig.listBoxSortida=icontrol(fig.uipanelListboxes,...

```



```

496     'Style','listbox',...
497     'Tag','listboxSortida',...
498     'Position',[(mde+ampLB+2*mee+mbp) (mde+abT+mee) ampLB alcLB],...
499     'enable','off',...
500     'BackgroundColor','w');
501
502 fig.botoTots=uicontrol(fig.uipanelListboxes,...
503     'Style','pushbutton',...
504     'String','All',...
505     'Tag','botoTots',...
506     'Position',[mde mde ampLB abT],...
507     'Callback',@BotoTots);
508 fig.botoCap=uicontrol(fig.uipanelListboxes,...
509     'Style','pushbutton',...
510     'String','None',...
511     'Tag','botoCap',...
512     'enable','off',...
513     'Position',[(mde+ampLB+2*mee+mbp) mde ampLB abT],...
514     'Callback',@BotoCap);
515
516 fig.botoAfegeix=uicontrol(fig.uipanelListboxes,...
517     'Style','pushbutton',...
518     'String','>',...
519     'Tag','botoAfegeix',...
520     'FontWeight','bold',...
521     'Position',[(mde+ampLB+mee)           ...
522                 ((p(4)+mee+abT)/2+mee/2) ...
523                 mbp                       ...
524                 alcLB/3                   ],...
525     'Callback',@BotoAfegeix);
526 fig.botoTreu=uicontrol(fig.uipanelListboxes,...
527     'Style','pushbutton',...
528     'String','<',...
529     'Tag','botoTreu',...
530     'FontWeight','bold',...
531     'Position',[(mde+ampLB+mee)           ...
532                 ((p(4)+mee+abT)/2-alcLB/3-mee/2) ...
533                 mbp                       ...
534                 alcLB/3                   ],...
535     'Enable','off',...
536     'Callback',@BotoTreu);
537
538 % Netegem les variables que ja no necessitarem
539 clear mee mde mbp ampLB abT alcLB
540
541 fig.buttonProcess=uicontrol(fig.uipanelDiscrimAnalysis,...
542     'Style','pushbutton',...
543     'String','Extract',...
544     'Tag','buttonProcess',...
545     'Position',[((fig.panelsAmplada-fig.panelsAmplada*7/10)/2) ...
546                 (p(2)/2-(22/2)) ...
547                 fig.panelsAmplada*7/10 ...
548                 23 ...
549                 ],...

```

```

549     'Visible','off',...
550     'Callback',{@ButtonProcess,fig});
551 clear p
552
553 % Panel Nivell de descomposició i elements
554 pC=get(fig.uipanelCustNK,'Position');
555 posY=pC(2)+pC(4)+2;
556
557 pRC=get(fig.radioCust,'Position');
558 mX=pRC(2)-2-posY;
559
560 fig.uipanelNivellDescomp=uipanel('Parent',fig.uipanelDiscrimAnalysis,...
561     'Units','Pixels',...
562     'Tag','uipanelNivellDescomp',...
563     'Position',[fig.margeP      ...
564                 posY          ...
565                 fig.midaAmpPanInt ...
566                 mX              ],...
567     'BorderType','none',...
568     'Visible','off');
569
570 mat=pC(3)*0.55;      % amplada text
571 % mac=pC(3)*0.17;    % Amplada slider
572 m=pC(3)*0.05;       % Marge entre text i text
573 % mtc=pC(3)*0.01;   % Marge entre text i text
574 maltx=pC(4)*0.23;  % Alçada text
575 maled=pC(4)*0.315; % Alçada text, slider i edit
576 mtv=pC(3)*0.15;    % amplada text variable
577 mac=m; %Amplada slider
578
579 fig.txtNivellDescomp=uicontrol(fig.uipanelNivellDescomp,...
580     'Style','text',...
581     'Tag','txtNivellDescomp',...
582     'Units','Pixels',...
583     'String','Level of decomposition:',...
584     'Position',[(fig.midaAmpPanInt-(mat+m+mtv+mac))/2 ...
585                 (mX-maltx)/2 ...
586                 mat ...
587                 maltx ...
588                 ]);
589
590 fig.entraNivellDescomp=uicontrol(fig.uipanelNivellDescomp,...
591     'Style','edit',...
592     'Tag','entraNivellDescomp',...
593     'String','6',...
594     'Units','Pixels',...
595     'Position',[(fig.midaAmpPanInt-(mat+m+mtv+mac))/2+mat+m ...
596                 (mX-maled)/2 ...
597                 mtv ...
598                 maled],...
599     'Callback',@EntraNivellDescomp);
600
601 fig.sliderNivellDescomp=uicontrol(fig.uipanelNivellDescomp,...
602     'Style','slider',...
603     'Tag','sliderNivellDescomp',...
604     'Min',1,...
605     'Max',6,...
606     'SliderStep',[.2 0.1],... % [pas_a_la_fletxa pas_a_l'indicador]

```

```

604     'Value', 6, ...
605     'Units', 'Pixels', ...
606     'Position', [(fig.midaAmpPanInt - (mat+m+mtv+m+mac))/2+mat+m+mtv ...
607                 (mX-maled)/2 ...
608                 mac ...
609                 maled], ...
610     'Callback', @SliderNivellDescomp);
611
612 % ---Feature Extraction -> Wavelet Packet Transform
613 fig.uipanelWaveletPT=uipanel('Parent', fig.figLDB, ...
614     'Title', 'Wavelet Packet Transform', ...
615     'TitlePosition', 'righttop', ...
616     'Units', 'Pixels', ...
617     'Tag', 'uipanelWaveletPT', ...
618     'Position', [fig.panelsX ...
619                 (fig.panelDiscrimAnalysisMenuY-fig.panelSignalAlcada) ...
620                 fig.panelsAmplada ...
621                 fig.panelDiscrimAnalysisMenuAlcada], ...
622     'BorderType', 'beveledout', ...
623     'Visible', 'off');
624 % Panel que contindrà un listbox
625 pbP=get(fig.buttonProcess, 'Position');
626 fig.uipanelListboxFEWPT=uipanel('Parent', fig.uipanelWaveletPT, ...
627     'Units', 'Pixels', ...
628     'Tag', 'uipanelListboxFEWPT', ...
629     'Position', [fig.margeP ...
630                 pbP(2)+pbP(4)+pbP(2) ...
631                 fig.midaAmpPanInt ...
632                 fig.panelDiscrimAnalysisMenuAlcada- ...
633                 2*(pbP(2)+pbP(4)+pbP(2)+...
634                 pbP(4)/2)], ... %marge
635     'BorderType', 'beveledin', ...
636     'Visible', 'on');
637
638 mde=6; % Marge dret, esquerre, dalt i baix entre listboxes i panel.
639
640 % Mida dels ListBoxes:
641 % midaListBoxes=midaPanel-2*(mde)
642 % Amplada de ListBox:
643 ampLB=(fig.midaAmpPanInt-2*mde);
644 % Alçada dels Listboxes:
645 alcLB=(fig.panelDiscrimAnalysisMenuAlcada-...
646     2*(pbP(2)+pbP(4)+pbP(2)+pbP(4)/2+mde));
647
648 fig.listboxFEWPT=uicontrol(fig.uipanelListboxFEWPT, ...
649     'Style', 'listbox', ...
650     'Tag', 'listboxFEWPT', ...
651     'Position', [mde mde ampLB alcLB], ...
652     'BackgroundColor', 'w');
653 % Panel Nivell de descomposició i elements
654
655 posY=pbP(2)+pbP(4)+pbP(2)+... % posició Y panel + Alçada + marge
656     (fig.panelDiscrimAnalysisMenuAlcada-2*(pbP(2)+pbP(4)+pbP(2)+ ...
657     pbP(4)/2)+mde);

```

```

658
659 mX=fig.panelDiscrimAnalysisMenuAlcada-posY-2*mde;
660
661 fig.uipanelNivellDescompWPT=uipanel('Parent',fig.uipanelWaveletPT,...
662     'Units','Pixels',...
663     'Tag','uipanelNivellDescompWPT',...
664     'Position',[fig.margeP      ...
665                 posY           ...
666                 fig.midaAmpPanInt ...
667                 mX             ],...
668     'BorderType','none',...
669     'Visible','on');
670
671 mat=pC(3)*0.55;      % amplada text
672 % mac=pC(3)*0.17;   % Amplada slider
673 m=pC(3)*0.05;      % Marge entre text i text
674 % mtc=pC(3)*0.01;  % Marge entre text i text
675 maltx=pC(4)*0.23; % Alçada text
676 maled=pC(4)*0.315; % Alçada text, slider i edit
677 mtv=pC(3)*0.15;   % amplada text variable
678 mac=m; %Amplada slider
679
680 fig.txtNivellDescompWPT=uicontrol(fig.uipanelNivellDescompWPT,...
681     'Style','text',...
682     'Tag','txtNivellDescompWPT',...
683     'Units','Pixels',...
684     'String','Level of decomposition:',...
685     'Position',[(fig.midaAmpPanInt-(mat+m+mtv+mac))/2 ...
686                 (mX-maltx)/2           ...
687                 mat                   ...
688                 maltx                  ]);
689 fig.entraNivellDescompWPT=uicontrol(fig.uipanelNivellDescompWPT,...
690     'Style','edit',...
691     'Tag','entraNivellDescompWPT',...
692     'String','6',...
693     'Units','Pixels',...
694     'Position',[(fig.midaAmpPanInt-(mat+m+mtv+mac))/2+mat+m ...
695                 (mX-maled)/2           ...
696                 mtv                     ...
697                 maled],...
698     'Callback',@EntraNivellDescompWPT);
699 fig.sliderNivellDescompWPT=uicontrol(fig.uipanelNivellDescompWPT,...
700     'Style','slider',...
701     'Tag','sliderNivellDescompWPT',...
702     'Min',1,...
703     'Max',6,...
704     'SliderStep',[.2 0.1],... % [pas_a_la_fletxa pas_a_l'indicador]
705     'Value',6,...
706     'Units','Pixels',...
707     'Position',[(fig.midaAmpPanInt-(mat+m+mtv+m+mac))/2+mat+m+mtv ...
708                 (mX-maled)/2           ...
709                 mac                     ...
710                 maled],...
711     'Callback',@SliderNivellDescompWPT);

```

```

712
713 % Boto per processar Wavelet Packet transform
714 p=get(fig.buttonProcess,'Position');
715 fig.buttonProcessWPT=uicontrol(fig.uipanelWaveletPT,...
716     'Style','pushbutton',...
717     'Tag','buttonProcessWPT',...
718     'String','Extract',...
719     'Position',p,...
720     'Visible','on',...
721     'Callback',{@ButtonProcessWPT,fig});
722
723 clear ampLB pbP mde alcLB mat m maltx maled mtv mac mX posY p pC pRC
724
725 % ---Axes
726 margeAltAxis=57; % Marge des de Y=0 als axes
727 margeCostatAxis=88; % Marge des de X=0 als axes
728
729 axisAmplada=finestra(3)-fig.panelsAmplada-2*margeCostatAxis; % Amplada
730 axisAlcada=finestra(4)-2*margeAltAxis; % Alçada
731
732 fig.axesPlot=axes('Parent',fig.figLDB,...
733     'Units','Pixels',...
734     'Position',[margeCostatAxis ...
735                 margeAltAxis ...
736                 axisAmplada ...
737                 axisAlcada ],...
738     'HandleVisibility','callback',...
739     'Visible','off');
740 fig.axesFix=axes('Parent',fig.figLDB,...
741     'Units','Pixels',...
742     'Position',[margeCostatAxis ...
743                 margeAltAxis ...
744                 axisAmplada ...
745                 axisAlcada ],...
746     'HandleVisibility','callback',...
747     'Visible','off');
748
749 fig.axesPolar=axes('Parent',fig.figLDB,...
750     'Units','Pixels',...
751     'Position',[margeCostatAxis ...
752                 margeAltAxis ...
753                 axisAmplada ...
754                 axisAlcada ],...
755     'HandleVisibility','callback',...
756     'Visible','off');
757 % Per fer els contorns
758 op=get(fig.axesFix,'outerposition');
759 p=get(fig.axesFix,'position');
760 mee=(p(2)-op(2))/2; % Marge entre elements
761 fig.axesPolarTots=axes('Parent',fig.figLDB,...
762     'Units','Pixels',...
763     'Position',[margeCostatAxis ...
764                 margeAltAxis+(axisAlcada/2)+mee ...
765                 axisAmplada ...

```

```

766         axisAlcada/2-mee/2     ],...
767     'HandleVisibility','callback',...
768     'Visible','off');
769 fig.axesSignals_VSxAux=axes('Parent',fig.figLDB,...
770     'Units','Pixels',...
771     'Position',[margeCostatAxis ...
772         margeAltAxis+axisAlcada/4+mee ...
773         axisAmplada ...
774         axisAlcada/4-mee     ],...
775     'Visible','off');
776 fig.axesSignals_VSxAuxRec=axes('Parent',fig.figLDB,...
777     'Units','Pixels',...
778     'Position',[margeCostatAxis ...
779         margeAltAxis+axisAlcada/4+mee ...
780         axisAmplada ...
781         axisAlcada/4-mee     ],...
782     'Visible','off');
783 fig.axesxAux_VSsignals=axes('Parent',fig.figLDB,...
784     'Units','Pixels',...
785     'Position',[margeCostatAxis ...
786         margeAltAxis ...
787         axisAmplada ...
788         axisAlcada/4-mee     ],...
789     'Visible','off');
790 % Per fer el Profile
791 fig.axesSignals_VSxAuxProfile=axes('Parent',fig.figLDB,...
792     'Units','Pixels',...
793     'Position',[margeCostatAxis ...
794         margeAltAxis+axisAlcada/2+mee ...
795         axisAmplada ...
796         axisAlcada/2-mee     ],...
797     'Visible','off');
798 fig.axesSignals_VSxAuxProfileRec=axes('Parent',fig.figLDB,...
799     'Units','Pixels',...
800     'Position',[margeCostatAxis ...
801         margeAltAxis+axisAlcada/2+mee ...
802         axisAmplada ...
803         axisAlcada/2-mee     ],...
804     'Visible','off');
805 fig.axesxAux_VSsignalsProfile=axes('Parent',fig.figLDB,...
806     'Units','Pixels',...
807     'Position',[margeCostatAxis ...
808         margeAltAxis ...
809         axisAmplada ...
810         axisAlcada/2-mee     ],...
811     'Visible','off');
812
813 % Panel de taula ranking
814 mee=3;
815 % Ho desem en una variable perquè s'inicialitzarà cada vegada.
816 % La taula és activa i la seva mida pot quedar modificada.
817 fig.midaPanelTaula=[(finestra(3)-fig.panelsAmplada-400)/2 ...
818     margeAltAxis ...

```

```

819             400      ...
820             (axisAlcada/2-mee)    ];
821 fig.panelTaula=uipanel(fig.figLDB,...
822     'Units','Pixels',...
823     'Tag','panelTaula',...
824     'BorderType','beveledin',...
825     'Position',fig.midaPanelTaula,...
826     'Visible','off',...
827     'BackgroundColor',[0.5 0.5 0.5],...
828     'BorderType','none');
829 ch=fig.midaPanelTaula(3)*0.05;
830 mf=fig.midaPanelTaula(3)*0.09; % mida dels noms de les files
831 mac=(fig.midaPanelTaula(3)-mf-mf/2-ch-2*mf)/2; % Amplada caixa
832     % editable. 2*mf serà els edits del JKL i mf/2 l'slider
833 maled=fig.midaPanelTaula(4)/14; % Alçada edit
834 fig.midaTitol=[mf ...
835     (fig.midaPanelTaula(4)/14*13)      ...
836     mac                                ...
837     maled                                ];
838 fig.titol1=uicontrol(fig.panelTaula,...
839     'Style','edit',...
840     'Enable','inactive',...
841     'String','Library',...
842     'Position',fig.midaTitol,...
843     'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
844
845 fig.titol2=uicontrol(fig.panelTaula,...
846     'Style','edit',...
847     'Enable','inactive',...
848     'String','Power Discrim.',...
849     'Position',[fig.midaTitol(1)+mac ...
850     fig.midaTitol(2)      ...
851     fig.midaTitol(3)      ...
852     fig.midaTitol(4)     ],...
853     'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
854
855
856 % Creem 13 editText, un a sota l'altre, que tindran el número
857 % del ranking (de la fila)
858 v=13;
859 for i=1:13
860     strRL=strcat('files',num2str(i));
861     fig.(char(strRL))=uicontrol(fig.panelTaula,...
862     'Style','edit',...
863     'Enable','inactive',...
864     'String',i,...
865     'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'),...
866     'Position',[0 ...
867     (fig.midaPanelTaula(4)/14*(v-1))      ...
868     mf      ...
869     maled   ]);
870     v=v-1;
871 end
872 % Creem 13 editText, un a sota l'altre, que tindran el nom

```

```

873 v=13;
874 for i=1:13
875     strRL=strcat('rankLibrary',num2str(i));
876     fig.(char(strRL))=uicontrol(fig.panelTaula,...
877         'Style','edit',...
878         'Enable','inactive',...
879         'BackgroundColor','w',...
880         'Position',[mf ...
881             (fig.midaPanelTaula(4)/14*(v-1)) ...
882             mac ...
883             maled ...
884             v=v-1;
885 end
886 % Creem 13 editText, un a sota l'altre, que tindran el valor
887 v=13;
888 for i=1:13
889     strRL=strcat('rankPowDisc',num2str(i));
890     fig.(char(strRL))=uicontrol(fig.panelTaula,...
891         'Style','edit',...
892         'Enable','inactive',...
893         'BackgroundColor','w',...
894         'Position',[mf+mac ...
895             (fig.midaPanelTaula(4)/14*(v-1)) ...
896             mac ...
897             maled ...
898             v=v-1;
899 end
900 v=13;
901 for i=1:13
902     strCH=strcat('checkMostraJKL',num2str(i));
903     fig.(char(strCH))=uicontrol(fig.panelTaula,...
904         'Style','checkbox',...
905         'String','',...
906         'Enable','on',...
907         'Visible','off',...
908         'BackgroundColor',[0.5 0.5 0.5],...
909         'Tag',strCH,...
910         'Position',[fig.midaPanelTaula(3)-2*mf-ch ...
911             (fig.midaPanelTaula(4)/14*(v-1)) ...
912             ch ...
913             maled],...
914         'Callback',@CheckMostraJKL);
915     v=v-1;
916 end
917 fig.slideTaula=uicontrol(fig.panelTaula,...
918     'Style','slider',...
919     'Min',1,...
920     'Max',13,...
921     'SliderStep',[.2 0.1],... % [pas_a_la_fletxa pas_a_l'indicador]
922     'Value',13,... % Com que queda vertical, va al revés
923     'Units','Pixels',...
924     'Position',[fig.midaPanelTaula(3)-2*mf-mf/2-ch ...
925         0 ...

```



```

926         mf/2
927         fig.midaPanelTaula(4)],...
928     'Callback',@SlideTaula);
929
930 fig.titol3=uicontrol(fig.panelTaula,...
931     'Style','edit',...
932     'Enable','inactive',...
933     'String','j,k,l',...
934     'Position',[fig.midaPanelTaula(3)-2*mf ...
935         fig.midaTitol(2) ...
936         2*mf ...
937         maled ],...
938     'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
939
940 fig.mostraJKL=uicontrol(fig.panelTaula,...
941     'Style','edit',...
942     'Enable','Inactive',...
943     'Tag','mostraJKL',...
944     'BackgroundColor','w',...
945     'Position',[fig.midaPanelTaula(3)-2*mf ...
946         (fig.midaPanelTaula(4)/14*12) ...
947         2*mf ...
948         maled ]);
949
950 % Netegem les variables que ja no necessitarem
951 clear margeAltAxis margeCostatAxis axisAmplada axisAlcada op
952
953 % ---barra Wait
954 waitAmplada=finestra(3)-fig.panelsAmplada;
955 waitAlt=60;
956 fig.textWait=uicontrol(fig.figLDB,...
957     'Style','text',...
958     'Tag','textWait',...
959     'String','Please wait...!',...
960     'FontSize',35,... % Mida de la lletra
961     'ForegroundColor',[0 0 0],... % Color de la lletra
962     'HorizontalAlignment','Center',... % Alineació centrada
963     'Units','Pixels',...
964     'Position',[0 0 waitAmplada waitAlt],...
965     'Visible','off');
966
967 % Netegem les variables que ja no necessitarem
968 clear waitAmplada waitAlt
969
970 % Menús-----
971 % Menú File
972 fig.menuFile = uimenu(fig.figLDB,...
973     'Label','File',... % Etiqueta
974     'Tag','menuFile',...
975     'Callback',@MenuFile);
976 fig.menuLoad= uimenu(fig.menuFile,...
977     'Tag','menuLoad',...
978     'Label','&Load',...
979     'Callback',@MenuLoad);

```

```

980     fig.menuSave= uimenu(fig.menuFile,...
981         'Label','&Save',...
982         'Tag','menuSave',...
983         'Callback',@MenuSave);
984     %---%
985     fig.menuLoadSession=uimenu(fig.menuFile,...
986         'Label','Load session',...
987         'Tag','menuLoadSession',...
988         'Separator','on',...
989         'Callback',@MenuLoadSession);
990     fig.menuSaveSession= uimenu(fig.menuFile,...
991         'Label','Save session',...
992         'Tag','menuSaveSession',...
993         'Callback',@MenuSaveSession);
994     fig.menuCloseSession= uimenu(fig.menuFile,...
995         'Tag','menuCloseSession',...
996         'Label','Close session',...
997         'Enable','off',...
998         'Callback',@MenuCloseSession);
999     %---%
1000    fig.menuExport=uimenu(fig.menuFile,...
1001        'Label','Export',...
1002        'Tag','menuExport',...
1003        'Separator','on',...
1004        'Callback',@MenuExport);
1005    %---%
1006    fig.menuExit= uimenu(fig.menuFile,...
1007        'Label','&Exit',...
1008        'Tag','menuExit',...
1009        'Separator','on',... % Separador que precedirà l'element
1010        'Callback',@MenuExit);
1011    % Menu View
1012    fig.menuView = uimenu(fig.figLDB,...
1013        'Label','View',...
1014        'Tag','menuView',...
1015        'Callback',@MenuView);
1016    fig.menuToolbar=uimenu(fig.menuView,...
1017        'Label','Show Toolbar',...
1018        'Tag','menuToolbar',...
1019        'Callback',@MenuToolbar);
1020    fig.menuPlot= uimenu(fig.menuView,...
1021        'Tag','menuPlot',...
1022        'Label','Signal &Representation');
1023    fig.menuPolar=uimenu(fig.menuPlot,...
1024        'Label','Polar',...
1025        'Tag','menuPolar',...
1026        'Checked','off',...
1027        'Tag','menuPolar',...
1028        'Callback',@MenuPolar);
1029    fig.menuShowHidePlot=uimenu(fig.menuPlot,...
1030        'Tag','menuShowHidePlot',...
1031        'Label','Show plot menu',...
1032        'Callback',@MenuShowHidePlot);
1033    % Menú Tools

```

```

1034 fig.menuTools = uimenu(fig.figLDB, ...
1035     'Label', 'Tools', ...
1036     'Tag', 'menuTools', ...
1037     'Callback', @MenuTools);
1038
1039 %---%
1040 fig.menuFeatureExtraction=uimenu(fig.menuTools, ...
1041     'Tag', 'menuFeatureExtraction', ...
1042     'Label', 'Feature Extraction');
1043 fig.menuWaveletPT=uimenu(fig.menuFeatureExtraction, ...
1044     'Tag', 'menuWaveletPT', ...
1045     'Label', 'Wavelet packet transform', ...
1046     'Callback', @MenuWaveletPT);
1047 % En faltaran 2 per fer:
1048 fig.menuSinusIVPT=uimenu(fig.menuFeatureExtraction, ...
1049     'Tag', 'menuSinusIVPT', ...
1050     'Enable', 'off', ... % de moment a 'enable off'
1051     'Label', 'Sinus IV packet transform');
1052 % i:
1053 fig.menuCosinusIVPT=uimenu(fig.menuFeatureExtraction, ...
1054     'Tag', 'menuCosinusIVPT', ...
1055     'Enable', 'off', ... % de moment a 'enable off'
1056     'Label', 'Cosine IV packet transform');
1057
1058 fig.menuDiscrimAnalysis=uimenu(fig.menuTools, ...
1059     'Tag', 'menuDiscrimAnalysis', ...
1060     'Label', 'Discriminant Analysis');
1061 fig.menuLDB= uimenu(fig.menuDiscrimAnalysis, ...
1062     'Label', 'LDB', ...
1063     'Tag', 'menuLDB', ...
1064     'Callback', @MenuLDB);
1065 %---%
1066
1067 % Menu Wizard
1068 fig.menuWizard=uimenu(fig.figLDB, ...
1069     'Tag', 'menuWizard', ...
1070     'Label', 'Wizard');
1071     fig.menuOpenWizard=uimenu(fig.menuWizard, ...
1072         'Tag', 'menuOpenWizard', ...
1073         'Label', 'Use &Wizard', ...
1074         'Callback', @MenuOpenWizard);
1075
1076 % ---Panel Reconstruct
1077 fig.uipanelReconstruct=uipanel('Parent', fig.figLDB, ...
1078     'Title', 'Reconstruct', ...
1079     'TitlePosition', 'righttop', ...
1080     'Tag', 'uipanelReconstruct', ...
1081     'Units', 'Pixels', ...
1082     'Position', [fig.panelsX ...
1083         (fig.panelPlotY-fig.panelSignalAlcada) ...
1084         fig.panelsAmplada ...
1085         fig.panelPlotAlcada ], ...
1086     'BorderType', 'beveledout', ...
1087     'Visible', 'off');

```

```

1088 % Elements del panelReconstruct
1089 % Segona fila
1090 nf=5; % Numero files que volem
1091 atGN=50; % Amplada dels radios
1092 apGN= 115; % Amplada del editText
1093 mee=14; % Marge entre elements
1094
1095 %posarem a la fila
1096 mitjana=fig.panelsAmplada-(atGN+apGN+mee);
1097
1098 fig.textLibraryPop=uicontrol(fig.uipanelReconstruct,...
1099     'Style','text',...
1100     'Tag','textLibraryPop',...
1101     'String','Library: ',...
1102     'FontWeight','bold',...
1103     'Enable','off',...
1104     'Position',[mitjana/2-atGN/4 ...
1105                 (fig.panelPlotAlcada/(nf+2)*(nf-0.5)+16/2) ...
1106                 atGN ...
1107                 16 ...
1108                 ]);
1109 fig.popupLibrary=uicontrol(fig.uipanelReconstruct,...
1110     'Style','popupmenu',...
1111     'String','Bior1.1',...
1112     'Tag','popupLibrary',...
1113     'Position',[(mitjana/2+atGN+mee) ...
1114                 (fig.panelPlotAlcada/(nf+2)*(nf-0.5)+22/4) ...
1115                 apGN ...
1116                 22 ...
1117                 ], ...
1118     'BackgroundColor','w',...
1119     'enable','off');
1120
1121 % Primera fila
1122 fig.radioButton=uicontrol(fig.uipanelReconstruct,...
1123     'Style','radiobutton',...
1124     'Tag','radioSelect',...
1125     'String','Select',...
1126     'Value',1,...
1127     'Enable','on',...
1128     'Position',[mitjana/2-atGN/4 ...
1129                 (fig.panelPlotAlcada/(nf+2)*(nf-2)+16/2) ...
1130                 atGN ...
1131                 16 ...
1132                 ],...
1133     'Enable','off',...
1134     'Callback',@RadioSelect);
1135 fig.editSelect=uicontrol(fig.uipanelReconstruct,...
1136     'Style','edit',...
1137     'Tag','editSelect',...
1138     'String','',...
1139     'Position',[(mitjana/2+atGN+mee) ...
1140                 (fig.panelPlotAlcada/(nf+2)*(nf-2)+16/2) ...
1141                 apGN ...
1142                 16 ...
1143                 ], ...
1144     'BackgroundColor','w',...

```

```

1142     'Enable','on');
1143
1144 % Tercera fila
1145 fig.margeP=12; % Marge dels panels interiors a l'exterior
1146 fig.midaAmpPanInt=fig.panelsAmplada-12*2; % amplada panels interiors
1147
1148 p=get(fig.editSelect,'Position');
1149
1150 str={'Write numbers and intervals separated by comma. Example: 1,3,5-
1151     12,14.'};
1152
1153 fig.textExempleSeleccio=icontrol('Parent',fig.uipanelReconstruct,...
1154     'Style','text',...
1155     'Tag','textExempleSeleccio',...
1156     'String',str,...
1157     'Units','Pixels',...
1158     'Position',[fig.margeP ((p(2)-20)/2) fig.midaAmpPanInt 44]);
1159
1160 [outstring pos] = textwrap(fig.textExempleSeleccio,str);
1161 set(fig.textExempleSeleccio,...
1162     'String',outstring,...
1163     'Position',[fig.margeP ...
1164                 (fig.panelPlotAlcada/(nf+2)*(nf-4)+16/2) ...
1165                 fig.midaAmpPanInt ...
1166                 pos(4)],...
1167     'Enable','off');
1168
1169 abV=fig.panelsAmplada*5/16; % Amplada dels botons View i Remove
1170 mitjana=fig.panelsAmplada-(2*abV+mee);
1171
1172 fig.buttonViewRec=icontrol(fig.uipanelReconstruct,...
1173     'Style','pushbutton',...
1174     'String','View',...
1175     'Tag','buttonViewRec',...
1176     'Enable','off',...
1177     'Position',[mitjana/2 ...
1178                 (fig.panelPlotAlcada/(nf+2)*(nf-5)+16/2) ...
1179                 abV ...
1180                 20 ...
1181                 ],...
1182     'Callback',@ButtonViewRec);
1183
1184 fig.buttonRemoveRec=icontrol(fig.uipanelReconstruct,...
1185     'Style','pushbutton',...
1186     'String','Remove',...
1187     'Tag','buttonRemoveRec',...
1188     'Enable','off',...
1189     'Position',[(mitjana/2+abV+mee) ...
1190                 (fig.panelPlotAlcada/(nf+2)*(nf-5)+16/2) ...
1191                 abV ...
1192                 20 ...
1193                 ],...
1194     'Callback',@ButtonRemoveRec);
1195

```

```

1196 clear nf abV atGN apGN mee mitjana p str pos outstring
1197
1198 %-----%
1199
1200 % ---Panel Reconstruct Custom Dictionaries
1201 fig.uipanelReconstructCustDict=uipanel('Parent',fig.figLDB,...
1202     'Title','Descriptor options for reconstruction',...
1203     'TitlePosition','righttop',...
1204     'Tag','uipanelReconstructCustDict',...
1205     'Units','Pixels',...
1206     'Position',[fig.panelsX ...
1207         ((finestra(4)-fig.panelPlotAlcada*2)- ...
1208             fig.panelSignalAlcada) ...
1209         fig.panelsAmplada ...
1210         fig.panelPlotAlcada*2 ...
1211         'BorderType','beveledout',...
1212         'Visible','off');
1213 % Elements del panelReconstruct
1214 % Segona fila
1215 nf=12; % Numero files que volem
1216 apGN=60; % Amplada dels popups i els texts
1217 mee=14; % Marge entre elements
1218
1219 fig.textJPop=icontrol(fig.uipanelReconstructCustDict,...
1220     'Style','text',...
1221     'Tag','textJPop',...
1222     'String','J: ',...
1223     'FontWeight','bold',...
1224     'TooltipString','Frequency resolution',...
1225     'Position',[ (fig.panelsAmplada/2-apGN/2-mee-apGN) ...
1226         (fig.panelPlotAlcada*2/(nf+2)*(nf-1)+16/2) ...
1227         apGN ...
1228         16 ...
1229         ]);
1229 fig.popupJ=icontrol(fig.uipanelReconstructCustDict,...
1230     'Style','popupmenu',...
1231     'String','1',...
1232     'Tag','popupJ',...
1233     'Position',[ (fig.panelsAmplada/2-apGN/2-mee-apGN) ...
1234         (fig.panelPlotAlcada*2/(nf+2)*(nf-2)+22/4) ...
1235         apGN ...
1236         22 ...
1237         ], ...
1238     'Backgroundcolor','w',...
1239     'TooltipString','Frequency resolution',...
1240     'Callback',@PopupJ);
1240
1241 fig.textKPop=icontrol(fig.uipanelReconstructCustDict,...
1242     'Style','text',...
1243     'Tag','textKPop',...
1244     'String','K: ',...
1245     'FontWeight','bold',...
1246     'TooltipString','Time or frequency position',...
1247     'Position',[ (fig.panelsAmplada/2-apGN/2) ...
1248         (fig.panelPlotAlcada*2/(nf+2)*(nf-1)+16/2) ...

```

```

1249         apGN           ...
1250         16             ]);
1251 fig.popupK=uicontrol(fig.uipanelReconstructCustDict,...
1252 'Style','popupmenu',...
1253 'String','1',...
1254 'Tag','popupK',...
1255 'TooltipString','Time or frequency position',...
1256 'Position', [(fig.panelsAmplada/2-apGN/2)           ...
1257              (fig.panelPlotAlcada*2/(nf+2)*(nf-2)+22/4) ...
1258              apGN           ...
1259              22             ], ...
1260 'BackgroundColor','w');
1261
1262 fig.textLPop=uicontrol(fig.uipanelReconstructCustDict,...
1263 'Style','text',...
1264 'Tag','textLPop',...
1265 'String','L: ',...
1266 'FontWeight','bold',...
1267 'TooltipString','Time resolution',...
1268 'Position', [(fig.panelsAmplada/2+apGN/2+mee)           ...
1269              (fig.panelPlotAlcada*2/(nf+2)*(nf-1)+16/2) ...
1270              apGN           ...
1271              16             ]);
1272 fig.popupL=uicontrol(fig.uipanelReconstructCustDict,...
1273 'Style','popupmenu',...
1274 'String','1',...
1275 'Tag','popupL',...
1276 'TooltipString','Time resolution',...
1277 'Position', [(fig.panelsAmplada/2+apGN/2+mee)           ...
1278              (fig.panelPlotAlcada*2/(nf+2)*(nf-2)+22/4) ...
1279              apGN           ...
1280              22             ], ...
1281 'BackgroundColor','w');
1282
1283 fig.buttonAddToReconstr=uicontrol(fig.uipanelReconstructCustDict,...
1284 'Style','pushbutton',...
1285 'Tag','buttonAddToReconstr',...
1286 'String','Add',...
1287 'Position', [mee           ...
1288              (fig.panelPlotAlcada*2/(nf+2)*(nf-3.5)+16/2) ...
1289              fig.panelsAmplada/2-mee           ...
1290              22             ], ...
1291 'Enable','off',...
1292 'Callback',@ButtonAddToReconstr);
1293 fig.buttonAddNodeToReconstr=uicontrol(fig.uipanelReconstructCustDict,...
1294 'Style','pushbutton',...
1295 'Tag','buttonAddNodeToReconstr',...
1296 'String','Add node',...
1297 'Position', [fig.panelsAmplada/2+mee           ...
1298              (fig.panelPlotAlcada*2/(nf+2)*(nf-3.5)+16/2) ...
1299              fig.panelsAmplada/2-mee*2           ...
1300              22             ], ...

```

```

1301     'Enable','off',...
1302     'Callback',@ButtonAddNodeToReconstr);
1303
1304     % Tercera fila
1305     fig.listBoxJKL=icontrol(fig.uipanelReconstructCustDict,...
1306     'Style','listbox',...
1307     'Tag','listboxJKL',...
1308     'Position',[mee ...
1309     mee ...
1310     fig.panelsAmplada/2-mee ...
1311     (fig.panelPlotAlcada*2/(nf+2)*(nf-5)+16/2)],...
1312     'BackgroundColor','w',...
1313     'Enable','off');
1314
1315     pos=[fig.panelsAmplada/2+mee ...
1316     (fig.panelPlotAlcada*2/(nf+2)*(nf-5)+16/2)/2-mee ...
1317     fig.panelsAmplada/2-mee*2 ...
1318     ((fig.panelPlotAlcada*2/(nf+2)*(nf-5)+16/2))/2-mee];
1319     fig.buttonReconsCustDict=icontrol(fig.uipanelReconstructCustDict,...
1320     'Style','pushbutton',...
1321     'Tag','buttonReconsCustDict',...
1322     'String','Reconstruct',...
1323     'Position',pos,...
1324     'Enable','off',...
1325     'Callback',@ButtonReconsCustDict);
1326     fig.botoRemoveToReconstruct=icontrol(fig.uipanelReconstructCustDict,...
1327     'Style','pushbutton',...
1328     'Tag','botoRemoveToReconstruct',...
1329     'String','Remove',...
1330     'Position',[pos(1) ...
1331     (fig.panelPlotAlcada*2/(nf+2)*(nf-5)+16/2)- ...
1332     pos(4)/3+mee ...
1333     pos(3) ...
1334     pos(4)/3],...
1335     'Enable','off',...
1336     'Callback',@ButtonRemoveToReconstr);
1337     fig.botoRemoveAllToReconstruct=icontrol(...
1338     fig.uipanelReconstructCustDict,...
1339     'Style','pushbutton',...
1340     'Tag','botoRemoveToReconstruct',...
1341     'String','Remove all',...
1342     'Position',[pos(1) ...
1343     mee ...
1344     pos(3) ...
1345     pos(4)/3],...
1346     'Enable','off',...
1347     'Callback',@ButtonRemoveAllToReconstr);
1348
1349     clear nf abV atGN apGN mee mitjana p str pos outstring finestra
1350
1351     % Ja podem carregar la llibreria Complete_dLibrary
1352
1353     load Complete_dLibrary;

```



```

1354 % Mirem de no carregar 'LSin' ni 'LCos'
1355 fig.Complete_dLibrary=Complete_dLibrary;
1356 fig.Complete_dLibrary(...
1357     (ismember(fig.Complete_dLibrary, 'LCos')==1))=[];
1358 fig.Complete_dLibrary(...
1359     (ismember(fig.Complete_dLibrary, 'LSin')==1))=[];
1360
1361 fig.Complete_dLibrary=sort(fig.Complete_dLibrary); % Ordre alfabètic
1362
1363 % Inicialitzem els valors pels listboxes
1364 % Inicialment fig.esquerra tindrà tots els valors de Complete_dLibrary
1365 fig.esquerra=fig.Complete_dLibrary;
1366 % i fig.dreta estarà buit
1367 fig.dreta={};
1368 % Última posició del listbox esquerre (màxims elements):
1369 fig.ultimaPosicioEsquerra=length(fig.esquerra);
1370 % Última posició del listbox dret (màxims elements):
1371 fig.ultimaPosicioDreta=0;
1372
1373 % Inicialitzem la variable dels subespais
1374 if isfield(fileGlobal,'subspaceReconstr')
1375     fileGlobal = rmfield(fileGlobal, 'subspaceReconstr');
1376 end
1377 clear Complete_dLibrary LSin LCos i j
1378
1379 % Desem en una variable el valor 'Std' (valor per defecte) per indicar
1380 % si fem Analysis Discriminant 'Standard' o 'Custom'
1381 fig.StdOrCust='Std';
1382 guidata(gcf,fig); % emmagatzema l'estructura
1383
1384 % Per tal que no es pugui modificar des de la línia de comandes:
1385 set(fig.figLDB,'HandleVisibility','callback');
1386
1387 end
1388
1389 % -----
1390 %   Callbacks pel GUI
1391 % -----
1392 function LDB_CloseRequestFcn (varargin)
1393 % global fileGlobal fileNameGlob filePathNameGlob
1394 fig=guidata(gcbo);
1395 % surt=questdlg('Are you sure you want to exit?','Exit',...
1396 % 'Yes','No','No');
1397 % if strcmp(surt,'No')
1398 %     return;
1399 % end
1400 clear global % esborra totes les variables globals
1401
1402 delete(fig.figLDB); % Tanca la figura
1403
1404 end
1405
1406
1407 function MenuExit (varargin)
1408
1409 LDB_CloseRequestFcn();
1410
1411 end

```

```

1412
1413
1414 function PopupGroupNumber (varargin)
1415 fig=guidata(gcbo); % llegeix guidata.
1416 global fileGlobal
1417
1418 % Obté el número seleccionat al 'popupGroupNumber':
1419 i=get(fig.popupGroupNumber,'Value');
1420 % Busca els índexs que corresponen al número sel·leccionat
1421 ipos = find(fileGlobal.ranswer==i);
1422
1423 % Posa el primer element de ipos a editFirst:
1424 set(fig.editFirst,'String',num2str(ipos(1)));
1425 % Posa l'últim element de ipos a editLast:
1426 set(fig.editLast,'String',num2str(ipos(end)));
1427
1428 % -----%
1429 % Netegem les variables que ja no necessitarem
1430 clear i ipos;
1431
1432 guidata(gcbo,fig); % Desem els canvis fets
1433 end
1434
1435
1436 function EditFirst (varargin)
1437 global fileGlobal
1438 fig=guidata(gcbo); % llegeix guidata.
1439 % Obté el número seleccionat al 'popupGroupNumber':
1440 i=get(fig.popupGroupNumber,'Value');
1441 % Busca els índexs que corresponen al número sel·leccionat
1442 ipos = find(fileGlobal.ranswer==i);
1443 %Obtenim el valor de 'fig.editFirst'
1444 x=(str2double(get(fig.editFirst,'String')));
1445
1446 %Comprovació d'errors d'entrada
1447 if mod(x,1)~=0, % si no es enter
1448     x=round(x); % l'arrodonirem a l'enter més pròxim
1449     set(fig.editFirst,'String',x);
1450 end
1451 if (isnan(x)) % Si no és un valor numèric
1452     % Posarem el primer valor vàlid trobat
1453     set(fig.editLast,'Enable','off');
1454     set(fig.textWait,...
1455         'String','You must enter a numeric value',...
1456         'Enable','on',...
1457         'Visible','on');
1458     pause(1)
1459
1460     for i=1:3 % Farà pampallugues on hi ha l'error
1461         set(fig.editFirst,...
1462             'enable','off');
1463         pause(0.1)
1464         set(fig.editFirst,...
1465             'enable','on');
1466         pause(0.1)
1467     end
1468

```

```

1469     set(fig.editFirst, 'String', ipos(1));
1470
1471     pause(1)
1472     set(fig.editLast, 'Enable', 'on');
1473     set(fig.textWait, ...
1474         'String', 'Please wait...', ...
1475         'Enable', 'off', ...
1476         'Visible', 'off');
1477 end
1478
1479 if (str2double(get(fig.editFirst, 'String')) < ipos(1)
1480     % Si entrem una dada inferior al vector Group Number
1481
1482     % Desactivem 'editLast' perquè no puguem entrar-hi dades
1483     set(fig.editLast, 'Enable', 'off');
1484     str=['Minimal value: ' num2str(ipos(1))]; % Concatenem strings
1485     set(fig.textWait, ... % Al text de progrés
1486         'String', str, ... % hi posem l'string concatenat
1487         'Enable', 'on', ... % l'activem
1488         'Visible', 'on'); % i la fem visible
1489     pause(1)
1490
1491     for i=1:3 % Farà pampallugues on hi ha l'error
1492         set(fig.editFirst, ...
1493             'enable', 'off');
1494         pause(0.1)
1495         set(fig.editFirst, ...
1496             'enable', 'on');
1497         pause(0.1)
1498     end
1499
1500     %Canviem l'string de 'editFirst'
1501     set(fig.editFirst, 'String', ipos(1));
1502
1503     pause(1)
1504     set(fig.editLast, 'Enable', 'on'); %Ja podem activar 'editLast'
1505     set(fig.textWait, ...
1506         'String', 'Please wait...', ... % Li posem el seu string inicial
1507         'Enable', 'off', ... % el desactivem
1508         'Visible', 'off'); % i el fem invisible
1509 end
1510 if (str2double(get(fig.editFirst, 'String')) > ...
1511     (str2double(get(fig.editLast, 'String'))))
1512     % si entrem una dada superior a editLast
1513     set(fig.editLast, 'Enable', 'off');
1514     % Farem un cell-array enlloc de concatenar perquè és més ràpid que
1515     % strvcat (concatenació vertical)
1516     strCell{1,1}=...
1517         ("First" needs to be less than or equal to "Last": ');
1518     strCell{2,1}=['Last=' (num2str(get(fig.editLast, 'String')))];
1519
1520     set(fig.textWait, ... % Al text de progrés
1521         'FontSize', 19, ... % li canviem la mida de la lletra
1522         'String', strCell, ... % hi posem l'string concatenat
1523         'Enable', 'on', ... % l'activem
1524         'Visible', 'on'); % i la fem visible

```

```

1525     pause(1.5)
1526
1527     for i=1:3 % Farà pampallugues on hi ha l'error
1528         set(fig.editFirst,...
1529             'enable','off');
1530         pause(0.1)
1531         set(fig.editFirst,...
1532             'enable','on');
1533         pause(0.1)
1534     end
1535
1536     % Canviem l'string de editFirst
1537     set(fig.editFirst,'String',(num2str(get(fig.editLast,'String'))));
1538
1539     pause(1.5)
1540     set(fig.editLast,'Enable','on'); %Ja podem activar 'editLast'
1541     set(fig.textWait,...
1542         'String','Please wait...',...% Li posem el seu string inicial
1543         'FontSize',35,...           % li posem mida de lletra inicial
1544         'Enable','off',...         % el desactivem
1545         'Visible','off');          % i el fem invisible
1546 end
1547 if (str2double(get(fig.editFirst,'String'))>ipos(end)
1548     % si entrem una dada superior al vector Group Number
1549     set(fig.editLast,'Enable','off');
1550     str=['Maximum value: ' num2str(ipos(end))];
1551     set(fig.textWait,...
1552         'String',str,...
1553         'Enable','on',...
1554         'Visible','on');
1555     pause(1)
1556
1557     for i=1:3 % Farà pampallugues on hi ha l'error
1558         set(fig.editFirst,...
1559             'enable','off');
1560         pause(0.1)
1561         set(fig.editFirst,...
1562             'enable','on');
1563         pause(0.1)
1564     end
1565
1566     set(fig.editFirst,'String',ipos(end));
1567
1568     pause(1)
1569     set(fig.editLast,'Enable','on');
1570     set(fig.textWait,...
1571         'String','Please wait...',...
1572         'Enable','off',...
1573         'Visible','off');
1574 end
1575
1576     %Ja podem obtenir la dada de 'fig.editFirst' i desar-la a 'fig.first'
1577     fig.first=str2double(get(fig.editFirst,'String'));
1578
1579     % Netegem les variables que ja no necessitarem
1580     clear x str strCell i ipos

```

```

1581
1582 guidata(gcbo,fig);           % Desem els canvis fets
1583 end
1584
1585
1586 function EditLast (varargin)
1587 fig=guidata(gcbo); % llegeix guidata.
1588 global fileGlobal
1589 i=get(fig.popupGroupNumber,'Value');
1590 ipos = find(fileGlobal.ranswer==i);
1591
1592 x=(str2double(get(fig.editLast,'String')));
1593
1594 if (isnan(x))
1595     set(fig.editFirst,'Enable','off');
1596     set(fig.textWait,...
1597         'String','You must enter a numeric value',...
1598         'Enable','on',...
1599         'Visible','on');
1600     pause(1)
1601
1602     for i=1:3 % Farà pampallugues on hi ha l'error
1603         set(fig.editLast,...
1604             'enable','off');
1605         pause(0.1)
1606         set(fig.editLast,...
1607             'enable','on');
1608         pause(0.1)
1609     end
1610
1611     set(fig.editLast,'String',ipos(end));
1612
1613     pause(1)
1614     set(fig.editFirst,'Enable','on');
1615     set(fig.textWait,...
1616         'String','Please wait... ',...
1617         'Enable','off',...
1618         'Visible','off');
1619 end
1620
1621 x=(str2double(get(fig.editLast,'String')));
1622
1623 if mod(x,1)~=0, % si no és enter
1624     x=round(x); % l'arrodonirem a l'enter més pròxim
1625     set(fig.editLast,'String',x);
1626 end
1627
1628 % si entrem una dada inferior a editFirst:
1629 if (str2double(get(fig.editLast,'String'))...
1630     <(str2double(get(fig.editFirst,'String'))))
1631     set(fig.editFirst,'Enable','off');
1632     strCell{1,1}=(...
1633         '"Last" needs to be greater than or equal to "First".');
1634     strCell{2,1}= ['First=' (num2str(get(fig.editFirst,'String')))];
1635     set(fig.textWait,...
1636         'FontSize',19,....

```

```

1637         'String',strCell,...
1638         'Enable','on',...
1639         'Visible','on');
1640     pause(1.5)
1641
1642     for i=1:3 % Farà pampallugues on hi ha l'error
1643         set(fig.editLast,...
1644             'enable','off');
1645         pause(0.1)
1646         set(fig.editLast,...
1647             'enable','on');
1648         pause(0.1)
1649     end
1650
1651     set(fig.editLast,'String',(num2str(get(fig.editFirst,'String'))));
1652
1653     pause(1.5)
1654     set(fig.editFirst,'Enable','on');
1655     set(fig.textWait,...
1656         'String','Please wait...',...
1657         'FontSize',35,...
1658         'Enable','off',...
1659         'Visible','off');
1660 end
1661
1662 % si entrem una dada inferior al vector Group Number
1663 if (str2double(get(fig.editLast,'String'))<ipos(1)
1664     set(fig.editFirst,'Enable','off');
1665     str=['Minimal value: ' num2str(ipos(1))];
1666     set(fig.textWait,...
1667         'String',str,...
1668         'Enable','on',...
1669         'Visible','on');
1670     pause(1)
1671
1672     for i=1:3 % Farà pampallugues on hi ha l'error
1673         set(fig.editLast,...
1674             'enable','off');
1675         pause(0.1)
1676         set(fig.editLast,...
1677             'enable','on');
1678         pause(0.1)
1679     end
1680
1681     set(fig.editLast,'String',ipos(1));
1682
1683     pause(1)
1684     set(fig.editFirst,'Enable','on');
1685     set(fig.textWait,...
1686         'String','Please wait...',...
1687         'Enable','off',...
1688         'Visible','off');
1689 end
1690 % si entrem una dada superior al vector Group Number
1691 if (str2double(get(fig.editLast,'String'))>ipos(end)
1692     set(fig.editFirst,'Enable','off');

```

```

1693     str=['Maximum value: ' num2str(ipos(end))];
1694     set(fig.textWait,...
1695         'String',str,...
1696         'Enable','on',...
1697         'Visible','on');
1698     pause(1)
1699
1700     for i=1:3 % Farà pampallugues on hi ha l'error
1701         set(fig.editLast,...
1702             'enable','off');
1703         pause(0.1)
1704         set(fig.editLast,...
1705             'enable','on');
1706         pause(0.1)
1707     end
1708
1709     set(fig.editLast,'String',ipos(end));
1710
1711     pause(1)
1712     set(fig.editFirst,'Enable','on');
1713     set(fig.textWait,...
1714         'String','Please wait...',...
1715         'Enable','off',...
1716         'Visible','off');
1717 end
1718 fig.last=str2double(get(fig.editLast,'String'));
1719
1720 % Netegem les variables que ja no necessitarem
1721 clear x str strCell i ipos
1722
1723 guidata(gcbo,fig); % Desem els canvis fets
1724 end
1725
1726
1727 function RadioStd (varargin)
1728 fig=guidata(gcbo); % llegeix guidata.
1729
1730 % Indiquem quin radioButton està actiu
1731 set(fig.radioStd,'Value',1)
1732 set(fig.radioCust,'Value',0)
1733
1734 fig.StdOrCust='Std'; % Desem en una variable el valor 'Std'
1735
1736 % Pels Listboxes del Filter Menu
1737
1738 fig.esquerra=fig.Complete_dLibrary; % Ho carreguem a 'esquerra'
1739 fig.dreta={}; % Buidem 'dreta'
1740
1741 % Última posició del listBox esquerre (màxims elements):
1742 fig.ultimaposicioEsquerra=length(fig.esquerra);
1743 % Última posició del listBox dret (màxims elements):
1744 fig.ultimaposicioDreta=0;
1745
1746 % Obtenim les mides i posició de uipanelCustNK i de uipanelListboxes
1747 p=get(fig.uipanelCustNK,'Position');
1748 % u=get(fig.uipanelListboxes,'Position');
1749 u=fig.panelDiscrimAnalysisMenuAlçada/2-10; % Alçada de uipanelListboxes
1750 % Alçada del panel uipanelListboxes:

```

```

1751 v=u+p(4)+10; % alçada uipanelCustNK + de uipanelListboxes + marge
1752
1753 set(fig.uipanelNivellDescomp,'Visible','on');
1754
1755 set(fig.uipanelCustNK,'Visible','off')
1756
1757 % Posem el títol al panel de les llibreries,
1758 % ja que queda esborrat quan premem el radio de Custom
1759 set(fig.uipanelListboxes,...
1760     'Visible','on',...
1761     'Title','Libraries',...
1762     'TitlePosition','centertop ',...
1763     'Position',[fig.margeP          ...
1764                 (p(2)-fig.panelDiscrimAnalysisMenuAlcada/2) ...
1765                 fig.midaAmpPanInt    ...
1766                 v                      ] )
1767
1768 % Obtenim les mides i posició de uipanelListboxes:
1769 p=get(fig.uipanelListboxes,'Position');
1770
1771 mee=3; % Marge entre elements interns: ListBoxes-mee-botons
1772 mde=6; % Marge dret, esquerre, dalt i baix entre listboxes i panel.
1773 mbp=16; % Amplada del botó petit, botoAfegeix i botoTreu
1774 % Fent una petita equació per trobar l'amplada dels Listboxes:
1775 % midaPanelAmplada=mde+midaListBoxes+mee+mbp+mee+midaListBoxes+mde
1776 % midaPanelAmplada=2*mde+2*midaListBoxes+2*mee+mbp ...
1777 % midaListBoxes=(midaPanelAmplada-mbp-2*(mde+mee))/2
1778 ampLB=(p(3)-mbp-2*(mde+mee))/2; % Amplada de ListBoxes
1779
1780 abT=14; % Alçada dels botons botoTots i botoCap
1781 alcLB=(p(4)-2*mde-abT-mee*4); % Alçada dels Listboxs
1782
1783 set(fig.listBoxEntrada,...
1784     'Position',[mde          ...
1785                 (mde+abT+mee) ...
1786                 ampLB        ...
1787                 alcLB        ],...
1788     'String',fig.Complete_dLibrary,... % Mostrarem tot l'string 'c'
1789     'Visible','on')
1790 set(fig.listBoxSortida,...
1791     'Position',[ (mde+ampLB+2*mee+mbp) ...
1792                 (mde+abT+mee)         ...
1793                 ampLB                  ...
1794                 alcLB                  ],...
1795     'String',fig.dreta,...
1796     'Visible','on')
1797 set(fig.botoTots,...
1798     'Position',[mde          ...
1799                 mde          ...
1800                 ampLB        ...
1801                 abT          ],...
1802     'Visible','on')
1803 set(fig.botoCap,...
1804     'Position',[ (mde+ampLB+2*mee+mbp) ...
1805                 mde                    ...

```



```

1806         ampLB           ...
1807         abT             ],...
1808     'Visible','on');
1809 set(fig.botoAfegeix,...
1810     'Position',[(mde+ampLB+mee) ...
1811                 ((p(4)+mee+abT)/2+mee/2) ...
1812                 mbp ...
1813                 alcLB/3 ...
1814     'Visible','on');
1815 set(fig.botoTreu,...
1816     'Position',[(mde+ampLB+mee) ...
1817                 ((p(4)+mee+abT)/2-alcLB/3-mee/2) ...
1818                 mbp ...
1819                 alcLB/3 ...
1820     'Visible','on');
1821
1822 set(fig.buttonProcess,'Visible','on','enable','off')
1823
1824 set([fig.botoAfegeix fig.botoTots fig.listboxEntrada],'Enable','on');
1825 set([fig.botoTreu     fig.botoCap fig.listboxSortida],'Enable','off');
1826
1827 % Netegem les variables que ja no necessitarem
1828 clear str c abT alcLB ampLB mbp mde mee p u v
1829
1830 guidata(gcbo,fig)           % Desem els canvis fets
1831 end
1832
1833
1834 function RadioCust (varargin)
1835 fig=guidata(gcbo); % llegeix guidata.
1836
1837 set(fig.radioStd,'Value',0)
1838 set(fig.radioCust,'Value',1)
1839
1840 fig.StdOrCust='Cust';
1841
1842 set(fig.uipanelListboxes,...
1843     'Title',''); % Treiem el títol del panel de les llibreries,
1844     % ja que queda posat quan premem el radio de Standard
1845
1846 % Pels Listboxes del Filter Menu
1847
1848 set(fig.uipanelCustNK,...
1849     'Visible','on')
1850
1851 % Obtenim les mides i posició de uipanelCustNK
1852 p=get(fig.uipanelCustNK,'Position');
1853 set(fig.uipanelListboxes,...
1854     'Visible','on',...
1855     'Title','',...
1856     'Position',[fig.margeP ...
1857                 (p(2)-fig.panelDiscrimAnalysisMenuAlcada/2) ...
1858                 fig.midaAmpPanInt ...
1859                 fig.panelDiscrimAnalysisMenuAlcada/2-10 ...
1860     set(fig.buttonProcess,'Visible','on')
1861     set([fig.entraN fig.entraK],'Enable','on');

```

```

1862
1863 fig.esquerra={};
1864 fig.dreta={};
1865
1866 % Obtenim les mides i posició de uipanelListboxes
1867 p=get(fig.uipanelListboxes,'Position');
1868
1869 mde=6; % Marge dret, esquerre, dalt i baix entre listboxes i panel.
1870
1871 alcLB=(p(4)-2*mde); % Alçada dels Listboxes
1872 ampLBCust=p(3)-2*mde;
1873
1874 set(fig.listBoxEntrada,...
1875     'Position',[mde     ...
1876                 mde     ...
1877                 ampLBCust ...
1878                 alcLB   ],...
1879     'String',fig.esquerra)
1880
1881 set([fig.listBoxSortida ... % No mostrem
1882     fig.botoTots         fig.botoCap ...
1883     fig.botoAfegeix     fig.botoTreu ],'Visible','off');
1884
1885 set(fig.buttonProcess,'enable','off')
1886
1887 set([fig.botoAfegeix fig.botoTots fig.botoCap],...
1888     'Enable','off');%Aquests els desactivem
1889 % Ja podem activar el botó buttonCreateComb
1890 set(fig.buttonCreateComb,'enable','on');
1891
1892
1893 clear p alcLB ampLBCust mde
1894 set(fig.listBoxEntrada,'enable','off')
1895 % Fem visible el panel uipanelNivellDescomp
1896 set(fig.uipanelNivellDescomp,'visible','on');
1897
1898 guidata(gcbo,fig)           % Desem els canvis fets
1899 end
1900
1901
1902 function ButtonView (varargin)
1903 % Botó View del panel 'Signal Representation'
1904 fig=guidata(gcbo); % llegeix guidata.
1905 global fileGlobal fileNameGlob ContourOrProfile
1906 if isfield(fileGlobal,'signals')
1907     EnableDisable(gcbo,'off',0)
1908     % posem una pausa perquè es vegi millor l'acció i no falli
1909     pause(0.01)
1910
1911     set([fig.buttonRemove ...
1912         fig.checkPan     fig.checkCursor...
1913         fig.checkZoom    fig.checkGrid], 'Enable','on');
1914
1915     AxesResetInvisible();
1916
1917     set(fig.panelTaula,'Visible','off')
1918     zoom off

```

```

1919
1920 EditFirst(); % Cridem ques funcions que corregiran l'entrada
1921 EditLast(); % de dades
1922
1923 fig.first=str2double(get(fig.editFirst,'String'));
1924 fig.last=str2double(get(fig.editLast,'String'));
1925
1926 if strcmp(ContourOrProfile,'Profile')
1927     axes(fig.axesPlot);
1928     set(fig.axesPlot,'Visible','on');
1929
1930     % Array de caràcters amb els colors de línia:
1931     % El blanc '-w' no li posarem perquè no es veuria amb el fons
1932     % No hi posem el vermell perquè l'usarem per altres coses
1933     linia=['-b';'-g';'-c';'-m';'-y';'-k'];
1934     % índex per llegir aquest array:
1935     j=get(fig.popupGroupNumber,'Value');
1936
1937     seny=(size(fileGlobal.xsignals,1));
1938
1939     y=0:seny/(seny-1):seny;
1940     plot(y,fileGlobal.xsignals(:,fig.first:fig.last),linia(j,:));
1941
1942     hold on
1943
1944     xlabel('Samples','Color','w');
1945     % Perquè es vegi bé el caràcter _, el substituïm per \_
1946     str = strrep(fileNameGlob, '_', '\_');
1947
1948     title(num2str(str),'Color','w');
1949
1950     set(gca,'XTick',0:seny/8:seny)
1951     grid on
1952     axis tight
1953     zoom reset;
1954
1955     clear y
1956 else
1957     axes(fig.axesPlot);
1958     set(fig.axesPlot,'Visible','on');
1959     % Array de caràcters amb els colors de línia:
1960     % El blanc '-w' no li posarem perquè no es veuria amb el fons
1961     % No hi posem el vermell perquè l'usarem per altres coses
1962     linia=['-b';'-g';'-c';'-m';'-y';'-k'];
1963     % índex per llegir aquest array:
1964     j=get(fig.popupGroupNumber,'Value');
1965
1966     angle=0:2*pi/(size(fileGlobal.xsignals,1)-1):2*pi;
1967     % angle=-angle; % Si no canviéssim la direcció dels eixos
1968     % hauríem de canviar la direcció de l'angle.
1969
1970     polar(angle',fileGlobal.xsignals(:,fig.first),linia(j,:));
1971
1972     hold on
1973
1974     if fig.first~=fig.last
1975         for i=fig.first+1:fig.last
1976             polar(angle',fileGlobal.xsignals(:,i),linia(j,:));
1977         end

```

```

1978     end
1979     % Perquè es vegi bé el caràcter _, el substituïm per \_
1980     str = strrep(fileNameGlob, '_', '\_');
1981
1982     title(num2str(str), 'Color', 'w');
1983
1984     set(gca, 'YDir', 'reverse') % canviem la direcció dels eixos
1985 end
1986 if(get(fig.checkGrid, 'Value'))
1987     grid on;
1988 else
1989     grid off;
1990 end
1991 if(get(fig.checkPan, 'Value'))
1992     pan on;
1993 else
1994     pan off;
1995 end
1996 if(get(fig.checkZoom, 'Value'))
1997     zoom on;
1998 else
1999     zoom off;
2000 end
2001 if(get(fig.checkCursor, 'Value'))
2002     datacursormode on;
2003 else
2004     datacursormode off;
2005 end
2006 EnableDisable(gcbo, 'on', 0)
2007 end
2008
2009 clear str
2010
2011 guidata(gcbo, fig)           % Desem els canvis fets
2012 end
2013
2014
2015 function ButtonRemove (varargin)
2016 % Botó Remove del panel 'Signal Representation'
2017 fig=guidata(gcbo); % llegeix guidata.
2018 global fileGlobal fileNameGlob ContourOrProfile
2019
2020 EnableDisable(gcbo, 'off', 0)
2021 % posem una pausa perquè es vegi millor l'acció i no falli
2022 pause(0.01)
2023 set(fig.textWait, ...
2024     'String', 'Removing and drawing...', ...
2025     'Enable', 'on', ...
2026     'Visible', 'on');
2027 drawnow
2028
2029 AxesResetInvisible();
2030
2031 set(fig.axesFix, 'Visible', 'on');
2032
2033 zoom reset;

```

```

2034 datacursormode off
2035 zoom off
2036 pan off
2037
2038 set([fig.checkPan fig.checkCursor fig.checkZoom], ...
2039     'Value',0,'Enable','off');
2040 set(fig.checkGrid,'Enable','off');
2041
2042 % Farà que es vegi correctament el caràcter _:
2043 str = strrep(fileNameGlob, '_', '\_');
2044 str=['(' str ')']; % Hi afegim 'Totes les parts'
2045
2046 if strcmp(ContourOrProfile,'Profile')
2047     drawnow
2048     str=['Radial view ' str];
2049     FuncPlot(fig.axesFix, ...
2050             str, ...
2051             fileGlobal.xsignals,fileGlobal.ranswer, ...
2052             'Samples')
2053 else
2054     str=['Contour view ' str];
2055     FuncPolar(fig.axesFix, ...
2056             str, ...
2057             fileGlobal.xsignals,fileGlobal.ranswer, ...
2058             'Samples')
2059 end
2060
2061 EnableDisable(gcbo,'on',0)
2062
2063 set(fig.buttonRemove,'Enable','off')
2064
2065 set(fig.textWait, ...
2066     'String','Please wait...', ...
2067     'Visible','off');
2068
2069 clear linia fi i j jend str
2070
2071 guidata(gcbo,fig) % Desem els canvis fets
2072 end
2073
2074
2075 function EntraK (varargin)
2076 % Caixa per entrar els paràmetres del filtre -1 (K)
2077 fig=guidata(gcbo); % llegeix guidata. Així les dades anteriors no es perdran
2078
2079 x=str2double(get(fig.entraK,'String'));
2080 if (isnan(x))
2081     set(fig.entraN,'Enable','off');
2082     set(fig.textWait, ...
2083         'String','You must enter a numeric value', ...
2084         'Enable','on', ...
2085         'Visible','on');
2086     pause(1.5)
2087
2088 for i=1:3 % Farà pampallugues on hi ha l'error
2089     set(fig.entraK, ...
2090         'enable','off');

```

```

2091         pause(0.1)
2092         set(fig.entraK,...
2093             'enable','on');
2094         pause(0.1)
2095     end
2096
2097     set(fig.entraK,'String',1);
2098     pause(1.5)
2099     set(fig.entraN,'Enable','on');
2100     set(fig.textWait,...
2101         'String','Please wait...',...
2102         'Enable','off',...
2103         'Visible','off');
2104 end
2105
2106 x=str2double(get(fig.entraK,'String'));
2107 if mod(x,1)~=0,% si no es enter
2108     x=round(x);
2109     set(fig.entraK,'String',x); % l'arrodonirem a l'enter més pròxim
2110 end
2111
2112 x=str2double(get(fig.entraK,'String'));
2113 if x<1 % Si hi posem 0, o hi queda per arrodoniment
2114     set(fig.entraK,'String',1); % que entri 1
2115 end
2116
2117 % Netegem les variables que ja no necessitarem
2118 clear x
2119 end
2120
2121 function EntraN (varargin)
2122 % Caixa per entrar el mostreig de l'interval [0,pi) del filtre
2123 fig=guidata(gcbo); % llegeix guidata. Així les dades anteriors no es perdran
2124
2125 x=str2double(get(fig.entraN,'String'));
2126 if (isnan(x))
2127     set(fig.entraK,'Enable','off');
2128     set(fig.textWait,...
2129         'String','You must enter a numeric value',...
2130         'Enable','on',...
2131         'Visible','on');
2132     pause(1.5)
2133
2134     for i=1:3 % Farà pampallugues on hi ha l'error
2135         set(fig.entraN,...
2136             'enable','off');
2137         pause(0.1)
2138         set(fig.entraN,...
2139             'enable','on');
2140         pause(0.1)
2141     end
2142
2143     set(fig.entraN,'String',1);
2144
2145     pause(1.5)
2146     set(fig.entraK,'Enable','on');
2147     set(fig.textWait,'String','Please wait...',...
2148         'Enable','off',...
2149         'Visible','off');
2150 end
2151

```

```

2152 x=str2double(get(fig.entraN,'String'));
2153 if mod(x,1)~=0,% si no es enter
2154     x=round(x);
2155     set(fig.entraN,'String',x); % l'arrodoniríem a l'enter més pròxim
2156 end
2157
2158 x=str2double(get(fig.entraN,'String'));
2159 if x<1 % si hi posem 0 o hi queda després d'arrodonir
2160     set(fig.entraN,'String',1); % que entri 1
2161 end
2162
2163 % Netegem les variables que ja no necessitarem
2164 clear x
2165 end
2166
2167
2168 function ButtonCreateComb (varargin)
2169 % Genera totes les combinacions possibles d'un conjunt de 1 a N.
2170 % En total hi haurà N^K combinacions, on K és número d'elements.
2171
2172 fig=guidata(gcbo); % llegeix guidata.
2173
2174 % Agafem les dades del panel entraN i entraK
2175 fig.N=str2double(get(fig.entraN,'String')); %Per saber si és NaN
2176 fig.K=str2double(get(fig.entraK,'String'));
2177
2178 N=(1:str2double(get(fig.entraN,'String'))); % Vector de 1 a N
2179
2180
2181 if (~isnan(fig.N)) && (~isnan(fig.K))
2182     set(fig.listboxEntrada,'Value',1)
2183
2184     % Modificació de la interfície
2185     EnableDisable(gcbo,'off',0) % Enviem l'objecte que ha fet l'acció,
2186     % diem que posi els elements a off
2187     % i diem que no és l'inici d'un fitxer
2188
2189     % Barra progrés
2190     set(fig.textWait,...
2191         'String','Creating combinations... ',...
2192         'Visible','on',...
2193         'Enable','on')
2194     drawnow
2195
2196     % Combinacions
2197     ii = fig.K:-1:1; % invertirà l'ordre dels elements de la matriu
2198     ord=cell(1,fig.K);
2199     for i=1:fig.K % ord={[1 2 3],[1 2 3],[1 2 3]} per fer el ndgrid
2200         ord{1,i}=N;
2201     end
2202     if fig.K==1,% Si només entrem K==1
2203         A = ord{1}(:) ;
2204     else % Si K>1
2205         [A{1:fig.K}] = ndgrid(ord{ii}); % Com meshgrid. Cell amb arrays
2206         % de les files copiades a A(1,1), les columnes a A(1,2)... en
2207         % forma de matriu i la dimensió n són còpies dels elements del
2208         % vector
2209         A = reshape(cat(fig.K+1,A{:}), [], fig.K); % Concatena
2210         A = A(:,ii) ;
2211     end

```

```

2212     % Anomenem els vectors:
2213     x=1;
2214     B=cell(size(A,1),1); % Reservem memòria
2215     for i=1:size(A,1)
2216         % Concatenació d'strings:Totes les files posades en un cell
2217         % array, i amb els 2 espais substituïts per ', ', tot precedit
2218         % per 'Cust_V):
2219         B{x,1}=strcat('m-wave o',num2str(2*size(A,2)+1),'_i',num2str(i));
2220         x=x+1;
2221     end
2222
2223     %Centrem ara la matriu A en l'interval [0,pi)
2224     for i=1:size(A,1)
2225         for j=1:size(A,2)
2226             A(i,j)=(A(i,j)-1)/pi; % (n-1)/pi
2227         end
2228     end
2229
2230     fig.cust_Matr_V=A; % Desem en una variable fixe
2231
2232     clear A
2233
2234     drawnow % Regenera la figura i fa més sensible la GUI.
2235     set(fig.textWait,...
2236         'String','Please wait... ',...
2237         'Visible','off')
2238
2239     guidata(gcbo,fig) % Desem els canvis fets
2240     EnableDisable(gcbo,'on',0)
2241
2242     % Combinacions
2243     fig.cust_V=B;
2244     clear B
2245
2246     fig.esquerra=fig.cust_V;
2247
2248     % Mostrarem tot l'string 'c'
2249     set(fig.listboxEntrada,'String',fig.cust_V,'enable','on')
2250
2251     set(fig.buttonProcess,'enable','on');
2252 else
2253     if (isnan(fig.N)) % Hem entrat un NaN a entraN
2254         for i=1:3 % Farà pampallugues on hi ha l'error
2255             set(fig.entraN,...
2256                 'enable','off');
2257             pause(0.1)
2258             set(fig.entraN,...
2259                 'enable','on');
2260             pause(0.1)
2261         end
2262     else % Hem entrat un NaN a entraK
2263         for i=1:3 % Farà pampallugues on hi ha l'error
2264             set(fig.entraK,...
2265                 'enable','off');
2266             pause(0.1)
2267             set(fig.entraK,...
2268                 'enable','on');
2269             pause(0.1)
2270         end
2271     end

```



```

2272 end
2273
2274 clear N i ii j ord x
2275
2276 guidata(gcbo,fig) % Desem els canvis fets
2277 end
2278
2279
2280 function BotoAfegeix (varargin)
2281 fig=guidata(gcbo); % llegeix guidata.
2282
2283 val = get(fig.listboxEntrada,'Value');
2284 [tmp]=fig.esquerra{val}; % converteix del cell array a un temporal
2285
2286 % Creem un cell array nou concatenat amb els escollits anteriorment
2287 fig.dreta(length(fig.dreta)+1,1)={tmp};
2288
2289 % Agumenta en 1 posició la última posició
2290 fig.ultimaposicioDreta=fig.ultimaposicioDreta+1;
2291
2292 fig.dreta=sort(fig.dreta); % endrecem
2293
2294 % Suprimim el camp afegit al listBoxDreta del listBoxEsquerra
2295 fig.esquerra(val)=[];
2296 fig.ultimaposicioEsquerra=fig.ultimaposicioEsquerra-1;
2297
2298 % Si hem agafat l'últim element hem de posar 'Value' al penúltim:
2299 if val>length(fig.esquerra)
2300     if fig.ultimaposicioEsquerra>0
2301         set(fig.listboxEntrada,'Value',fig.ultimaposicioEsquerra);
2302     else
2303         set(fig.botoTots,'enable','off');
2304         % no pot prendre el valor 0, per tant el posem a 1:
2305         set(fig.listboxEntrada,'Value',1,'enable','off');
2306     end
2307 end
2308
2309 if (isempty(fig.esquerra)) % Si només quedava un element a la llista...
2310     % Deshabilitarem el boto d'afegir-ne més
2311     set(fig.botoAfegeix,'enable','off');
2312     set(fig.listboxEntrada,'enable','off');
2313     % Habilitem el boto de treure'n més
2314     set([fig.botoTreu fig.listboxSortida],...
2315         'enable','on');
2316     if isfield(fig,'Complete_dLibrary')
2317         set(fig.buttonProcess,'enable','on');
2318     elseif isfield(fig,'variables')
2319         set([fig.botoExport],...
2320             'enable','on');
2321     end
2322 else
2323     if (~isempty(fig.dreta))
2324         set([fig.botoTreu fig.botoCap fig.listboxSortida],...
2325             'enable','on');
2326         if isfield(fig,'Complete_dLibrary')
2327             set(fig.buttonProcess,'enable','on');

```

```

2328         elseif isfield(fig,'variables')
2329             set([fig.botoExport],...
2330                 'enable','on');
2331         end
2332     else
2333         if isfield(fig,'Complete_dLibrary')
2334             set(fig.buttonProcess,'enable','off');
2335         elseif isfield(fig,'variables')
2336             set([fig.botoExport],...
2337                 'enable','off');
2338         end
2339     end
2340     % habilitarem el boto d'afegir-ne més
2341     set(fig.botoAfegeix,'enable','on');
2342     set(fig.listboxEntrada,'enable','on');
2343 end
2344
2345 set(fig.listboxSortida,'Value',1) % Ens assegurem que té un valor>0
2346 %Ho desem a les variables
2347 set(fig.listboxSortida,'String',fig.dreta);
2348 set(fig.listboxEntrada,'String',fig.esquerra);
2349
2350 % Netegem les variables que ja no necessitarem
2351 clear val tmp
2352
2353 guidata(gcbo,fig) % Desem els canvis fets
2354 end
2355
2356
2357 function BotoTreu (varargin)
2358 fig=guidata(gcbo); % llegeix guidata.
2359
2360 val = get(fig.listboxSortida,'Value');
2361 tmp=fig.dreta{val}; % converteix del cell array a un temporal
2362
2363 fig.dreta(val)=[];
2364 fig.ultimaposicioDreta=fig.ultimaposicioDreta-1;
2365
2366 % Afegim el camp suprimit del listBox2 al listBox1
2367 fig.esquerra(length(fig.esquerra)+1,1)={tmp};
2368 fig.ultimaposicioEsquerra=fig.ultimaposicioEsquerra+1;
2369 fig.esquerra=sort(fig.esquerra);
2370
2371 % Si hem agafat l'últim element hem de posar 'Value' al penúltim.
2372 if val>length(fig.dreta)
2373     if fig.ultimaposicioDreta>0
2374         set(fig.listboxSortida,'Value',fig.ultimaposicioDreta);
2375     else
2376         set([fig.botoAfegeix fig.botoCap],'enable','off');
2377         % no pot prendre el valor 0, per tant el posem a 1
2378         set(fig.listboxSortida,'Value',1,'enable','off');
2379     end
2380 end
2381
2382 if (isempty(fig.dreta))% Si només quedava un element a la llista...
2383     % habilitarem el boto d'afegir-ne més
2384     set(fig.botoAfegeix,'enable','on');

```

```

2385     % deshabilitarem el boto de treure'n més
2386     set(fig.botoTreu, 'enable', 'off');
2387     if isfield(fig, 'Complete_dLibrary')
2388         set(fig.buttonProcess, 'enable', 'off');
2389     elseif isfield(fig, 'variables')
2390         set([fig.botoExport], ...
2391             'enable', 'off');
2392     end
2393 else
2394     if (~isempty(fig.dreta))
2395         set([fig.botoAfegeix fig.botoTots ...
2396             fig.listboxEntrada], ...
2397             'enable', 'on');
2398         if isfield(fig, 'Complete_dLibrary')
2399             set(fig.buttonProcess, 'enable', 'on');
2400         elseif isfield(fig, 'variables')
2401             set([fig.botoExport], ...
2402                 'enable', 'on');
2403         end
2404     else
2405         if isfield(fig, 'Complete_dLibrary')
2406             set(fig.buttonProcess, 'enable', 'off');
2407         elseif isfield(fig, 'variables')
2408             set([fig.botoExport], ...
2409                 'enable', 'off');
2410         end
2411     end
2412     % habilitarem el boto de treure'n més
2413     set(fig.botoTreu, 'enable', 'on');
2414 end
2415
2416 set(fig.listboxEntrada, 'Value', 1) % Ens assegurem que té un valor>0
2417 % Ho desem a les variables
2418 set(fig.listboxSortida, 'String', fig.dreta);
2419 set(fig.listboxEntrada, 'String', fig.esquerra);
2420
2421 % Netegem les variables que ja no necessitarem
2422 clear val tmp
2423
2424 guidata(gcbo, fig) % Desem els canvis fets
2425 end
2426
2427
2428 function BotoTots (varargin)
2429 fig=guidata(gcbo); % llegeix guidata.
2430
2431 if isfield(fig, 'Complete_dLibrary')
2432     fig.ultimaposicioDreta=length(fig.Complete_dLibrary); % Última
2433     % posició del listbox esquerre (màxims elements)
2434     fig.ultimaposicioEsquerra=0;
2435     fig.dreta=fig.Complete_dLibrary;
2436     fig.esquerra={};
2437 elseif isfield(fig, 'variables')
2438     fig.ultimaposicioDreta=length(fig.variables);
2439     fig.ultimaposicioEsquerra=0;
2440     fig.dreta=fig.variables;

```

```

2441     fig.esquerra={};
2442 end
2443 set(fig.listboxSortida,...
2444     'String',fig.dreta,...
2445     'Value',1);
2446 set(fig.listboxEntrada,...
2447     'String',fig.esquerra,...
2448     'Value',1);
2449
2450 if (isempty(fig.esquerra))% Si només quedava un element a la llista...
2451     % Deshabilitarem el boto d'afegir-ne més
2452     set(fig.botoAfegeix,'enable','off');
2453     set(fig.botoTots,'Enable','off');
2454     set(fig.listboxEntrada,'enable','off');
2455     % Habilitem el boto de treure'n més i el buttonProcess
2456     set([fig.botoTreu fig.botoCap fig.listboxSortida],'enable','on');
2457 end
2458 if isfield(fig,'Complete_dLibrary')
2459     set(fig.buttonProcess,'enable','on');
2460 elseif isfield(fig,'variables')
2461     set([fig.botoExport],...
2462         'enable','on');
2463 end
2464 guidata(gcbo,fig)           % Desem els canvis fets
2465 end
2466
2467
2468 function BotoCap (varargin)
2469 fig=guidata(gcbo); % llegeix guidata.
2470
2471 if isfield(fig,'Complete_dLibrary')
2472     fig.ultimaposicioEsquerra=length(fig.Complete_dLibrary);
2473     fig.ultimaposicioDreta=0;
2474     fig.dreta={};
2475     fig.esquerra=fig.Complete_dLibrary;
2476 elseif isfield(fig,'variables')
2477     fig.ultimaposicioEsquerra=length(fig.variables);
2478     fig.ultimaposicioDreta=0;
2479     fig.dreta={};
2480     fig.esquerra=fig.variables;
2481 end
2482 set(fig.listboxSortida,'String',fig.dreta,'Value',1);
2483 set(fig.listboxEntrada,'String',fig.esquerra,'Value',1);
2484
2485 if (isempty(fig.dreta))% Si només quedava un element a la llista...
2486     % habilitarem el boto d'afegir-ne més
2487     set([fig.botoAfegeix fig.botoTots...
2488         fig.listboxEntrada],'enable','on');
2489
2490     % deshabilitarem el boto de treure'n més i el buttonProcess
2491     set([fig.botoTreu fig.botoCap ...
2492         fig.listboxSortida],'enable','off');
2493     if isfield(fig,'Complete_dLibrary')
2494         set(fig.buttonProcess,'enable','off')
2495     elseif isfield(fig,'variables')
2496         set([fig.botoExport],...

```

```

2497         'enable','off');
2498     end
2499 end
2500
2501 guidata(gcbo,fig)           % Desem els canvis fets
2502 end
2503
2504
2505 function ButtonProcess (varargin)
2506 fig=guidata(gcbo); % llegeix guidata.
2507 global fileGlobal
2508 if strcmp(fig.StdOrCust,'Cust') % Si el File Type és Custom
2509     clc
2510
2511     % Recollim els nivells de descomposició
2512     fig.nivellDescomposatsJ=str2double(...
2513         get(fig.entraNivellDescomp,'String'));
2514
2515     % Fem disable en el guide
2516     EnableDisable(gcbo,'off',0) % No fem la càrrega d'un fitxer
2517
2518     fig.StdOrCust='Cust'; % Pel buttonViewRec
2519
2520     set(fig.textWait,...
2521         'String','Extracting. Please wait...',...
2522         'Visible','on',...
2523         'Enable','on')
2524     drawnow % Regenerem la figura
2525
2526
2527
2528     [fileGlobal.Bspaces,...
2529     fileGlobal.BCoefitients,...
2530     fileGlobal.D_Ranking]=f_mlldb2(...
2531         fileGlobal.xsignals,...
2532         fileGlobal.ranswer,...
2533         fig.nivellDescomposatsJ,...
2534         fig.StdOrCust,...
2535         fig.cust_Matr_V);
2536
2537     drawnow % Regenera la figura i fa més sensible la GUI.
2538
2539     guidata(gcbo,fig) % Desem els canvis fets
2540     EnableDisable(gcbo,'on',0)
2541
2542     drawnow % Regenera la figura i fa més sensible la GUI.
2543
2544     guidata(gcbo,fig) % Desem els canvis fets
2545     FetElProcess(fig)
2546     fig=guidata(gcbo); % llegeix guidata.
2547
2548
2549     if isfield(fileGlobal,'D_Ranking')
2550         if isfield(fig,'chckStrFiles')
2551             % Reiniciem
2552             fig = rmfield(fig, 'chckStrFiles');

```

```

2553     end
2554     DibuixaBspaces ();
2555     InicialitzaTaula ();
2556 end
2557
2558 set (fig.buttonRemoveRec, 'Enable', 'off');
2559
2560 set (fig.textWait, ...
2561     'String', 'Please wait...', ...
2562     'Visible', 'off')
2563 else % Si el File Type és Standard
2564     % Agafarem el valor de cadascun dels botons del panel de listboxes
2565     % per tornar-los a fer enable o no més tard, segons estiguin ara.
2566     bTots=get (fig.botoTots, 'Enable');
2567
2568     bCap=get (fig.botoCap, 'Enable');
2569
2570     bAfegeix=get (fig.botoAfegeix, 'Enable');
2571
2572     bTreu=get (fig.botoTreu, 'Enable');
2573
2574     set ([fig.botoAfegeix  fig.botoCap  ...
2575         fig.botoTots      fig.botoTreu], 'Enable', 'off');
2576
2577     % Recollim els nivells de descomposició
2578     fig.nivellDescomposatsJ=str2double (...
2579         get (fig.entraNivellDescomp, 'String'));
2580
2581     % Fem disable en el guide
2582     EnableDisable (gcbo, 'off', 0)
2583
2584     fig.StdOrCust='Std'; % Pel buttonViewRec
2585
2586     set (fig.textWait, 'String', 'Extracting. Please wait...', ...
2587         'Visible', 'on', 'enable', 'on')
2588     drawnow % Regenerem la figura
2589
2590     % Si hem seleccionat tots els diccionaris entrarà l'string 'All'
2591     if length (fig.dreta) == length (fig.Complete_dLibrary)
2592         [fileGlobal.Bspaces, ...
2593         fileGlobal.BCoefitients, ...
2594         fileGlobal.D_Ranking]=f_mlldb2 (...
2595             fileGlobal.xsignals, ...
2596             fileGlobal.ranswer, ...
2597             fig.nivellDescomposatsJ, ...
2598             fig.StdOrCust, ...
2599             'All');
2600     else
2601         for k=1:116 % Hi posem 116 perquè en 117 ja agafarà 'All'
2602             eval (sprintf ('X_%d=[];', k));
2603         end
2604         for k=1:length (fig.dreta)
2605             eval (sprintf ('X_%d=char (fig.dreta (k,1));', k));
2606         end
2607         clc
2608     % La funció found_lib de f_mlldb2 compara per string amb el cell

```

```

2609 % array Complete_dLibrary. Li he de passar string per string
2610 [fileGlobal.Bspaces, ...
2611 fileGlobal.BCoefitients, ...
2612 fileGlobal.D_Ranking]=f_mlldb2(...
2613 fileGlobal.xsignals, ...
2614 fileGlobal.ranswer, ...
2615 fig.nivellDescomposatsJ, ...
2616 fig.StdOrCust, ...
2617 X_1 ,X_2 ,X_3 ,X_4 ,X_5 ,X_6 ,X_7 ,X_8 ,X_9 ,X_10, ...
2618 X_11,X_12,X_13,X_14,X_15,X_16,X_17,X_18,X_19,X_20, ...
2619 X_21,X_22,X_23,X_24,X_25,X_26,X_27,X_28,X_29,X_30, ...
2620 X_31,X_32,X_33,X_34,X_35,X_36,X_37,X_38,X_39,X_40, ...
2621 X_41,X_42,X_43,X_44,X_45,X_46,X_47,X_48,X_49,X_50, ...
2622 X_51,X_52,X_53,X_54,X_55,X_56,X_57,X_58,X_59,X_60, ...
2623 X_61,X_62,X_63,X_64,X_65,X_66,X_67,X_68,X_69,X_70, ...
2624 X_71,X_72,X_73,X_74,X_75,X_76,X_77,X_78,X_79,X_80, ...
2625 X_81,X_82,X_83,X_84,X_85,X_86,X_87,X_88,X_89,X_90, ...
2626 X_91,X_92,X_93,X_94,X_95,X_96,X_97,X_98,X_99,X_100, ...
2627 X_101,X_102,X_103,X_104,X_105,X_106,X_107,X_108,X_109,X_110, ...
2628 X_111,X_112,X_113,X_114,X_115,X_116);
2629 end
2630
2631 % Netegem les variables que ja no necessitarem
2632 clear k X_*
2633
2634 guidata(gcbo,fig) % Desem els canvis fets
2635 EnableDisable(gcbo,'on',0)
2636
2637 drawnow % Regenera la figura i fa més sensible la GUI.
2638
2639 FetElProcess(fig)
2640
2641 if isfield(fileGlobal,'D_Ranking')
2642     if isfield(fig,'chckStrFiles')
2643         % Reiniciem
2644         fig = rmfield(fig, 'chckStrFiles');
2645     end
2646     DibuixaBspaces();
2647     InicialitzaTaula();
2648 end
2649
2650 set(fig.botoTots,'Enable',bTots);
2651 set(fig.botoCap,'Enable',bCap);
2652 set(fig.botoAfegeix,'Enable',bAfegeix);
2653 set(fig.botoTreu,'Enable',bTreu);
2654
2655 set(fig.buttonRemoveRec,'Enable','off');
2656
2657 set(fig.textWait,...
2658     'String','Please wait... ',...
2659     'Visible','off')
2660 end
2661 clear b*
2662 guidata(gcbo,fig) % Desem l'estructura
2663 end

```

```

2664
2665
2666 function MenuPolar (varargin)
2667 fig=guidata(gcbo); % llegeix guidata.
2668 global fileGlobal ContourOrProfile
2669
2670 vis=get(fig.uipanelPlot,'visible');
2671
2672 on1 =get(fig.axesSignals_VSxAux,'visible');
2673 on11=get(fig.axesSignals_VSxAuxRec,'visible');
2674 on2 =get(fig.axesSignals_VSxAuxProfile,'visible');
2675 on22=get(fig.axesSignals_VSxAuxProfileRec,'visible');
2676 on3 =get(fig.axesFix,'Visible');
2677 if strcmp(get(fig.menuPolar, 'Checked'),'off') % si menuPolar desmarcat
2678     set(fig.menuPolar, 'Checked', 'on'); % li posem la marca
2679     ContourOrProfile='Contour';
2680     if strcmp(on1,'on') && ...
2681         ~strcmp(get(fig.panelTaula,'Visible'),'on')
2682         guidata(gcbo,fig) % Desem els canvis fets
2683         ButtonViewRec(fig)
2684         fig=guidata(gcbo); % llegeix guidata.
2685
2686     elseif strcmp(on2,'on') && ~strcmp(on22,'off') &&...
2687         ~strcmp(get(fig.panelTaula,'Visible'),'on')
2688         guidata(gcbo,fig) % Desem els canvis fets
2689         ButtonViewRec(fig)
2690         fig=guidata(gcbo); % llegeix guidata.
2691     elseif strcmp(on22,'on') && strcmp(on2,'off') &&...
2692         ~strcmp(get(fig.panelTaula,'Visible'),'on') && ...
2693         isfield(fileGlobal,'selected_Coefficients')
2694         guidata(gcbo,fig) % Desem els canvis fets
2695         ButtonReconsCustDict();
2696         fig=guidata(gcbo); % llegeix guidata.
2697     elseif strcmp(on3,'on') && ...
2698         ~strcmp(get(fig.panelTaula,'Visible'),'on')
2699         guidata(gcbo,fig) % Desem els canvis fets
2700         ButtonRemove(fig)
2701         fig=guidata(gcbo); % llegeix guidata.
2702     elseif ~strcmp(get(fig.panelTaula,'Visible'),'on') && ...
2703         strcmp(get(fig.uipanelPlot,'Visible'),'on')
2704         guidata(gcbo,fig) % Desem els canvis fets
2705         ButtonView(fig)
2706         fig=guidata(gcbo); % llegeix guidata.
2707     elseif ~strcmp(get(fig.panelTaula,'Visible'),'on')
2708         guidata(gcbo,fig) % Desem els canvis fets
2709         ButtonView(fig)
2710         fig=guidata(gcbo); % llegeix guidata.
2711     end
2712
2713     if strcmp(vis,'on')
2714         set(fig.menuShowHidePlot,'Label','Hide plot menu');
2715     end
2716 else % si el menuPolar està activat, marcat
2717     set(fig.menuPolar, 'Checked', 'off'); % el desmarca
2718
2719     ContourOrProfile='Profile';
2720

```



```

2721     if strcmp(on1,'on') && ...
2722         ~strcmp(get(fig.panelTaula,'Visible'),'on') && ...
2723         ~isfield(fileGlobal,'selected_Coefficients')
2724         guidata(gcbo,fig) % Desem els canvis fets
2725         ButtonViewRec(fig)
2726         fig=guidata(gcbo); % llegeix guidata.
2727     elseif strcmp(on11,'on') && strcmp(on1,'off') && ...
2728         ~strcmp(get(fig.panelTaula,'Visible'),'on') && ...
2729         isfield(fileGlobal,'selected_Coefficients')
2730         guidata(gcbo,fig) % Desem els canvis fets
2731         ButtonReconsCustDict();
2732         fig=guidata(gcbo); % llegeix guidata.
2733
2734     elseif strcmp(on2,'on') && ...
2735         ~strcmp(get(fig.panelTaula,'Visible'),'on') && ...
2736         isfield(fileGlobal,'selected_Coefficients')
2737         guidata(gcbo,fig) % Desem els canvis fets
2738         ButtonViewRec(fig)
2739         fig=guidata(gcbo); % llegeix guidata.
2740     elseif strcmp(on3,'on') && ...
2741         ~strcmp(get(fig.panelTaula,'Visible'),'on') && ...
2742         isfield(fileGlobal,'selected_Coefficients')
2743         guidata(gcbo,fig) % Desem els canvis fets
2744         ButtonRemove(fig)
2745         fig=guidata(gcbo); % llegeix guidata.
2746     elseif ~strcmp(get(fig.panelTaula,'Visible'),'on') && ...
2747         strcmp(get(fig.uipanelPlot,'Visible'),'on')
2748         guidata(gcbo,fig) % Desem els canvis fets
2749         ButtonView(fig)
2750         fig=guidata(gcbo); % llegeix guidata.
2751     elseif ~strcmp(get(fig.panelTaula,'Visible'),'on')
2752         guidata(gcbo,fig) % Desem els canvis fets
2753         ButtonRemove(fig)
2754         fig=guidata(gcbo); % llegeix guidata.
2755     end
2756     if strcmp(vis,'on')
2757         set(fig.menuShowHidePlot,'Label','Hide plot menu');
2758     end
2759 end
2760
2761 guidata(gcbo,fig) % Desem els canvis fets
2762 end
2763
2764
2765 function MenuToolbar (varargin)
2766 % Mostra la barra d'eines
2767
2768 fig=guidata(gcbo); % llegeix guidata.
2769 if strcmp(get(fig.menuToolbar, 'Label'),'Show Toolbar')
2770     set(fig.menuToolbar,'Label','Hide Toolbar');
2771     p=get(fig.figLDB,'Position');
2772
2773     set(fig.figLDB,'Toolbar','figure',...
2774         'Position',[p(1) p(2)*2/3 p(3) p(4)]);
2775     set(findall(gcf,'tag','Standard.FileOpen'),...
2776         'visible','off');
2777     set(findall(gcf,'tag','Standard.EditPlot'),...

```

```

2778     'visible','off');
2779     set(findall(gcf,'tag','Standard.NewFigure'),...
2780         'visible','off');
2781     set(findall(gcf,'tag','Exploration.Rotate'),...
2782         'visible','off');
2783     set(findall(gcf,'tag','Annotation.InsertColorbar'),...
2784         'visible','off');
2785     set(findall(gcf,'tag','DataManager.Linking'),...
2786         'visible','off');
2787 else
2788     set(fig.figLDB,'WindowStyle','normal');
2789     plottools('off')
2790     p=get(fig.figLDB,'Position');
2791     set(fig.figLDB,'Toolbar','none',...
2792         'Position',[p(1) 44 p(3) p(4)]);
2793
2794     set(fig.menuToolbar,'Label','Show Toolbar');
2795 end
2796 guidata(gcbo,fig) % Desem els canvis fets
2797 end
2798
2799
2800 function MenuSaveSession (varargin)
2801 % Desa la sessió actual
2802
2803 fig=guidata(gcbo); % llegeix guidata.
2804 global fileGlobal fileNameGlob filePathNameGlob ContourOrProfile
2805
2806 visLeg=strcmp(get(legend,'visible'),'on');
2807 if visLeg==1
2808     [legend_h,object_h,plot_h,text_strings] = legend;
2809     text_strings=text_strings';
2810     legend('off')
2811 end
2812 toolbar=get(findall(fig.figLDB,'toolbar','figure'),'Tag');
2813
2814 tags=get(findobj(fig.figLDB,'-regex','tag','.'),'tag');
2815 visible=get(findobj(fig.figLDB,'-regex','tag','.'),'visible');
2816
2817 editTag=get(findobj(fig.figLDB,'style','edit'),'tag');
2818 edit=get(findobj(fig.figLDB,'style','edit'),'string');
2819 text=get(findobj(fig.figLDB,'style','text'),'string');
2820 textTag=get(findobj(fig.figLDB,'style','text'),'tag');
2821
2822 menuChecked=get(findall(fig.figLDB,'checked','on'),'Tag');
2823 menuString=get(findall(fig.figLDB,'type','uimenu'),'label');
2824 menuTag=get(findall(fig.figLDB,'type','uimenu'),'tag');
2825
2826 % Eliminem camps de la toolbar
2827 visible(strmatch('DataCursor',tags))=[];
2828 menuString(strmatch('DataCursor',menuTag))=[];
2829 menuTag(strmatch('DataCursor',menuTag))=[];
2830 tags(strmatch('DataCursor',tags))=[];
2831
2832 visible(strmatch('BrushSeries',tags))=[];
2833 menuString(strmatch('BrushSeries',menuTag))=[];
2834 menuTag(strmatch('BrushSeries',menuTag))=[];
2835 tags(strmatch('BrushSeries',tags))=[];

```

```

2836
2837 visible(strmatch('LiveLine',tags))=[];
2838 menuString(strmatch('LiveLine',menuTag))=[];
2839 menuTag(strmatch('LiveLine',menuTag))=[];
2840 tags(strmatch('LiveLine',tags))=[];
2841
2842 radioTag=get(findobj(fig.figLDB,'style','radiobutton'),'tag');
2843 radioVal=get(findobj(fig.figLDB,'style','radiobutton'),'Value');
2844
2845 [axes] = get(...
2846     [fig.axesFix          fig.axesPolarTots ...
2847     fig.axesSignals_VSxAux  fig.axesSignals_VSxAuxProfile...
2848     fig.axesSignals_VSxAuxProfileRec fig.axesSignals_VSxAuxRec],...
2849     'Visible');
2850
2851 if isfield(fileGlobal,'xsignals')
2852     file.xsignals=fileGlobal.xsignals;
2853 end
2854 if isfield(fileGlobal,'ranswer')
2855     file.ranswer=fileGlobal.ranswer;
2856 end
2857 if isfield(fileGlobal,'sJ')
2858     file.sJ=fileGlobal.sJ;
2859 end
2860 if isfield(fileGlobal,'Bspaces')
2861     if size(fileGlobal.Bspaces,3)>12
2862         file.Bspaces=fileGlobal.Bspaces(:, :, 1:12);
2863     else
2864         file.Bspaces=fileGlobal.Bspaces;
2865     end
2866 end
2867 stringPelWarndlg={'We will save only first 12 '};
2868 if isfield(fileGlobal,'BCoefitients')
2869     if size(fileGlobal.BCoefitients,3)>12
2870         file.BCoefitients=fileGlobal.BCoefitients(:, :, 1:12);
2871         stringPelWarndlg(end+1,1)=...
2872             {'descriptors'};
2873         switch get(fig.radioStd,'Value')
2874             case 1 % Per Str
2875                 mida=(size(fig.dreta,1));
2876                 while mida~=12
2877                     set(fig.listboxSortida,...
2878                         'Value',(size(fig.dreta,1)));
2879                     guidata(gcbo); % llegeix guidata.
2880                     BotoTreu();
2881                     fig=guidata(gcbo); % llegeix guidata.
2882                     mida=mida-1;
2883                 end
2884             case 0 % Per Cust
2885                 mida=size(fig.esquerra,1);
2886                 while mida~=12
2887                     set(fig.listboxEntrada,...
2888                         'Value',(size(fig.esquerra,1)));
2889                 end
2890                 fig.esquerra(end,:)=[];
2891

```

```

2892             mida=mida-1;
2893         end
2894         set(fig.listboxEntrada,...
2895             'String',fig.esquerra,'Value',1);
2896     end
2897 else
2898     file.BCoefitients=fileGlobal.BCoefitients;
2899 end
2900 end
2901 if isfield(fileGlobal,'D_Ranking')
2902     file.D_Ranking=fileGlobal.D_Ranking;
2903 end
2904 if isfield(fig,'cust_Matr_V')
2905     if size(fig.cust_Matr_V,1)>12
2906         cust_Matr_V=fig.cust_Matr_V(1:12,:);
2907         cust_V=fig.cust_V;
2908         switch get(fig.radioStd,'Value')
2909             case 0 % Per Cust
2910                 stringPelWarndlg(end+1,1)={...
2911                     'dictionaries'};
2912             end
2913         else
2914             cust_Matr_V=fig.cust_Matr_V;
2915             cust_V=fig.cust_V;
2916         end
2917     else
2918         cust_Matr_V=[];
2919         cust_V=[];
2920     end
2921 if isfield(fileGlobal,'xAux_Signal')
2922     file.xAux_Signal=fileGlobal.xAux_Signal;
2923 end
2924 if isfield(fileGlobal,'subspaceReconstr')
2925     if size(fileGlobal.subspaceReconstr,1)>12
2926         file.subspaceReconstr=fileGlobal.subspaceReconstr(1:12,:);
2927         stringPelWarndlg(end+1,1)={ ...
2928             'subspaces'};
2929         mida=size(fileGlobal.subspaceReconstr,1);
2930         cllListboxJKL=cell(12,1);
2931         i=1;
2932         while mida~=12
2933             str1=horzcat...
2934                 (num2str(fileGlobal.subspaceReconstr(i,1)),...
2935                 ',',...
2936                 num2str(fileGlobal.subspaceReconstr(i,2)),...
2937                 ',',...
2938                 num2str(fileGlobal.subspaceReconstr(i,3)));
2939             cllListboxJKL{i}=str1;
2940             i=i+1;
2941             mida=mida-1;
2942         end
2943     else
2944         cllListboxJKL=cell(size(fileGlobal.subspaceReconstr,1),1);
2945         for i=1:size(fileGlobal.subspaceReconstr,1)
2946             str1=horzcat...

```

```

2947         (num2str(fileGlobal.subspaceReconstr(i,1)), ...
2948         ',', ...
2949         num2str(fileGlobal.subspaceReconstr(i,2)), ...
2950         ',', ...
2951         num2str(fileGlobal.subspaceReconstr(i,3)));
2952         cllListboxJKL{i}=str1; % Conté el cell del listboxJKL
2953     end
2954     file.subspaceReconstr=fileGlobal.subspaceReconstr;
2955 end
2956 else
2957     cllListboxJKL=cell(1);
2958 end
2959 if isfield(fileGlobal, 'selected_Coefficients')
2960     if size(fileGlobal.selected_Coefficients,1)>12
2961         file.selected_Coefficients=...
2962         fileGlobal.selected_Coefficients(1:12,:);
2963         file.selected_Coefficients=fileGlobal.selected_Coefficients;
2964         switch get(fig.radioStd, 'Value')
2965             case 0 % Per Cust
2966                 stringPelWarndlg(end+1,1)={ ...
2967                 'selected coefficients'};
2968             end
2969         else
2970             file.selected_Coefficients=fileGlobal.selected_Coefficients;
2971         end
2972     end
2973
2974     if ~strcmp(stringPelWarndlg{end,1}, 'We will save only first 12 ')
2975         % Si hi hem afegit algun string
2976         str=[stringPelWarndlg{1,1} stringPelWarndlg{2,1}];
2977         if size(stringPelWarndlg,1)>2
2978             for i=3:size(stringPelWarndlg,1)-1
2979                 str=[str ' ' stringPelWarndlg{i,1}];
2980             end
2981             str=[str ' and ' stringPelWarndlg{end,1} '.'];
2982         end
2983         warndlg(str, 'Save session', 'modal');
2984     end
2985
2986     session={file fileNameGlob filePathNameGlob ...
2987             ContourOrProfile tags visible ...
2988             editTag edit text textTag toolbar...
2989             menuChecked menuString menuTag axes radioTag radioVal...
2990             fig.esquerra fig.dreta cust_Matr_V cust_V cllListboxJKL};
2991
2992     % Si volem desar al userpath:
2993     % up=strrep(userpath, ';', '');
2994     % if ~isempty(up)
2995     %     cami=strcat(up, '/savedSession.mat');
2996     %     save(cami, 'session');
2997     % else
2998     %     save('savedSession.mat', 'session');
2999     % end
3000 if visLeg==1
3001     legend(text_strings)

```

```

3002 end
3003 set(fig.textWait,...
3004     'Enable','on',...
3005     'String','Saved',...
3006     'Visible','on')
3007 pause(2)
3008
3009 set(fig.textWait,...
3010     'String','Please wait...',...
3011     'Visible','off')
3012 clear file session *On *Off strings toolbar menu* text* *_h ...
3013     *Tag uicon* tags
3014 guidata(gcbo,fig) % Desem els canvis fets
3015 end
3016
3017
3018 function MenuLoadSession (varargin)
3019 % Carrega la sessio desada
3020
3021 fig=guidata(gcbo); % llegeix guidata.
3022 global fileGlobal fileNameGlob filePathNameGlob ContourOrProfile
3023 % Si existeix el fitxer
3024 % Si la sessió és al userpath:
3025 % up=strrep(userpath,',' '\'');
3026 % cami=strcat(up,'/savedSession.mat');
3027 % Sinó:
3028 cami='savedSession.mat';
3029
3030 if exist(cami,'file')==2
3031
3032     EnableDisable(gcbo,'off',1) % Posa a zero la interfície
3033     fig=guidata(gcbo); % llegeix guidata.
3034
3035     set(fig.textWait,...
3036         'String','Please wait...',...
3037         'Visible','on',...
3038         'enable','on');
3039     drawnow
3040
3041     session=cell(1,22); % Té 22 camps
3042
3043     load(cami);
3044
3045     fileGlobal = session{1,1};
3046     fileNameGlob = session{1,2};
3047     filePathNameGlob = session{1,3};
3048     ContourOrProfile = session{1,4};
3049     tags = session{1,5};
3050     visible = session{1,6};
3051     editTag = session{1,7};
3052     edit = session{1,8};
3053     text = session{1,9};
3054     textTag = session{1,10};
3055     toolbar = session{1,11};
3056     menuChecked = session{1,12};
3057     menuString = session{1,13};
3058     menuTag = session{1,14};

```

```

3059 axes = session{1,15};
3060 radioTag = session{1,16};
3061 radioVal = session{1,17};
3062 fig.cust_Matr_V = session{1,20};
3063 fig.cust_V = session{1,21};
3064 cllListboxJKL= session{1,22};
3065
3066 for i=1:size(menuTag,1) % Canviem els 'Label'
3067     if ~isempty(menuTag{i})
3068         set(fig.(char(menuTag(i,:))),...
3069             'Label',(char(menuString(i,:))));
3070     end
3071 end
3072
3073 guidata(gcbo,fig) % Desem els canvis fets
3074 EnableDisable(gcbo,'on',0)
3075 fig=Modif_interf_inicial(gcbo,1); % 1: nova sessió
3076
3077
3078 set(fig.textWait,...
3079     'String','Please wait.....',...
3080     'Visible','on');
3081
3082 for i=1:size(editTag,1) % Canviem els 'String' dels edit
3083     if ~isempty(editTag{i})
3084         set(fig.(char(editTag(i,:))),...
3085             'String',num2str(char(edit{i,:})));
3086     end
3087 end
3088
3089 for i=1:size(textTag,1) % Canviem els 'string' dels text
3090     if ~isempty(textTag{i})
3091         set(fig.(char(textTag(i,:))), 'String', (char(text{i,:})));
3092     end
3093 end
3094 for i=1:size(radioTag,1) % Canviem les 'value'
3095     if ~isempty(radioTag{i})
3096         set(fig.(char(radioTag(i,:))),...
3097             'Value',((radioVal{i,:})));
3098     end
3099 end
3100 % Recollim ara esquerra i dreta, perquè EnableDisable les posa a 0
3101 fig.esquerra = session{1,18};
3102 fig.dreta = session{1,19};
3103
3104 for i=1:size(tags,1) % Elements visibles i invisibles
3105     if ~isempty(tags{i})
3106         set(fig.(char(tags(i))), 'visible', (char(visible{i,1})));
3107     end
3108 end
3109
3110 set(fig.listboxFEWPT, 'String', fig.Complete_dLibrary);
3111 if size(fig.esquerra,1) <= size(fig.Complete_dLibrary,1) && ...
3112     get(fig.radioCust, 'Value')==0
3113     fig.StdOrCust='Std'; % Serà Standard

```

```

3114     esquerda=fig.esquerra;
3115     dreta=fig.dreta;
3116     RadioStd();
3117     fig.esquerra=esquerra;
3118     fig.dreta=dreta;
3119     set (fig.listBoxEntrada, ...
3120         'String', fig.esquerra)
3121     set (fig.listBoxSortida, ...
3122         'String', fig.dreta)
3123     fig.ultimaPosicioEsquerra=length(fig.esquerra);
3124     fig.ultimaPosicioDreta=length(fig.dreta);
3125     if isempty(fig.dreta)
3126         set([fig.botoTreu fig.botoCap fig.listBoxSortida], ...
3127             'Visible', 'on', ...
3128             'enable', 'off');
3129     elseif isempty(fig.esquerra)
3130         set([fig.botoAfegeix fig.botoTots ...
3131             fig.listBoxEntrada fig.buttonProcess], ...
3132             'Visible', 'on', ...
3133             'enable', 'off');
3134     else
3135         set([fig.botoTreu fig.botoCap ...
3136             fig.botoAfegeix fig.botoTots ...
3137             fig.listBoxSortida fig.listBoxEntrada ...
3138             fig.buttonProcess], ...
3139             'Visible', 'on', ...
3140             'enable', 'on');
3141     end
3142
3143     elseif get (fig.radioCust, 'Value')==1
3144         fig.StdOrCust='Cust'; % Serà Custom
3145         p=get (fig.uipanelCustNK, 'Position');
3146         set (fig.uipanelListboxes, ...
3147             'Visible', 'on', ...
3148             'Title', '', ...
3149             'Position', [fig.margeP
3150                 (p(2)-fig.panelDiscrimAnalysisMenuAlcada/2) ...
3151                 fig.midaAmpPanInt
3152                 fig.panelDiscrimAnalysisMenuAlcada/2-10    ])
3153         p=get (fig.uipanelListboxes, 'Position');
3154
3155         mde=6; % Marge dret, esquerre, dalt i baix entre listboxes i
3156             % panel.
3157
3158         alcLB=(p(4)-2*mde); % Alçada dels Listboxes
3159         ampLBCust=p(3)-2*mde;
3160
3161         set (fig.listBoxEntrada, ...
3162             'Position', [mde
3163                 mde
3164                 ampLBCust
3165                 alcLB    ], ...
3166             'String', fig.esquerra)
3167
3168         set (fig.listBoxEntrada, 'String', fig.cust_V)

```



```

3169         if isfield(fig,'cust_Matr_V')
3170             set(fig.buttonProcess,'enable','on');
3171         end
3172     end
3173
3174     if isempty(toolbar) % Si hi havia la toolbar que vagi al Toolbar
3175         set(fig.menuToolbar,'Label','Hide Toolbar');
3176         MenuToolbar ();
3177     else
3178         set(fig.menuToolbar,'Label','Show Toolbar');
3179         MenuToolbar ();
3180     end
3181
3182     if isfield(fileGlobal,'subspaceReconstr')
3183         set(fig.listboxJKL,'value',1,'String',c11ListboxJKL);
3184     end
3185
3186     if isfield (fileGlobal,'D_Ranking') && ... % Taula i Bspaces
3187         strcmp (get (fig.panelTaula, 'Visible'), 'on')
3188         DibuixaBspaces ();
3189         InicialitzaTaula ();
3190         a=fileGlobal.D_Ranking(:,1);
3191         set (fig.popupLibrary, 'String',a, 'Enable', 'on', 'Value',1);
3192
3193         set (fig.radioSelect, 'Enable', 'on', 'Value',1);
3194         set (fig.editSelect, 'Enable', 'on');
3195
3196         p=get (fig.textExempleSeleccio, 'Position');
3197         str=[ 'Write numbers and intervals separated by comma. Ex: 1,3,5-
3198             12,14. Max range: 1-' ...
3199             num2str (size (fileGlobal.Bspaces,1))] ;
3200         [outstring pos] = textwrap (fig.textExempleSeleccio,str);
3201         set (fig.textExempleSeleccio, ...
3202             'String',outstring, ...
3203             'Enable', 'on', ...
3204             'Position', [p(1) ...
3205                 p(2) ...
3206                 p(3) ...
3207                 pos(4)]);
3208     end
3209
3210     % (fig.axesFix axes{1});
3211     % (fig.axesPolarTots axes{2});
3212     % (fig.axesSignals_VSxAux axes{3});
3213     % (fig.axesSignals_VSxAuxProfile axes{4});
3214     % (fig.axesSignals_VSxAuxProfileRec axes{5});
3215     % (axesSignals_VSxAuxRec axes{6});
3216
3217     % Dibuixem els diferents axis
3218     if strcmp (axes{4}, 'on') && ... % axesSignals_VSxAuxProfile
3219         ~strcmp (get (fig.panelTaula, 'Visible'), 'on')
3220
3221         if isfield (fileGlobal, 'xsignals') && ...
3222             isfield (fileGlobal, 'xAux_Signal')
3223
3224             set (fig.textWait, ...

```

```

3225         'String','Drawing.....',...
3226         'Visible','on');
3227
3228     drawnow
3229
3230     AxesResetInvisible();
3231
3232     zoom off
3233
3234     if strcmp(ContourOrProfile,'Profile')
3235         FuncPlot(fig.axesSignals_VSxAuxProfile,...
3236                 'Original signal',...
3237                 fileGlobal.xsignals,fileGlobal.ranswer)
3238         FuncPlot(fig.axesxAux_VSsignalsProfile,...
3239                 'Signal reconstructed',...
3240                 fileGlobal.xAux_Signal,fileGlobal.ranswer)
3241     else
3242         % Farà que es vegi correctament el caràcter _:
3243         str = strrep(fileNameGlob, '_', '\_');
3244         str=['(' str ')']; % Hi afegim 'Totes les parts'
3245         str=['Contour view ' str];
3246         FuncPolar(fig.axesPolarTots,str,...
3247                 fileGlobal.xsignals,fileGlobal.ranswer)
3248
3249         %-----%
3250         % I ara que dibuixi les dues figures de sota
3251         FuncPlot(fig.axesSignals_VSxAux,'Original signal',...
3252                 fileGlobal.xsignals,fileGlobal.ranswer)
3253         FuncPlot(...
3254                 fig.axesxAux_VSsignals,'Signal reconstructed',...
3255                 fileGlobal.xAux_Signal,fileGlobal.ranswer)
3256
3257         clear linia angle fi i j str
3258     end
3259     set(fig.textWait,...
3260         'String','Please wait... ',...
3261         'Visible','off');
3262
3263     elseif ~isfield(fileGlobal,'xAux_Signal')
3264         guidata(gcbo,fig) % Desem els canvis fets
3265         ButtonViewRec(fig)
3266         fig=guidata(gcbo); % llegeix guidata.
3267     end
3268     elseif strcmp(axes{1},'on') % axesFix
3269         guidata(gcbo,fig) % Desem els canvis fets
3270         ButtonRemove();
3271         fig=guidata(gcbo); % llegeix guidata.
3272     elseif strcmp(axes{2},'on') % AxesPolarTots
3273         if strcmp(ContourOrProfile,'Contour')
3274             guidata(gcbo,fig) % Desem els canvis fets
3275             % Farà que es vegi correctament el caràcter _:
3276             str = strrep(fileNameGlob, '_', '\_');
3277             str=['(' str ')']; % Hi afegim 'Totes les parts'
3278             str=['Contour view ' str];
3279             FuncPolar(fig.axesPolarTots,str,...
3280                     fileGlobal.xsignals,fileGlobal.ranswer)
3281

```

```

3282         %-----%
3283         % I ara que dibuixi les dues figures de sota
3284         FuncPlot(fig.axesSignals_VSxAux,'Original signal',...
3285             fileGlobal.xsignals,fileGlobal.ranswer)
3286         FuncPlot(fig.axesxAux_VSsignals,'Signal reconstructed',...
3287             fileGlobal.xAux_Signal,fileGlobal.ranswer)
3288         fig=guidata(gcbo); % llegeix guidata.
3289     end
3290 elseif strcmp(axes{3},'on') && ... % axesSignals_VSxAux
3291     ~strcmp(get(fig.panelTaula,'Visible'),'on')
3292     guidata(gcbo,fig) % Desem els canvis fets
3293     ButtonViewRec(fig)
3294     fig=guidata(gcbo); % llegeix guidata.
3295 elseif strcmp(axes{5},'on') && ... % axesSignals_VSxAuxProfileRec
3296     ~strcmp(get(fig.panelTaula,'Visible'),'on')
3297     guidata(gcbo,fig) % Desem els canvis fets
3298     ButtonReconsCustDict(fig)
3299     fig=guidata(gcbo); % llegeix guidata.
3300 elseif strcmp(axes{6},'on') && ... % axesSignals_VSxAuxRec
3301     ~strcmp(get(fig.panelTaula,'Visible'),'on')
3302     guidata(gcbo,fig) % Desem els canvis fets
3303     ButtonReconsCustDict(fig)
3304     fig=guidata(gcbo); % llegeix guidata.
3305 elseif ~strcmp(get(fig.panelTaula,'Visible'),'on')
3306     % axesPlot
3307     guidata(gcbo,fig) % Desem els canvis fets
3308     ButtonView();
3309     fig=guidata(gcbo); % llegeix guidata.
3310 end
3311 guidata(gcbo,fig) % Desem els canvis fets
3312 for i=1:size(menuChecked,1) % Marquem els 'checked'
3313     switch char(menuChecked(i,:))
3314         case 'menuLDB'
3315             % l'hi enviem inicialment desmarcat
3316             set(fig.menuLDB,'checked','off')
3317             MenuLDB();
3318         case 'menuWaveletPT'
3319             % l'hi enviem inicialment desmarcat
3320             set(fig.menuWaveletPT,'checked','off')
3321             MenuWaveletPT();
3322         case 'menuPolar'
3323             % l'hi enviem inicialment desmarcat
3324             set(fig.menuPolar,'checked','on')
3325     end
3326 end
3327 end
3328
3329 set(fig.menuCloseSession,'Enable','on'); % Habilitem el menú
3330
3331 set(fig.textWait,...
3332     'String','Please wait...',...
3333     'Visible','off');
3334 else
3335     set(fig.textWait,...
3336         'Enable','on',...
3337         'String','No session saved',...
3338         'Visible','on')

```

```

3339     pause(2)
3340
3341     set(fig.textWait,...
3342         'String','Please wait...',...
3343         'Visible','off')
3344 end
3345
3346 guidata(gcbo,fig) % Desem els canvis fets
3347 end
3348
3349
3350 function MenuCloseSession (varargin)
3351
3352 EnableDisable(gcbo,'off',1) % Posa a zero la interfície
3353 fig=guidata(gcbo); % llegeix guidata.
3354
3355 % Canviant el nom del menú i cridant després la funció MenuToolbar ens
3356 % assegurem que la barra d'eines desapareix, ja que la funció funciona
3357 % comparant el nom del menú
3358 set(fig.menuToolbar, 'Label','Hide Toolbar')
3359 MenuToolbar();
3360
3361 clear global % esborra totes les variables globals
3362 global fileGlobal fileNameGlob filePathNameGlob ContourOrProfile
3363 fileGlobal=struct; % Inicialitzem variables
3364 filePathNameGlob=char;
3365 fileNameGlob=char;
3366 ContourOrProfile='Profile';
3367
3368 set(fig.menuCloseSession,'Enable','off'); % Inhabilitem el menú
3369
3370 guidata(gcbo,fig) % Desem els canvis fets
3371 end
3372
3373
3374 function MenuShowHidePlot (varargin)
3375 fig=guidata(gcbo); % llegeix guidata.
3376
3377 if strcmp(get(gcbo, 'Label'),'Show plot menu')
3378     % si el menuShowHidePlot mostra Hide
3379     set([fig.uipanelReconstruct fig.uipanelDiscrimAnalysis ...
3380         fig.uipanelWaveletPT fig.uipanelReconstructCustDict],...
3381         'Visible','off');
3382     set([fig.menuLDB fig.menuWaveletPT], ...
3383         'Checked', 'off');
3384
3385     % li diem que mostri 'Hide plot menu'
3386     set(gcbo,'Label','Hide plot menu');
3387     % i farà visible el uipanelPlot
3388     set(fig.uipanelPlot,...
3389         'Visible','on',...
3390         'Position',[fig.panelsX
3391                     (fig.panelPlotY-fig.panelSignalAlcada) ...
3392                     fig.panelsAmplada
3393                     fig.panelPlotAlcada
3394                     ]);

```

```

3395 else % si el menuPolar està activat, marcat
3396     set(gcbo,'Label','Show plot menu');
3397     % li diem que mostri 'Show plot menu'
3398
3399     % Fa invisible el uipanelPlot
3400     set(fig.uipanelPlot,'Visible','off');
3401 end
3402
3403 guidata(gcbo,fig) % Desem els canvis fets
3404 end
3405
3406
3407 function MenuWaveletPT (varargin)
3408 fig=guidata(gcbo); % llegeix guidata.
3409 % si el menuWaveletPT està desmarcat:
3410 if strcmp(get(fig.menuWaveletPT, 'Checked'),'off')
3411     % Si està desmarcat
3412     set([fig.uipanelDiscrimAnalysis fig.uipanelPlot ...
3413         fig.uipanelReconstruct fig.uipanelReconstructCustDict],...
3414         'Visible','off');
3415     set(fig.menuLDB,'Checked','off');
3416
3417     set(fig.menuShowHidePlot,'Label', 'Show plot menu');
3418
3419     set(fig.menuWaveletPT, 'Checked', 'on'); % li posem la marca
3420     % fa visible el uipanelWaveletPT
3421     set(fig.uipanelWaveletPT,...
3422         'Visible','on');
3423
3424 else % si el menuWaveletPT està activat, marcat
3425     set(fig.menuWaveletPT, 'Checked', 'off'); % el desmarca
3426     % i fa invisible el uipanelWaveletPT
3427     set(fig.uipanelWaveletPT,'Visible','off')
3428
3429 end
3430
3431 guidata(gcbo,fig) % Desem els canvis fets
3432 end
3433
3434
3435 function MenuLDB (varargin)
3436 fig=guidata(gcbo); % llegeix guidata.
3437 if strcmp(get(fig.menuLDB, 'Checked'),'off')
3438     % si el menuLDB està desmarcat:
3439     if size(fig.esquerra,1)<size(fig.Complete_dLibrary,1) && ...
3440         get(fig.radioCust,'Value')==0
3441         % Assegurems-nos que buttonProcess queda a enable off
3442         % si fig.dreta és buit
3443         set(fig.buttonProcess,'Enable','on');
3444     elseif size(fig.esquerra,1)<size(fig.Complete_dLibrary,1) && ...
3445         isempty(fig.dreta) && ~isempty(fig.esquerra) && ...
3446         get(fig.radioCust,'Value')==1
3447         set(fig.buttonProcess,'Enable','on');
3448     else
3449         set(fig.buttonProcess,'Enable','off');
3450     end

```

```

3451
3452     % Fem invisibles
3453     set([fig.uipanelWaveletPT fig.uipanelPlot ...
3454         fig.uipanelReconstruct fig.uipanelReconstructCustDict],...
3455         'Visible','off');
3456     set(fig.menuWaveletPT,'Checked','off');
3457
3458     set(fig.menuShowHidePlot,'Label','Show plot menu');
3459
3460     if get(fig.radioStd,'Value')==1
3461         guidata(gcbo,fig) % Desem els canvis fets
3462         BotoCap();
3463         fig=guidata(gcbo); % llegeix guidata.
3464     end
3465
3466     set(fig.menuLDB,'Checked','on'); % li posem la marca
3467     set(fig.uipanelDiscrimAnalysis,...
3468         'Visible','on');
3469
3470     else % si el menuLDB està activat, marcat
3471         set(fig.menuLDB,'Checked','off'); % el desmarca
3472         % i fa invisible el uipanelDiscrimAnalysis
3473         set(fig.uipanelDiscrimAnalysis,'Visible','off')
3474     end
3475
3476     % Netegem les variables que ja no necessitarem
3477     clear vis*
3478
3479     guidata(gcbo,fig) % Desem els canvis fets
3480     end
3481
3482
3483     function CheckGrid (varargin)
3484     fig=guidata(gcbo); % llegeix guidata.
3485     if(get(gcbo,'Value'))
3486         axes(fig.axesPlot);
3487         grid on;
3488     else
3489         axes(fig.axesPlot);
3490         grid off;
3491     end
3492     end
3493
3494
3495     function CheckPan (varargin)
3496     fig=guidata(gcbo); % llegeix guidata.
3497     if(get(gcbo,'Value'))
3498         set(fig.checkZoom,'Value',0);
3499         zoom off;
3500         % desactiva el data cursor mode en l'actual figura.
3501         datacursormode off;
3502         set(fig.checkCursor,'Value',0);
3503         axes(fig.axesPlot);
3504         pan on;
3505     else
3506         axes(fig.axesPlot);
3507         pan off;

```

```

3508 end
3509 guidata(gcbo, fig);
3510 end
3511
3512
3513 function CheckZoom (varargin)
3514 fig=guidata(gcbo); % llegeix guidata.
3515 if(get(gcbo,'Value'))
3516     set(fig.checkPan,'Value',0);
3517     set(fig.checkCursor,'Value',0);
3518     pan off;
3519     datacursormode off;
3520     axes(fig.axesPlot);
3521     zoom on;
3522 else
3523     axes(fig.axesPlot);
3524     zoom off;
3525 end
3526 guidata(gcbo, fig);
3527 end
3528
3529
3530 function CheckCursor (varargin)
3531 fig=guidata(gcbo); % llegeix guidata.
3532 if(get(gcbo,'Value'))
3533     set(fig.checkPan,'Value',0);
3534     set(fig.checkZoom,'Value',0);
3535     pan off;
3536     zoom off;
3537     axes(fig.axesPlot);
3538     datacursormode on;
3539 else
3540     axes(fig.axesPlot);
3541     datacursormode off;
3542 end
3543 guidata(gcbo, fig);
3544 end
3545
3546
3547 function MenuOpenWizard (varargin)
3548 % Obre l'Asistent
3549 fig=guidata(gcbo); % llegeix guidata.
3550 global ContourOrProfile loadOk
3551 fig.contourOrProfiletmp=ContourOrProfile;
3552
3553 LDB_Wizard(fig);
3554 uiwait(gcf)
3555
3556 if loadOk==1
3557     fig.StdOrCust='Std';
3558     RadioStd();
3559 end
3560
3561 clear loadOk
3562
3563 guidata(gcbo, fig);

```

```

3564 end
3565
3566
3567 function RadioSelect (varargin)
3568 fig=guidata(gcbo); % llegeix guidata.
3569
3570 set(fig.radioSelect,'Value',1)
3571
3572 guidata(gcbo, fig);
3573 end
3574
3575
3576 function ButtonViewRec(varargin)
3577 % Botó View pel panel 'Reconstruct'
3578
3579 fig=guidata(gcbo); % llegeix guidata.
3580
3581 global fileGlobal fileNameGlob ContourOrProfile
3582
3583 EnableDisable(gcbo,'off',0)
3584
3585 if isfield(fileGlobal,'xAux_Signal')
3586     fileGlobal = rmfield(fileGlobal, 'xAux_Signal');
3587 end
3588
3589 set(fig.textWait,...
3590     'Enable','on',...
3591     'String','Reconstructing...',...
3592     'Visible','on')
3593
3594 drawnow
3595
3596 % radio=get(fig.radioAll,'Value');
3597
3598 val=get(fig.popupLibrary,'Value'); %Obtenim el nom de la llibreria
3599 str=char(fileGlobal.D_Ranking(val,1));
3600
3601 % En cas que el D_Ranking sigui del Std
3602 % [LO_D,HI_D,LO_R,HI_R] = WFILTERS('wname') computes four
3603 % filters associated with the orthogonal or biorthogonal
3604 % wavelet named in the string 'wname'.
3605 %     LO_R, the reconstruction low-pass filter
3606 %     HI_R, the reconstruction high-pass filter
3607 if strcmp(fig.StdOrCust,'Std') % Variable que ve del ButtonProcess
3608     [LO_R,HI_R]=wfilters(str,'r'); % 'r': (Reconstruction filters)
3609 else
3610     % En cas que el D_Ranking vingui del Cust
3611     [LO_R, HI_R]=Paraunitary_m(fig.cust_Matr_V(val,:), 'r');
3612 end
3613 clear str
3614
3615 cadena=get(fig.editSelect,'String');
3616 set(fig.editSelect,'enable','off');
3617
3618 vect=ComparaCadena(cadena);
3619
3620 set(fig.editSelect,'enable','on');

```



```

3621
3622 if isempty(vect)
3623     fileGlobal.xAux_Signal=fIWPT_RecTree(...
3624         fileGlobal.BCoefitients(vect, :, val), ...
3625         fileGlobal.Bspaces(vect, 2:4, val), ...
3626         size(fileGlobal.BCoefitients, 1), ...
3627         LO_R, HI_R);
3628
3629     set(fig.textWait, ...
3630         'Visible', 'on');
3631 else
3632     for i=1:3 % Farà pampallugues on hi ha l'error
3633         set(fig.editSelect, ...
3634             'enable', 'off');
3635         pause(0.1)
3636         set(fig.editSelect, ...
3637             'enable', 'on');
3638         pause(0.1)
3639     end
3640 end
3641
3642
3643 clear val radio cadena vect HI_R LO_R
3644
3645 if isfield(fileGlobal, 'xsignals') && isfield(fileGlobal, 'xAux_Signal')
3646     set(fig.textWait, ...
3647         'String', 'Drawing.....', ...
3648         'Visible', 'on');
3649
3650     AxesResetInvisible();
3651
3652     zoom off
3653     set(fig.panelTaula, 'Visible', 'off')
3654
3655     if strcmp(ContourOrProfile, 'Profile')
3656         FuncPlot(fig.axesSignals_VSxAuxProfile, ...
3657             'Original signal', ...
3658             fileGlobal.xsignals, fileGlobal.ranswer)
3659         FuncPlot(fig.axesxAux_VSsignalsProfile, ...
3660             'Signal reconstructed', ...
3661             fileGlobal.xAux_Signal, fileGlobal.ranswer)
3662     else
3663         % Farà que es vegi correctament el caràcter _:
3664         str = strrep(fileNameGlob, '_', '\_');
3665         str=['(' str ')']; % Hi afegim 'Totes les parts'
3666         str=['Contour view ' str];
3667         FuncPolar(fig.axesPolarTots, str, ...
3668             fileGlobal.xsignals, fileGlobal.ranswer)
3669
3670         %-----%
3671         % I ara que dibuixi les dues figures de sota
3672         FuncPlot(fig.axesSignals_VSxAux, 'Original signal', ...
3673             fileGlobal.xsignals, fileGlobal.ranswer)
3674         FuncPlot(fig.axesxAux_VSsignals, 'Signal reconstructed', ...
3675             fileGlobal.xAux_Signal, fileGlobal.ranswer)

```

```

3676
3677     clear linia angle fi i j str
3678 end
3679 set(fig.textWait,...
3680     'String','Please wait...',...
3681     'Visible','off');
3682 end
3683
3684 set(fig.textWait,...
3685     'String','Please wait...',...
3686     'Visible','off')
3687
3688 EnableDisable(gcbo,'on',0)
3689
3690 set(fig.buttonRemoveRec,'Enable','on');
3691
3692 guidata(gcbo, fig);
3693 end
3694
3695
3696 function ButtonRemoveRec(varargin)
3697
3698 ButtonRemove();
3699
3700 fig=guidata(gcbo); % llegeix guidata.
3701 set(fig.buttonRemoveRec,'Enable','off');
3702 guidata(gcbo, fig);
3703 end
3704
3705
3706 function MenuSave (varargin)
3707 global fileGlobal filePathNameGlob fileNameGlob
3708 fig=guidata(gcbo); % llegeix guidata.
3709 % Demanem de sobreesciure
3710 surt=questdlg('Rewrite data on the disc?','Save','Yes','No','Yes');
3711 if strcmp(surt,'Yes')
3712     if isfield(fileGlobal,'xsignals')||...
3713         isfield(fileGlobal,'ranswer')||...
3714         isfield(fileGlobal,'sJ')||...
3715         isfield(fileGlobal,'Bspaces')||...
3716         isfield(fileGlobal,'BCoefitients')||...
3717         isfield(fileGlobal,'D_Ranking')||...
3718         isfield(fileGlobal,'xAux_Signal')||...
3719         isfield(fileGlobal,'selected_Coefficients')
3720     if ~strcmp(fileNameGlob,'Workspace')
3721         % Podem gravar l'estructura fileGlobal directament o variable
3722         % per variable. Ho farem variable per variable.
3723         save(filePathNameGlob,'-struct','fileGlobal')
3724         % Li diem que sobreescrigui el fitxer obert amb totes les
3725         % variables que tingui a 'file'
3726     else
3727         vars=fieldnames(fileGlobal);
3728         for i=1:size(vars,1)
3729             assignin('base',(char(vars(i,1))),...
3730                 fileGlobal.(char(vars(i,1))));

```

```

3731         end
3732     end
3733     str=['Saved to ' fileNameGlob];
3734     set(fig.textWait,...
3735         'Enable','on',...
3736         'String',str,...
3737         'Visible','on')
3738     pause(3)
3739
3740     set(fig.textWait,...
3741         'String','Please wait...',...
3742         'Visible','off')
3743     end
3744 else
3745     guidata(gcbo,fig) % Desem els canvis fets
3746 end
3747 end
3748
3749
3750 function SliderNivellDescomp (varargin)
3751 fig=guidata(gcbo); % llegeix guidata.
3752
3753 sl=get(gcbo,'Value');
3754 set(fig.entraNivellDescomp,'String',sl);
3755
3756 clear sl
3757 guidata(gcbo,fig) % Desem els canvis fets
3758 end
3759
3760
3761 function EntraNivellDescomp (varargin)
3762 fig=guidata(gcbo); % llegeix guidata.
3763 global fileGlobal
3764
3765 x=str2double(get(fig.entraNivellDescomp,'String'));
3766 if (isnan(x))
3767     set(fig.entraNivellDescomp,'Enable','off');
3768     set(fig.textWait,...
3769         'String','You must enter a numeric value',...
3770         'Enable','on',...
3771         'Visible','on');
3772     pause(1.5)
3773     set(fig.entraNivellDescomp,'String',fileGlobal.sJ);
3774     pause(1.5)
3775     set(fig.entraNivellDescomp,'Enable','on');
3776     set(fig.textWait,...
3777         'String','Please wait...',...
3778         'Visible','off');
3779 end
3780 x=str2double(get(fig.entraNivellDescomp,'String'));
3781 if mod(x,1)~=0, % si no es enter
3782     x=round(x); % l'arrodonirem a l'enter més pròxim
3783     set(fig.entraNivellDescomp,'String',x);
3784 end
3785

```

```

3786 min=get(fig.sliderNivellDescomp,'Min');
3787 x=str2double(get(fig.entraNivellDescomp,'String'));
3788 if x<min % Si hi posem 0, o hi queda per arrodoniment
3789     set(fig.entraNivellDescomp,'String',min); % que entri 1
3790 end
3791
3792 max=get(fig.sliderNivellDescomp,'Max');
3793 x=str2double(get(fig.entraNivellDescomp,'String'));
3794 if x>max % Si hi posem una dada major que la permesa
3795     set(fig.entraNivellDescomp,'String',max); % que entri la màxima
3796 end
3797
3798 % Netegem les variables que ja no necessitarem
3799 clear x max min
3800
3801 guidata(gcbo,fig) % Desem els canvis fets
3802 end
3803
3804
3805 function SliderNivellDescompWPT (varargin)
3806 fig=guidata(gcbo); % llegeix guidata.
3807
3808 sl=get(gcbo,'Value');
3809 set(fig.entraNivellDescompWPT,'String',sl);
3810
3811 clear sl
3812
3813 guidata(gcbo,fig) % Desem els canvis fets
3814 end
3815
3816
3817 function EntraNivellDescompWPT (varargin)
3818 fig=guidata(gcbo); % llegeix guidata.
3819 global fileGlobal
3820
3821 x=str2double(get(fig.entraNivellDescompWPT,'String'));
3822 if (isnan(x))
3823     set(fig.entraNivellDescompWPT,'Enable','off');
3824     set(fig.textWait,...
3825         'String','You must enter a numeric value',...
3826         'Enable','on',...
3827         'Visible','on');
3828     pause(1.5)
3829     set(fig.entraNivellDescompWPT,'String',fileGlobal.sJ);
3830     pause(1.5)
3831     set(fig.entraNivellDescompWPT,'Enable','on');
3832     set(fig.textWait,...
3833         'String','Please wait...',...
3834         'Visible','off');
3835 end
3836 x=str2double(get(fig.entraNivellDescompWPT,'String'));
3837 if mod(x,1)~=0, % si no es enter
3838     x=round(x); % l'arrodonirem a l'enter més pròxim
3839     set(fig.entraNivellDescompWPT,'String',x);
3840 end
3841

```

```

3842 min=get(fig.sliderNivellDescompWPT,'Min');
3843 x=str2double(get(fig.entraNivellDescompWPT,'String'));
3844 if x<min % Si hi posem 0, o hi queda per arrodoniment
3845     set(fig.entraNivellDescompWPT,'String',min); % que entri 1
3846 end
3847
3848 max=get(fig.sliderNivellDescompWPT,'Max');
3849 x=str2double(get(fig.entraNivellDescompWPT,'String'));
3850 if x>max % Si hi posem una dada major que la permesa
3851     set(fig.entraNivellDescompWPT,'String',max); % que entri la màxima
3852 end
3853
3854 % Netegem les variables que ja no necessitarem
3855 clear x max min
3856
3857 guidata(gcbo,fig) % Desem els canvis fets
3858 end
3859
3860
3861 function ButtonProcessWPT (varargin)
3862 fig=guidata(gcbo); % llegeix guidata.
3863 global fileGlobal
3864
3865 clc
3866
3867 % Fem disable en el guide
3868 EnableDisable(gcbo,'off',0)
3869 set(fig.textWait,...
3870     'String','Extracting. Please wait...',...
3871     'Visible','on',...
3872     'enable','on')
3873 drawnow % Regenerem la figura
3874
3875 % Recollim els nivells de descomposició
3876 fig.nivellDescomposatsJ=str2double(...
3877     get(fig.entraNivellDescompWPT,'String'));
3878
3879 val=get(fig.listboxFEWPT,'Value'); %Obtenim el nom de la llibreria
3880 str=char(fig.Complete_dLibrary(val,1));
3881
3882 % [LO_D,HI_D,LO_R,HI_R] = WFILTERS('wname') computes four
3883 % filters associated with the orthogonal or biorthogonal
3884 % wavelet named in the string 'wname'.
3885 % LO_D, the decomposition low-pass filter
3886 % HI_D, the decomposition high-pass filter
3887 [LO_D,HI_D]=wfilters(str,'d'); % 'd': (Decomposition filters)
3888
3889 % Cridem la funció fWP_DecTree, passant-li com a arguments el
3890 % senyal, el nivell que hem escollit i Lo_D i Hi_D acabats de crear
3891 fileGlobal.BCoefitients=fWP_DecTree(...
3892     fileGlobal.xsignals,...
3893     fig.nivellDescomposatsJ,...
3894     LO_D,HI_D);
3895
3896 % Netegem les variables que ja no necessitarem
3897 clear LO_D HI_D val str
3898

```

```

3899 guidata(gcbo,fig) % Desem els canvis fets
3900 EnableDisable(gcbo,'on',0)
3901 fig=guidata(gcbo); % llegeix guidata.
3902
3903 set(fig.buttonProcess,'enable','off'); % Ens assegurem que el botó de
3904 % processar del panel 'Feature Extraxtion' és independent d'aquest.
3905
3906 drawnow % Regenera la figura i fa més sensible la GUI.
3907
3908 set([fig.textCoefficients fig.varCoefficients],'enable','on');
3909 set(fig.varCoefficients, ...
3910     'String', ... % num_mostres * (nivells + node(0,0))
3911     num2str(size(fileGlobal.xsignals,2)*(fileGlobal.sJ+1)));
3912
3913 guidata(gcbo,fig) % Desem els canvis fets
3914 FetElProcess(fig)
3915 fig=guidata(gcbo); % llegeix guidata.
3916
3917 ButtonRemove();
3918
3919 set(fig.buttonReconsCustDict,'Enable','off');
3920
3921 set(fig.textWait,...
3922     'String','Please wait... ',...
3923     'Visible','off')
3924
3925 guidata(gcbo,fig) % Desem els canvis fets
3926 end
3927
3928
3929 function MenuTools (varargin)
3930 fig=guidata(gcbo); % llegeix guidata.
3931 global fileGlobal
3932 if isfield(fileGlobal,'xsignals')&&...
3933     isfield(fileGlobal,'ranswer')&&...
3934     isfield(fileGlobal,'sJ')
3935     set([fig.menuPlot ...
3936         fig.menuFeatureExtraction ...
3937         fig.menuDiscrimAnalysis],...
3938         'enable','on');
3939     % El Signal reconstruction no, perquè l'activarem més tard
3940 else
3941     set([fig.menuPlot ...
3942         fig.menuFeatureExtraction ...
3943         fig.menuDiscrimAnalysis],...
3944         'enable','off');
3945 end
3946
3947 guidata(gcbo,fig) % Desem els canvis fets
3948 end
3949
3950
3951 function MenuView (varargin)
3952 fig=guidata(gcbo); % llegeix guidata.
3953 global fileGlobal
3954 if isfield(fileGlobal,'xsignals')&&...

```

```

3955         isfield(fileGlobal, 'ranswer') && ...
3956         isfield(fileGlobal, 'sJ')
3957         set([fig.menuToolbar fig.menuPlot], 'enable', 'on');
3958     else
3959         set([fig.menuToolbar fig.menuPlot], 'enable', 'off');
3960     end
3961     guidata(gcbo, fig) % Desem els canvis fets
3962     end
3963
3964
3965     function MenuFile (varargin)
3966     fig=guidata(gcbo); % llegeix guidata.
3967     global fileGlobal
3968     variables = fieldnames(fileGlobal);
3969     if ~isempty(variables)
3970         set(fig.menuSaveSession, 'Enable', 'on'); % Inhabilitem el menú
3971         set(fig.menuExport, 'Enable', 'on'); % Inhabilitem el menú
3972         set(fig.menuSave, 'enable', 'on');
3973         set(fig.menuCloseSession, 'Enable', 'on'); % Inhabilitem el menú
3974     else
3975         set(fig.menuExport, 'Enable', 'off'); % Inhabilitem el menú
3976         set(fig.menuSave, 'enable', 'off');
3977         set(fig.menuCloseSession, 'Enable', 'off'); % Inhabilitem el menú
3978         set(fig.menuSaveSession, 'Enable', 'off'); % Inhabilitem el menú
3979     end
3980
3981     % up=strrep(userpath, ';', '');
3982     % cami=strcat(up, '/savedSession.mat');
3983     cami='savedSession.mat';
3984     if exist(cami, 'file')==2
3985         set(fig.menuLoadSession, 'Enable', 'on'); % Habilitem el menú
3986     else
3987         set(fig.menuLoadSession, 'Enable', 'off'); % Inabilitem el menú
3988     end
3989     clear variables cami
3990     guidata(gcbo, fig) % Desem els canvis fets
3991     end
3992
3993
3994     function PopupJ (varargin)
3995     fig=guidata(gcbo); % llegeix guidata.
3996     global fileGlobal
3997     j=get(fig.popupJ, 'Value');
3998     nmostres=(size(fileGlobal.BCoefitients,1));
3999
4000     k=1:2^(j-1);
4001
4002     l=1:nmostres/(k(end));
4003
4004     set(fig.popupK, ...
4005         'String', k-1, ... % Volem que mostri 0 com a senyal original
4006         'Value', 1, ... % 1 serà el senyal original
4007         'Enable', 'on');
4008     set(fig.popupL, ...
4009         'String', l-1, ... % Volem que mostri 0 com a senyal original
4010         'Value', 1, ... % 1 serà el senyal original

```

```

4011     'Enable','on');
4012
4013 clear j J
4014 guidata(gcbo,fig) % Desem els canvis fets
4015 end
4016
4017
4018 function ButtonAddToReconstr (varargin)
4019 % Adjunta un sol espai a la reconstrucció WPT
4020
4021 fig=guidata(gcbo); % llegeix guidata.
4022 global fileGlobal
4023
4024 j=get(fig.popupJ,'Value')-1; % L'string és value+1
4025 k=get(fig.popupK,'Value')-1;
4026 l=get(fig.popupL,'Value')-1;
4027
4028 if isfield(fileGlobal,'subspaceReconstr') && ...
4029     ~isempty(fileGlobal.subspaceReconstr)
4030     coincidencies=find( ... % Si no l'hi hem afegit abans
4031         fileGlobal.subspaceReconstr(:,1)==j & ...
4032         fileGlobal.subspaceReconstr(:,2)==k & ...
4033         fileGlobal.subspaceReconstr(:,3)==l, 1);
4034 else
4035     % Si no existeix l'inicialitzarem a []
4036     fileGlobal.subspaceReconstr=[];
4037     coincidencies=[];
4038 end
4039
4040 if isempty(coincidencies)
4041     jkl=strcat(num2str(j),',',num2str(k),',',num2str(l));
4042
4043     val=get(fig.listboxJKL,'Value');
4044     antic_str=get(fig.listboxJKL,'String');
4045
4046     celljkl={jkl};
4047
4048     str=vertcat(antic_str,celljkl);
4049
4050     set(fig.listboxJKL,'string',str,'Value',val,'enable','on');
4051
4052     set([fig.botoRemoveToReconstruct ...
4053         fig.botoRemoveAllToReconstruct fig.buttonReconsCustDict],...
4054         'Enable','on');
4055
4056     fileGlobal.subspaceReconstr(end+1,1)=j;
4057     fileGlobal.subspaceReconstr(end,2)=k;
4058     fileGlobal.subspaceReconstr(end,3)=l;
4059
4060
4061 end
4062 clear val *str *jkl i j k l coincidencies
4063 guidata(gcbo,fig) % Desem els canvis fets
4064 end
4065
4066

```



```

4067 function ButtonAddNodeToReconstr (varargin)
4068 % Adjunta tot el node a la reconstrucció WPT
4069
4070 fig=guidata(gcbo); % llegeix guidata.
4071 global fileGlobal
4072 EnableDisable(gcbo,'off',0)
4073 set([fig.botoRemoveToReconstruct ...
4074     fig.botoRemoveAllToReconstruct],...
4075     'Enable','off');
4076 drawnow
4077
4078 j=get(fig.popupJ,'Value')-1; % L'string és value+1
4079 k=get(fig.popupK,'Value')-1;
4080 L=cellstr(get(fig.popupL,'string'));
4081
4082 if ~isfield(fileGlobal,'subspaceReconstr')
4083     % Si no existeix l'inicialitzarem a []
4084     fileGlobal.subspaceReconstr=[];
4085 end
4086
4087 for n=1:str2double(char(L(end)))+1
4088     l=str2double(char(L(n)));
4089     % Comprovem si l'hi haviem afegit abans
4090     if isempty(fileGlobal.subspaceReconstr)
4091         coincidencies=[];
4092     else
4093         coincidencies=find( ...
4094             fileGlobal.subspaceReconstr(:,1)==j & ...
4095             fileGlobal.subspaceReconstr(:,2)==k & ...
4096             fileGlobal.subspaceReconstr(:,3)==l, 1);
4097     end
4098     if isempty(coincidencies)
4099         jkl=strcat(num2str(j),',',...
4100                 num2str(k),',',...
4101                 num2str(l));
4102
4103         val=get(fig.listboxJKL,'Value');
4104         antic_str=get(fig.listboxJKL,'String');
4105
4106         celljkl={jkl};
4107
4108         str=vertcat(antic_str,celljkl);
4109
4110         set(fig.listboxJKL,'string',str,'Value',val,'enable','on');
4111
4112         set([fig.botoRemoveToReconstruct ...
4113             fig.botoRemoveAllToReconstruct ...
4114             fig.buttonReconsCustDict],...
4115             'Enable','on');
4116
4117         fileGlobal.subspaceReconstr(end+1,1)=j;
4118         fileGlobal.subspaceReconstr(end,2)=k;
4119         fileGlobal.subspaceReconstr(end,3)=l;
4120     end
4121 end
4122 EnableDisable(gcbo,'on',0)

```

```

4123 set([fig.botoRemoveToReconstruct fig.botoRemoveAllToReconstruct],...
4124     'Enable','on');
4125 clear nmostres val *str *jkl i j k l coincidencies
4126 guidata(gcbo,fig) % Desem els canvis fets
4127 end
4128
4129
4130 function ButtonRemoveToReconstr (varargin)
4131 % Esborra un sol espai adjuntat per reconstruir WPT
4132
4133 fig=guidata(gcbo); % llegeix guidata.
4134 global fileGlobal
4135 % listBoxJKL
4136 valToRemove=get(fig.listBoxJKL,'value');
4137 strToRemove=get(fig.listBoxJKL,'string');
4138 if ~isempty(strToRemove)
4139     str=strToRemove(valToRemove);
4140     [j,remain]=strtok(str,',');
4141     [k,remain]=strtok(remain,',');
4142     [l]=strtok(remain,',');
4143     j=str2double(char(j));
4144     k=str2double(char(k));
4145     l=str2double(char(l));
4146     fila=find( ... % Troba la fila on són
4147         fileGlobal.subspaceReconstr(:,1)==j & ...
4148         fileGlobal.subspaceReconstr(:,2)==k & ...
4149         fileGlobal.subspaceReconstr(:,3)==l, 1);
4150
4151     % L'esborrem de fileGlobal.subspaceReconstr
4152     fileGlobal.subspaceReconstr(fila,:)=[];
4153
4154     % L'esborrem de fig.listBoxJKL
4155     strToRemove(valToRemove)=[];
4156     set(fig.listBoxJKL,'string',strToRemove,'Value',1);
4157     if isempty(strToRemove)
4158         set([fig.botoRemoveToReconstruct ...
4159             fig.botoRemoveAllToReconstruct fig.listBoxJKL...
4160             fig.buttonReconsCustDict],'enable','off');
4161     end
4162 end
4163 clear str j k l remain fila *ToRemove
4164 guidata(gcbo,fig) % Desem els canvis fets
4165 end
4166
4167
4168 function ButtonRemoveAllToReconstr (varargin)
4169 % Esborra tots els espais adjuntats per reconstruir
4170
4171 fig=guidata(gcbo); % llegeix guidata.
4172 global fileGlobal
4173
4174 if isfield(fileGlobal,'subspaceReconstr')
4175     fileGlobal.subspaceReconstr=[];
4176 end
4177 if isfield(fileGlobal,'selected_Coefficients')
4178     fileGlobal.selected_Coefficients=[];

```

```

4179 end
4180 set(fig.listBoxJKL,'value',1);
4181 set(fig.listBoxJKL,'string','');
4182
4183 set([fig.listBoxJKL fig.botoRemoveToReconstruct ...
4184      fig.botoRemoveAllToReconstruct fig.buttonReconsCustDict],...
4185      'Enable','off');
4186
4187 guidata(gcbo,fig) % Desem els canvis fets
4188 end
4189
4190
4191 function ButtonReconsCustDict (varargin)
4192 fig=guidata(gcbo); % llegeix guidata.
4193 global fileGlobal fileNameGlob ContourOrProfile
4194
4195 EnableDisable(gcbo,'off',0)
4196 set([fig.listBoxJKL fig.botoRemoveToReconstruct ...
4197      fig.botoRemoveAllToReconstruct],'Enable','off');
4198 set(fig.textWait,...
4199      'String','Reconstructing... ',...
4200      'Enable','on',...
4201      'Visible','on');
4202 drawnow
4203
4204 fileGlobal.selected_Coefficients=fgetCoefitient(...
4205      fileGlobal.BCoefitients,fileGlobal.subspaceReonstr);
4206
4207 val=get(fig.listBoxFEWPT,'Value'); %Obtenim el nom de la llibreria
4208 str=char(fig.Complete_dLibrary(val,1));
4209
4210 % [LO_D,HI_D,LO_R,HI_R] = WFILTERS('wname') computes four
4211 % filters associated with the orthogonal or biorthogonal
4212 % wavelet named in the string 'wname'.
4213 % LO_R, the reconstruction low-pass filter
4214 % HI_R, the reconstruction high-pass filter
4215 [LO_R,HI_R]=wfilters(str,'r'); % 'r': (Reconstruction filters)
4216 clear str val
4217
4218 fileGlobal.xAux_Signal=fIWPT_RecTRee(...
4219      fileGlobal.selected_Coefficients,...
4220      fileGlobal.subspaceReonstr,...
4221      size(fileGlobal.BCoefitients,1),...
4222      LO_R,HI_R);
4223
4224 if isfield(fileGlobal,'xsignals') && ...
4225      isfield(fileGlobal,'xAux_Signal') && ...
4226      isfield(fileGlobal,'ranswer')
4227 set(fig.textWait,...
4228      'String','Drawing..... ',...
4229      'Visible','on');
4230
4231 drawnow
4232
4233 AxesResetIInvisible();
4234

```

```

4235     zoom off
4236     set(fig.panelTaula, 'Visible', 'off')
4237
4238     if strcmp(ContourOrProfile, 'Profile')
4239         FuncPlot(fig.axesSignals_VSxAuxProfileRec, ...
4240             'Original signal', ...
4241             fileGlobal.xsignals, fileGlobal.ranswer)
4242         FuncPlot(fig.axesxAux_VSsignalsProfile, ...
4243             'Reconstruction', ...
4244             fileGlobal.xAux_Signal, fileGlobal.ranswer)
4245     else
4246         % Farà que es vegi correctament el caràcter _:
4247         str = strrep(fileNameGlob, '_', '\_');
4248         str=['(' str ')']; % Hi afegim 'Totes les parts'
4249         str=['Contour view ' str];
4250
4251         FuncPolar(fig.axesPolarTots, str, ...
4252             fileGlobal.xsignals, fileGlobal.ranswer)
4253
4254         %-----%
4255         % I ara que dibuixi les dues figures de sota
4256         FuncPlot(fig.axesSignals_VSxAuxRec, 'Original signal', ...
4257             fileGlobal.xsignals, fileGlobal.ranswer)
4258         FuncPlot(fig.axesxAux_VSsignals, 'Reconstruction', ...
4259             fileGlobal.xAux_Signal, fileGlobal.ranswer)
4260
4261         clear linia angle fi i j str
4262     end
4263     set(fig.textWait, ...
4264         'String', 'Please wait...', ...
4265         'Visible', 'off');
4266 end
4267 set([fig.listBoxJKL ...
4268     fig.botoRemoveToReconstruct
4269     fig.botoRemoveAllToReconstruct], 'Enable', 'on');
4270 EnableDisable(gcbo, 'on', 0)
4271 set(fig.textWait, ...
4272     'String', 'Please wait...', ...
4273     'Visible', 'off')
4274
4275 guidata(gcbo, fig) % Desem els canvis fets
4276 end
4277
4278
4279 % -----Funcions d'exportar i importar
4280 function MenuLoad (varargin)
4281 fig=guidata(gcbo); % llegeix l'estructura de la figura anterior
4282
4283 global loadOk
4284
4285 % Variables que enviarem a la funció ExportImport
4286 nomFigura='Import'; % Nom que tindrà la figura
4287 strRadioExportMAT='Load from MAT file'; % String del radioExportMAT
4288 strRadioExportWorkspace='Load from Workspace'; %El del radioWorkspace
4289 positionCancel=2/3; % Factor de multiplicació de la mida dels botons
4290 lbEntrada='off'; % Enables dels listboxes
4291 lbSortida='off';

```

```

4292 bAfegTot='off';           % Enable dels botons Afegeix i Tots
4293 bTreuCap='off';          % Enable dels botons Treu i Cap
4294 loadOk=0;
4295 ExportImport (nomFigura, ...
4296             strRadioExportMAT, ...
4297             strRadioExportWorkspace, ...
4298             positionCancel, lbEntrada, lbSortida, bAfegTot, bTreuCap);
4299
4300 uiwait(gcf)
4301 if loadOk==1
4302     % -----%
4303     EnableDisable(gcbo, 'off', 1) % Posa a zero la interfície
4304     % Modificació de la interfície inicial
4305     % Cridem la funció Modif_interf_inicial:
4306     fig=Modif_interf_inicial(gcbo, 0, fig); % 0: no és nova sessió
4307 end
4308
4309 clear global loadOk
4310 clear nomFigura strRadio* positionCancel lb* b*
4311 guidata(gcbo, fig)
4312 % -----%
4313
4314 end
4315
4316
4317 function MenuExport (varargin)
4318
4319 global fileNameGlob
4320
4321 % Variables que enviarem a la funció ExportImport
4322 nomFigura=['Export - ' fileNameGlob];
4323 strRadioExportMAT='Save to MAT file'; % String del radioExportMAT
4324 strRadioExportWorkspace='Save to Workspace'; %String del radioWorkspace
4325 positionCancel=1; % Factor de multiplicació de la mida dels botons
4326 lbEntrada='on'; % Enable dels listboxes
4327 lbSortida='off';
4328 bAfegTot='on'; % Enable dels botons Afegeix i Tots
4329 bTreuCap='off'; % Enable dels botons Treu i Cap
4330
4331 ExportImport (nomFigura, strRadioExportMAT, ...
4332             strRadioExportWorkspace, ...
4333             positionCancel, lbEntrada, lbSortida, bAfegTot, bTreuCap);
4334
4335 uiwait(gcf)
4336
4337 clear nomFigura strRadio* positionCancel lb* b*
4338
4339 end
4340
4341 % Nova finestra comuna per importar i exportar
4342 function ExportImport (nomFigura, ...
4343             strRadioExportMAT, ...
4344             strRadioExportWorkspace, ...
4345             positionCancel, ...
4346             lbEntrada, lbSortida, bAfegTot, bTreuCap, ...
4347             varargin)
4348 fig=guidata(gcf); % llegeix l'estructura de la figura principal

```

```

4349
4350 menu=gcb0; % Element que ha fet la crida
4351 valorLoad=fig.menuLoad;
4352 valorExport=fig.menuExport;
4353
4354 global fileGlobal
4355
4356 set(0,'Units','pixels') % ens assegurem que està en pixels
4357
4358 exp=struct; % Inicialitzem l'estructura
4359 finestra = [25+fig.bdwidth,           ... %[left     ...
4360            fig.scnsize(4)*2/3-50,     ... % bottom  ...
4361            fig.scnsize(3)*4/10,       ... % width   ...
4362            fig.scnsize(4)/3];         % height ]
4363 exp.exportFIG = figure(...
4364     'MenuBar','none',...             % amaga el menú estàndard
4365     'Name',nomFigura,...             % el nom de la figura serà Export
4366     'NumberTitle','off',...         % treu el número de figura (figure: 1)
4367     'Position',finestra,...         % la posiciona en el lloc calculat abans
4368     'Resize','off',...             % no ens deixa canviar-li el tamany
4369     'Units','Pixels',...           % la mesurarà en píxels
4370     'WindowStyle','modal',...      % Finestra modal: sempre visible
4371     'Visible','on',...             % La farem visible més tard
4372     'HandleVisibility','callback',...
4373     'CloseRequestFcn',@Export_CloseRequestFcn,...
4374     'Color',get(0,'defaultUicontrolBackgroundColor'));
4375
4376 exp.uipanelVarsContenedor=uipanel(exp.exportFIG,...
4377     'Title','Choose variables',...
4378     'Tag','uipanelVarsContenedor',...
4379     'Units','Pixels',...
4380     'TitlePosition','centertop ',...
4381     'Position',[0 ...
4382                 finestra(4)-finestra(4)*3/4 ...
4383                 finestra(3) ...
4384                 finestra(4)*3/4],...
4385     'BorderStyle','beveledout',...
4386     'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
4387 p=get(exp.uipanelVarsContenedor,'Position');
4388
4389 m=5;
4390
4391 % Obtenim les mides i posició de Listboxes
4392 p=[p(3)/2+3*m/2 ...
4393    5 ...
4394    p(3)/2-3*m ...
4395    p(4)-3*m];
4396
4397 mee=3; % Marge entre elements interns: ListBoxes-mee-botons
4398 mde=6; % Marge dret, esquerre, dalt i baix entre listboxes i panel.
4399
4400 % Fem una petita equació per trobar l'amplada dels ListBoxes:
4401 % midaPanelAmplada=mde+midaListBoxes+mee+mbp+mee+midaListBoxes+mde
4402 % midaPanelAmplada=2*mde+2*midaListBoxes+2*mee+mbp (...)
4403 % midaListBoxes=(midaPanelAmplada-mbp-2*(mde+mee))/2

```

```

4404 ampLB=(p(3)-2*mde); % Amplada de ListBoxes
4405
4406 abT=14; % Alçada dels botons botoTots i botoCap
4407 alcLB=(p(4)-2*mde-abT-mee*4); % Alçada dels Listboxs
4408 pos=[mde (mde+abT+mee) ampLB alcLB];
4409 exp.listBoxEntrada=icontrol(exp.uipanelVarsContenedor,...
4410     'Style','listbox',...
4411     'Tag','listboxEntrada',...
4412     'Position',pos,...
4413     'enable',lbEntrada,...
4414     'BackgroundColor','w');
4415
4416
4417 exp.textVarsEntrada=icontrol(exp.uipanelVarsContenedor,...
4418     'String','Input variables',...
4419     'Style','text',...
4420     'Tag','textVarsEntrada',...
4421     'Units','Pixels',...
4422     'Position',[pos(1) ...
4423                 pos(2)+pos(4)+mee ...
4424                 pos(3) ...
4425                 15],...
4426     'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
4427
4428
4429 posb=[mde mde ampLB abT];
4430 exp.botoTots=icontrol(exp.uipanelVarsContenedor,...
4431     'Style','pushbutton',...
4432     'String','All',...
4433     'Tag','botoTots',...
4434     'Enable',bAfegeTot,...
4435     'Position',posb,...
4436     'Callback',@BotoTots);
4437
4438 mee=3; % Marge entre elements interns: panels petits-mee-botons
4439
4440 % Obtenim les mides i posició de uipanelListboxes
4441 pg=get(exp.uipanelVarsContenedor,'Position');
4442
4443 mbp=(pg(3)/2-m-2*p(3)/2-mee)*3; % Amplada del botó Afegeix i Treu
4444 abp=(p(4)/2-m/2)*2/3; % Alçada
4445
4446 exp.botoAfegeix=icontrol(exp.uipanelVarsContenedor,...
4447     'Style','pushbutton',...
4448     'String','>',...
4449     'Tag','botoAfegeix',...
4450     'FontWeight','bold',...
4451     'Enable',bAfegeTot,...
4452     'Position',[(pg(3)/2-mbp/2) ...
4453                 (p(4)/2+m/2) ...
4454                 mbp ...
4455                 abp ...
4456     'Callback',@BotoAfegeix);
4457 exp.botoTreu=icontrol(exp.uipanelVarsContenedor,...

```

```

4458     'Style', 'pushbutton', ...
4459     'String', '<', ...
4460     'Tag', 'botoTreu', ...
4461     'FontWeight', 'bold', ...
4462     'Position', [ (pg (3) /2-mbp/2)           ...
4463                 (p (4) /2-m/2-abp)           ...
4464                 mbp                           ...
4465                 abp                           ], ...
4466     'Enable', bTreuCap, ...
4467     'Callback', @BotoTreu);
4468
4469 exp.listBoxSortida=icontrol (exp.uipanelVarsContenedor, ...
4470     'Style', 'listbox', ...
4471     'Tag', 'listboxSortida', ...
4472     'Position', [ (pg (3) /2)+mbp/2+3*mee ...
4473                 pos (2) ...
4474                 pos (3) ...
4475                 pos (4) ], ...
4476     'enable', lbSortida, ...
4477     'BackgroundColor', 'w');
4478
4479 exp.textVarsSortida=icontrol (exp.uipanelVarsContenedor, ...
4480     'String', 'Output variables', ...
4481     'Style', 'text', ...
4482     'Tag', 'textVarsSortida', ...
4483     'Units', 'Pixels', ...
4484     'Position', [ (pg (3) /2)+mbp/2+3*mee ...
4485                 pos (2)+pos (4) +mee ...
4486                 pos (3) ...
4487                 15], ...
4488     'BackgroundColor', get (0, 'defaultUicontrolBackgroundColor'));
4489
4490 exp.botoCap=icontrol (exp.uipanelVarsContenedor, ...
4491     'Style', 'pushbutton', ...
4492     'String', 'None', ...
4493     'Tag', 'botoCap', ...
4494     'Enable', bTreuCap, ...
4495     'Position', [ (pg (3) /2)+mbp/2+3*mee ...
4496                 posb (2) ...
4497                 posb (3) ...
4498                 posb (4) ], ...
4499     'Callback', @BotoCap);
4500
4501 % Netegem les variables que ja no necessitarem
4502 clear mee mde mbp ampLB abT alcLB
4503
4504 p1=get (exp.uipanelVarsContenedor, 'Position');
4505 alc=finestra (4) -p1 (4);
4506 p=[0 0 finestra (3) alc];
4507 exp.uipanelRadiosContenedor=uipanel (exp.exportFIG, ...
4508     'Tag', 'uipanelRadiosContenedor', ...
4509     'Units', 'Pixels', ...
4510     'Position', p, ...

```



```

4511     'BorderStyle', 'beveledout', ...
4512     'BackgroundColor', get(0, 'defaultUicontrolBackgroundColor'));
4513
4514     m=10;
4515     mrE=130;
4516     mr=(p(3)/2-mrE)/2;
4517     exp.radioExportMAT=uicontrol(exp.uipanelRadiosContenedor, ...
4518     'Style', 'radiobutton', ...
4519     'String', strRadioExportMAT, ...
4520     'Enable', 'on', ...
4521     'Tag', 'radioExportMAT', ...
4522     'Value', 1, ...
4523     'Position', [p(3)/2-mrE-mr    ...
4524                 p(4)/2+m/2      ...
4525                 mrE              ...
4526                 15                ], ...
4527     'Callback', @RadioExportMAT);
4528     exp.radioExportWorkspace=uicontrol(exp.uipanelRadiosContenedor, ...
4529     'Style', 'radiobutton', ...
4530     'String', strRadioExportWorkspace, ...
4531     'Enable', 'on', ...
4532     'Value', 0, ...
4533     'Tag', 'radioExportWorkspace', ...
4534     'Position', [p(3)/2-mrE-mr    ...
4535                 p(4)/2-m/2-15    ...
4536                 mrE              ...
4537                 15                ], ...
4538     'Callback', @RadioExportWorkspace);
4539
4540     mbE=30;
4541     ampB=(p(3)/4-2*m)*positionCancel;
4542
4543     exp.botoCancel=uicontrol(exp.uipanelRadiosContenedor, ...
4544     'Style', 'pushbutton', ...
4545     'String', 'Cancel', ...
4546     'Enable', 'on', ...
4547     'Tag', 'botoCancel', ...
4548     'Position', [p(3)-m-ampB    ...
4549                 p(4)/2-mbE/2  ...
4550                 ampB          ...
4551                 mbE            ], ...
4552     'Callback', @BotoCancelExport);
4553
4554     % Botons exclusius de cada menú que fa la crida:
4555     switch menu
4556     case valorLoad
4557         p=get(exp.uipanelRadiosContenedor, 'Position');
4558         a=get(exp.botoCancel, 'Position');
4559
4560         exp.botoImport=uicontrol(exp.uipanelRadiosContenedor, ...
4561         'Style', 'pushbutton', ...
4562         'String', 'Import', ...
4563         'Enable', 'on', ...

```

```

4564         'Tag', 'botoImport', ...
4565         'Position', [p(3)-10-a(3)-10-a(3)-10-a(3) ...
4566                 p(4)/2-a(4)/2 ...
4567                 a(3) ...
4568                 a(4) ... ], ...
4569         'Callback', @BotoImport);
4570
4571     % El botó load s'anomena botoExport als handles, ja que així
4572     % no hem de fer noves condicions en els següents Callbacks
4573     % (botoAfegeix, botoTreu, botoTots i botoCap), però el seu
4574     % Callback és BotoLoad:
4575     exp.botoExport=uicontrol(exp.uipanelRadiosContenedor, ...
4576         'Style', 'pushbutton', ...
4577         'String', 'Load', ...
4578         'Enable', 'off', ...
4579         'Tag', 'botoExport', ...
4580         'Position', [p(3)-10-a(3)-10-a(3) ...
4581                 p(4)/2-a(4)/2 ...
4582                 a(3) ...
4583                 a(4) ... ], ...
4584         'Callback', {@BotoLoad, fig});
4585     case valorExport
4586         % Botó exclusiu:
4587         p=get(exp.uipanelRadiosContenedor, 'Position');
4588         a=get(exp.botoCancel, 'Position');
4589
4590         exp.botoExport=uicontrol(exp.uipanelRadiosContenedor, ...
4591             'Style', 'pushbutton', ...
4592             'String', 'Export', ...
4593             'Enable', 'off', ...
4594             'Tag', 'botoExport', ...
4595             'Position', [p(3)*3/4-a(3)-10 ...
4596                     p(4)/2-a(4)/2 ...
4597                     a(3) ...
4598                     a(4) ... ], ...
4599             'Callback', @BotoExport);
4600
4601         % Nom de les variables que tenim:
4602         exp.variables = fieldnames(fileGlobal);
4603         % Inicialització de les posicions dels listboxes:
4604         exp.ultimaPosicioEsquerra=length(exp.variables);
4605         exp.ultimaPosicioDreta=0;
4606         exp.esquerra=exp.variables;
4607         exp.dreta={};
4608
4609         set(exp.listBoxEntrada, 'String', exp.esquerra);
4610     end
4611
4612     guidata(gcf, exp) % Desem els canvis fets
4613
4614 end
4615
4616
4617 function BotoImport (varargin)
4618     % Importa el nom de les variables que es volen carregar

```

```

4619
4620 % Retorna el directori del fitxer a path i el nom del fitxer a file:
4621
4622 exp=guidata(gcbo); % llegeix guidata.
4623 switch get(exp.radioExportMAT,'Value')
4624     case 1 % Importa des de fitxer .MAT
4625         ind=0; % variable que usarem per la detecció d'errors
4626         ok=1; % variable que usarem per saber si hem fet 'Cancel·lar'
4627         % Si fem 'Cancel·lar ok=0;
4628
4629         while ind==0 && ok~=0 % Si les extensions no són correctes
4630             % i no hem fet cancel·lar
4631             [exp.fileName, path, ok] = uigetfile(...
4632                 {'*.mat','MAT-files (*.mat)';... % 1
4633                 '*.txt','TXT-files (*.txt)'};... % i 2 ok possibles
4634                 'Open file');
4635
4636             % Detecció d'errors sobre *.* (no totes les versions de
4637             % Matlab en tenen)
4638
4639             % Busquem la cadena '.mat' al nom del fitxer
4640             mat=findstr('.mat',exp.fileName);
4641
4642             if isempty(mat)~=1
4643                 % Si no té la cadena .mat no tindrà l'extensió MAT
4644
4645                 % Ara mirarem si en 3 posicions més el nom del fitxer
4646                 % s'acaba
4647                 if mat(end)+3==length(exp.fileName) % Si és així
4648                     % l'extensió és correcta
4649                     ind=1; % Posarem la variable 'ind' a 1 perquè no
4650                     % torni a entrar al while
4651                 end
4652             end
4653
4654             % Busquem la cadena '.txt' al nom del fitxer
4655             txt=findstr('.txt',exp.fileName);
4656
4657             if isempty(txt)~=1
4658                 % Si no té la cadena .txt no tindrà l'extensió TXT
4659
4660                 %Ara mirarem si en 3 posicions més el nom del fitxer
4661                 % s'acaba
4662                 if txt(end)+3==length(exp.fileName) % Si és així
4663                     % l'extensió és correcta
4664                     ind=1; % Posarem la variable 'ind' a 1 perquè no
4665                     % torni a entrar al while
4666                 end
4667             end
4668         end % end while
4669     if ind==1 && ok<=2 && ok~=0
4670         % Si la càrrega ha estat correcta en extensió
4671         % i no hem fet 'Cancel·lar'...
4672         % Escriu tota la ruta del fitxer amb el nom inclòs:
4673         exp.filePathName=strcat(path,exp.fileName);
4674
4675

```

```

4676         % Nom de les variables que tenim:
4677         exp.variables = who('-file',exp.filePathName);
4678
4679         % Inicialitzacio de les posicions dels listboxes:
4680         exp.ultimaposicioEsquerra=length(exp.variables);
4681         exp.ultimaposicioDreta=0;
4682         exp.esquerra=exp.variables;
4683         exp.dreta={};
4684         if ~isempty(exp.esquerra)
4685             set(exp.listBoxEntrada,...
4686                 'String',exp.esquerra,...
4687                 'enable','on',...
4688                 'Value',1);
4689             set(exp.listBoxSortida,...
4690                 'Enable','off',...
4691                 'string','',...
4692                 'Value',1);
4693             set([exp.botoAfegeix exp.botoTots],'Enable','on');
4694             set([exp.botoTreu exp.botoCap],'Enable','off');
4695
4696         end
4697         % Netegem les variables que ja no necessitem
4698         clear ind ok txt mat path fileName;
4699
4700     else return
4701     end
4702     otherwise % Importa des de Workspace
4703         exp.variables=evalin('base','who');
4704         % Inicialitzacio de les posicions dels listboxes:
4705         exp.ultimaposicioEsquerra=length(exp.variables);
4706         exp.ultimaposicioDreta=0;
4707         exp.esquerra=exp.variables;
4708         exp.dreta={};
4709
4710         if ~isempty(exp.esquerra)
4711             set(exp.listBoxEntrada,...
4712                 'String',exp.esquerra,...
4713                 'enable','on');
4714             set(exp.listBoxSortida,'Enable','off','string','');
4715             set([exp.botoAfegeix exp.botoTots],'Enable','on');
4716             set([exp.botoTreu exp.botoCap],'Enable','off');
4717         end
4718     end
4719     guidata(gcbo,exp); % Desa les variables de l'estructura fig
4720 end
4721
4722
4723
4724 function BotoLoad (varargin)
4725 % Carrega les variables importades
4726
4727 exp=guidata(gcbo);
4728
4729 global fileGlobal fileNameGlob filePathNameGlob loadOk
4730
4731 switch get(exp.radioExportMAT,'Value')
4732     case 1 % Carreguem un fitxer
4733

```

```

4734     % Escribe tota la ruta del fitxer amb el nom inclòs a la variable
4735     % global:
4736     filePathNameGlob=exp.filePathName;
4737     fileNameGlob=exp.fileName;
4738     % Ho carrega a la variable 'fileGlobal'
4739     fileGlobal=load(filePathNameGlob, exp.dreta{:});
4740     loadOk=1; % Càrrega satisfactòria
4741
4742     otherwise % Carreguem variables del Workspace
4743         tmp=0;
4744         for i=1:size(exp.dreta,1)
4745             fileGlobal.(char(exp.dreta(i,1)))=...
4746                 evalin('base', (char(exp.dreta(i,1))));
4747             tmp=tmp+1;
4748         end
4749
4750         switch tmp
4751             case 0
4752                 loadOk=0; % Càrrega no satisfactòria
4753                 return
4754             otherwise
4755                 loadOk=1; % Càrrega satisfactòria
4756                 fileNameGlob='Workspace';
4757         end
4758     end
4759
4760     uiresume(exp.exportFIG);
4761     delete(exp.exportFIG); % Tanca la figura
4762     end
4763
4764
4765     function RadioExportMAT (varargin)
4766     exp=guidata(gcbo);
4767
4768     set(exp.radioExportWorkspace, 'Value', 0)
4769     set(exp.radioExportMAT, 'Value', 1)
4770
4771     guidata(gcf, exp) % Desem els canvis fets
4772     end
4773
4774
4775     function RadioExportWorkspace (varargin)
4776     exp=guidata(gcbo);
4777
4778     set(exp.radioExportMAT, 'Value', 0)
4779     set(exp.radioExportWorkspace, 'Value', 1)
4780
4781     guidata(gcf, exp) % Desem els canvis fets
4782     end
4783
4784
4785     function BotoExport (varargin)
4786     % Crida les funcions corresponents al tipus l'export que vulguem fer
4787
4788     exp=guidata(gcbo);
4789
4790     switch get(exp.radioExportMAT, 'Value')
4791         case 1 % exportem a MAT
4792             filterIndex=SaveToMATFile();
4793             switch filterIndex

```

```

4794         case 0
4795             return
4796         otherwise
4797             uiresume(exp.exportFIG);
4798             delete(exp.exportFIG); % Tanca la figura
4799         end
4800     otherwise % Exportem a Workspace
4801         SaveToWorkspace();
4802         uiresume(exp.exportFIG);
4803         delete(exp.exportFIG); % Tanca la figura
4804 end
4805 end
4806
4807
4808 function SaveToWorkspace (varargin)
4809 % Desa les variables seleccionades al Workspace
4810
4811 exp=guidata(gcbo); % llegeix guidata.
4812 global fileGlobal fileNameGlob
4813 tmp=0;
4814 % Podem gravar l'estructura fileGlobal en estructura o camp per camp.
4815 % Ho farem camp per camp.
4816
4817 set(exp.botoExport,'enable','off');
4818
4819 for i=1:size(exp.dreta,1)
4820     assignin('base',(char(exp.dreta(i,1))),...
4821             fileGlobal.(char(exp.dreta(i,1))))
4822     tmp=tmp+1;
4823 end
4824
4825 if tmp~=0
4826     assignin('base','FromFileName',fileNameGlob)
4827 end
4828
4829 set(exp.botoExport,'enable','on');
4830
4831 clear tmp
4832 end
4833
4834
4835 function filterIndex=SaveToMATFile (varargin)
4836 % Desa les variables seleccionades a un fitxer
4837
4838 exp=guidata(gcbo); % llegeix guidata.
4839
4840 global fileGlobal filePathNameGlob
4841 % Desarem l'estructura fileGlobal com a variables separades dins d'un
4842 % mateix fitxer MAT
4843
4844 [fileName,pathName,filterIndex] = uiputfile(...
4845     {'*.mat','MAT-files (*.mat)'},... % 1 ok possibles
4846     'Save file',...
4847     filePathNameGlob);
4848
4849 % Si es prem Cancel, es tanca la finestra, o el fitxer no existeix,
4850 % FileName=PathName=FilterIndex = 0:
4851 if filterIndex~=0
4852     pathNamecat=strcat(pathName,fileName);

```

```

4854
4855     % Ja podem desar les variables.
4856     % Podem gravar l'estructura fileGlobal directament o variable per
4857     % variable. Ho farem variable per variable.
4858     for i = 1:length(exp.dreta)
4859         eval([exp.dreta{i} '= fileGlobal.' exp.dreta{i} ';']);
4860     end
4861     save(pathNamecat, exp.dreta{:});
4862
4863     for i=1:length(exp.dreta)
4864         eval(['clear ' exp.dreta{i,1}]);
4865     end
4866 else
4867     return
4868 end
4869
4870 % Netegem les variables que ja no necessitem
4871 clear pathName fileName pathNamecat i
4872
4873 guidata(gcbo,exp) % Desem els canvis fets
4874 end
4875
4876
4877 function Export_CloseRequestFcn (varargin)
4878 exp=guidata(gcbo);
4879
4880 uiresume(exp.exportFIG);
4881
4882 delete(exp.exportFIG); % Tanca la figura
4883
4884 end
4885
4886
4887 function BotoCancelExport (varargin)
4888
4889 global loadOk
4890
4891 loadOk=0;
4892 Export_CloseRequestFcn();
4893
4894 end
4895
4896 % -----
4897 %   Funcions d'utilitat per LDB_GUI
4898 % -----
4899
4900 % ----Funció per modificar la interfície quan es carreguen les variables
4901 function fig=Modif_interf_inicial (gcbo,varargin)
4902 fig=guidata(gcbo); % llegeix guidata.
4903 global fileGlobal fileNameGlob ContourOrProfile
4904 if nargin==2
4905     novasessio=varargin{1};
4906 elseif nargin==3
4907     fig=varargin{2};
4908     novasessio=0;
4909 end
4910 % Modifiquem la interfície
4911 set([fig.textSignalName fig.varSignalName],'enable','on')
4912
4913 % Posem el nom del senyal:

```

```

4914 set(fig.varSignalName, 'String', num2str(fileNameGlob));
4915
4916 % Canviem la variable xdata a xsignals, si existeix:
4917 % F = ISFIELD(S,'camp') torna true si 'camp' és el nom d'un camp en
4918 % l'estructura S:
4919 if isfield(fileGlobal,'xdata') % xdata és el mateix que xsignals
4920     fileGlobal.xsignals=fileGlobal.xdata; % Guardem xdata a xsignals
4921     fileGlobal = rmfield(fileGlobal, 'xdata'); % Esborrem xdata
4922 end
4923 if isfield(fileGlobal,'xsignals')
4924     set([fig.textNumberSignals fig.varNumberSignals],'enable','on');
4925     set(fig.varNumberSignals, ...
4926         'String', num2str(size(fileGlobal.xsignals,2)));
4927 end
4928 % Si no existeix sJ la creem:
4929 if ~isfield(fileGlobal,'sJ')
4930     if isfield(fileGlobal,'xsignals')
4931         fileGlobal.sJ=log2(size(fileGlobal.xsignals,1));
4932     end
4933 end
4934 if isfield(fileGlobal,'sJ') % i posem a enable les caixes de text
4935     % corresponents
4936     set([fig.textLevels fig.varLevels],'enable','on');
4937     set(fig.varLevels, 'String', num2str(fileGlobal.sJ));
4938 end
4939 if isfield(fileGlobal,'BCoefitients') && isfield(fileGlobal,'sJ')
4940     set([fig.textCoefficients fig.varCoefficients],'Enable','on');
4941     set(fig.varCoefficients, ...
4942         'String', ... % num_mostres * (nivells + node(0,0))
4943         num2str(size(fileGlobal.BCoefitients,2)*...
4944         (log2(size(fileGlobal.BCoefitients,1))+1)));
4945 end
4946 % Canviem la variable rlabel a ranswer, si existeix:
4947 if isfield(fileGlobal,'rlabel') % rlabel és el mateix que ranswer
4948     fileGlobal.ranswer=fileGlobal.rlabel; % Guardem rlabel a ranswer
4949     fileGlobal = rmfield(fileGlobal, 'rlabel'); % Esborrem rlabel
4950 end
4951 if isfield(fileGlobal,'ranswer')
4952     set(fig.textVectorClass,'enable','on');
4953     set(fig.varVectorClass, ...
4954         'String',fileGlobal.ranswer(end),...
4955         'enable','on');
4956 end
4957 if isfield(fileGlobal,'ranswer') ...
4958     && isfield(fileGlobal,'xsignals') ...
4959     && isfield(fileGlobal,'sJ') && novasessio~=1
4960     set([fig.textVectorClass fig.varVectorClass],'enable','on');
4961
4962 % 'i' serà un vector que valdrà 1,2,3... fins al valor final
4963 % de ranswer
4964 i=1:fileGlobal.ranswer(end);
4965 set(fig.varVectorClass, 'String', num2str(i(end)));
4966
4967 set([fig.textGroupNumber fig.popupGroupNumber],'enable','on');
4968
4969 set(fig.popupGroupNumber, 'String', i);
4970
4971 set([fig.textFirst fig.editFirst ...
4972     fig.textLast fig.editLast], ...
4973     'enable','on');
4974

```



```

4975     i=get(fig.popupGroupNumber,'Value');
4976     ipos = find(fileGlobal.ranswer==i);
4977
4978
4979     % Posa el primer element de ipos a editFirst:
4980     set(fig.editFirst,'String',num2str(ipos(1)));
4981     % Posa l'últim element de ipos a editLast:
4982     set(fig.editLast,'String',num2str(ipos(end)));
4983
4984     set(fig.buttonView,'Enable','on');
4985
4986     set(fig.menuPolar,'Checked','off');
4987     ContourOrProfile='Profile';
4988
4989     set(fig.textLibType,'enable','on');
4990     % Per defecte, el RadioStd:
4991     set(fig.radioStd,'enable','on','Value',1);
4992
4993     %Panel Discriminant
4994     %Modifiquem el valor màxim de l'slider del nivell de descomposició
4995     % el valor que mostrarà i el pas ( en percentatge)
4996     set(fig.sliderNivellDescomp,...
4997         'Max',fileGlobal.sJ,...
4998         'Value',fileGlobal.sJ,...
4999         'SliderStep',[1/(fileGlobal.sJ-1) 1/(fileGlobal.sJ-1)]);
5000     set(fig.entraNivellDescomp,...
5001         'String',fileGlobal.sJ);
5002
5003     % Cridem la funció RadioStd per ensenyar directament els
5004     % seus listboxs i el seu contingut correctament
5005     guidata(gcbo,fig) % Desem els canvis fets fins ara
5006     RadioStd();
5007     fig=guidata(gcbo); % llegeix els canvis acabats de fer
5008
5009     set(fig.radioCust,'enable','on','Value',0);
5010
5011     % Panel WaveletPT
5012     %Modifiquem el valor màxim de l'slider del nivell de descomposició
5013     % el valor que mostrarà i el pas ( en percentatge)
5014     set(fig.sliderNivellDescompWPT,...
5015         'Max',fileGlobal.sJ,...
5016         'Value',fileGlobal.sJ,...
5017         'SliderStep',[1/(fileGlobal.sJ-1) 1/(fileGlobal.sJ-1)]);
5018     set(fig.entraNivellDescompWPT,...
5019         'String',fileGlobal.sJ);
5020     % i posem la cadena fig.Complete_dLibrary al seu listbox
5021     set(fig.listboxFEWPT,'String',fig.Complete_dLibrary);
5022     set(fig.buttonProcessWPT,'enable','on');
5023
5024     set(fig.checkGrid,'Value',1)
5025     set(fig.uipanelPlot,...
5026         'Visible','on',... % i farà visible el uipanelPlot
5027         'Position',[fig.panelsX ...
5028             (fig.panelPlotY-fig.panelSignalAlcada) ...
5029             fig.panelsAmplada ...
5030             fig.panelPlotAlcada]);
5031     set(fig.menuShowHidePlot,'Label','Hide plot menu'); % Canvia el nom
5032
5033     set(fig.textWait,...
5034         'String','Drawing...',...

```

```

5035         'Visible','off');
5036
5037     drawnow
5038
5039     % Farà que es vegi correctament el caràcter _:
5040     str = strrep(fileNameGlob, '_', '\_');
5041     str=['(' str ')']; % Hi afegim 'Totes les parts'
5042
5043     if strcmp(ContourOrProfile,'Profile')
5044         str=['Radial view ' str];
5045         FuncPlot(fig.axesFix,...
5046             str,...
5047             fileGlobal.xsignals,fileGlobal.ranswer,'Samples');
5048     end
5049
5050     set(fig.textWait,...
5051         'String','Please wait...',...
5052         'Visible','off');
5053
5054 end
5055
5056 guidata(gcbo,fig)           % Desem els canvis fets
5057 end
5058
5059 % ---Funció per habilitar o desabilitar tots els components
5060 function EnableDisable (gcbo,able,inici,varargin)
5061 fig=guidata(gcbo); % llegeix guidata.
5062 if nargin==4
5063     fig=varargin{1};
5064 end
5065 % Dependent de si inici==1 (si obrim un fitxer):
5066 % Si able=off els posa en enable=off
5067 % Si able=on els posa en enable=on
5068 global fileGlobal
5069
5070 it=fieldnames(fig);
5071 for i=1:size(it,1)
5072     if (~isempty(findstr(char(it(i,:)),'edit')))
5073         set(fig.(char(it(i,:))),'Enable',able);
5074     end
5075     if (~isempty(findstr(char(it(i,:)),'text')))
5076         set(fig.(char(it(i,:))),'Enable',able);
5077     end
5078     if (~isempty(findstr(char(it(i,:)),'var')))
5079         set(fig.(char(it(i,:))),'Enable',able);
5080         if inici==1
5081             set(fig.(char(it(i,:))),'Enable','off','String','');
5082         end
5083     end
5084     if (~isempty(findstr(char(it(i,:)),'popup')))
5085         set(fig.(char(it(i,:))),'Enable',able);
5086     end
5087     if (~isempty(findstr(char(it(i,:)),'button')))
5088         set(fig.(char(it(i,:))),'Enable',able);
5089         if inici==1
5090             set(fig.buttonProcess,'Visible','off');
5091             set(fig.buttonRemoveRec,'Enable','off');
5092             set(fig.buttonRemove,'Enable','off');
5093         end
5094     end

```

```

5095     end
5096     if (~isempty(findstr(char(it(i,:)), 'radio')))
5097         if inici==1
5098             set(fig.(char(it(i,:))), 'Enable', 'off', 'Value', 0);
5099         else
5100             set(fig.(char(it(i,:))), 'Enable', able)
5101         end
5102     end
5103     if (~isempty(findstr(char(it(i,:)), 'check')))
5104         if inici==1
5105             set(fig.(char(it(i,:))), 'Value', 0, 'Enable', able);
5106         else
5107             set(fig.(char(it(i,:))), 'Enable', able);
5108         end
5109     end
5110     if (~isempty(findstr(char(it(i,:)), 'entra')))
5111         if inici~=1
5112             set(fig.(char(it(i,:))), 'Enable', able);
5113         else
5114             set(fig.(char(it(i,:))), 'String', '');
5115         end
5116     end
5117
5118     if (~isempty(findstr(char(it(i,:)), 'slider')))
5119         if inici~=1
5120             set(fig.(char(it(i,:))), 'Enable', able);
5121         end
5122     end
5123     if (~isempty(findstr(char(it(i,:)), 'listbox')))
5124         if inici==1
5125             set(fig.(char(it(i,:))), 'string', '', 'Value', 1);
5126         end
5127     end
5128 end
5129
5130 if isempty(get(fig.varCoefficients, 'String'))
5131     set(fig.textCoefficients, 'Enable', 'off');
5132 end
5133
5134 if inici==1
5135     if nargin==4
5136         AxesResetIInvisible(fig)
5137     else
5138         AxesResetIInvisible();
5139     end
5140 set(fig.panelTaula, 'Visible', 'off')
5141
5142 % Deshabilitem les opcions del gràfic
5143 zoom reset;
5144 datacursormode off
5145 zoom off
5146 pan off
5147 % Panels que s'amaguen:
5148 set([fig.uipanelDiscrimAnalysis fig.uipanelListboxes ...
5149     fig.uipanelCustNK fig.uipanelPlot ...
5150     fig.uipanelReconstruct fig.uipanelWaveletPT...
5151     fig.uipanelReconstructCustDict], ...
5152     'Visible', 'off');
5153
5154 % Les marques dels seus menús
5155 set([fig.menuLDB fig.menuWaveletPT], ...

```

```

5156         'Checked','off');
5157     set(fig.menuShowHidePlot,'Label','Show plot menu');
5158
5159     % Amaguem la barra textWait per si ha quedat activa
5160     set(fig.textWait,'Visible','off')
5161
5162     % Inicialitzem els listboxes:
5163     fig.esquerra=fig.Complete_dLibrary;
5164     % i fig.dreta estarà buit
5165     fig.dreta={};
5166     % Última posició del listBox esquerre (màxims elements):
5167     fig.ultimaposicioEsquerra=length(fig.esquerra);
5168     % Última posició del listBox dret (màxims elements):
5169     fig.ultimaposicioDreta=0;
5170     set(fig.listBoxSortida,'String',fig.dreta,'Value',1);
5171     set(fig.listBoxEntrada,'String',fig.esquerra,'Value',1);
5172
5173     % Inicialitzem la variable dels subespais
5174     if isfield(fileGlobal,'subspaceReconstr')
5175         fileGlobal = rmfield(fileGlobal, 'subspaceReconstr');
5176     end
5177
5178     clear globals
5179
5180 end
5181
5182 guidata(gcbo,fig)           % Desem els canvis fets
5183 end
5184
5185
5186 function vect=ComparaCadena (varargin)
5187 % Funció que compara la cadena introduïda separada per comes (un sol
5188 % número) o guions (interval)
5189
5190 fig=guidata(gcbo); % llegeix guidata.
5191 global fileGlobal
5192 vect=[];
5193 pM=[];
5194 k=1;
5195 % Cas i=1, quan comencem;
5196 % Desem en una variable P_1 la cadena fins a la primera coma i
5197 % la resta a remain
5198 eval(sprintf('[P_1 remain]=strtok(varargin{1},','');',k));
5199 % busquem dins de la variable P_1 si hi ha un guió
5200 eval(sprintf('f=findstr(P_1,'-');',k));
5201
5202 if isempty(f) % Si no és un interval (un interval: 1-5)
5203     eval(sprintf('[a]=(P_1);',k));
5204     eval(sprintf('a=str2num(a);'));
5205     if isscalar(a) && mod(a,1)==0
5206         if a>=1
5207             vect=eval(sprintf('[a];'));
5208             eval(sprintf('b_old=a;'));
5209         end
5210     else
5211         set(fig.textWait,...
5212             'String','You must enter a valid numeric value.',...
5213             'fontSize',19,...
5214             'Visible','on');
5215         pause(1.5)

```

```

5216         set(fig.textWait,...
5217             'String','Please wait...',...
5218             'fontSize',35,...
5219             'Visible','off');
5220     % El vector final el posarem a 0 perquè no
5221     % processi
5222     vect=[];
5223     return % Farem que surti del bucle
5224 end
5225
5226 else % Si és un interval
5227     eval(sprintf('[a remainGuio]=strtok(P_%d, '-');',k));
5228     eval(sprintf('[b remainGuio]=strtok(remainGuio, '-');'));
5229     eval(sprintf('a=str2num(a);'));
5230     eval(sprintf('b=str2num(b);'));
5231     if isscalar(a) && isscalar(b) && mod(a,1)==0 && mod(b,1)==0
5232         if b>=1 && a>=1
5233             if b>a
5234                 vect=eval(sprintf('[a:b];'));
5235                 eval(sprintf('b_old=b;'));
5236             else
5237                 if b==a
5238                     vect=eval(sprintf('[b];'));
5239                 else
5240                     vect=eval(sprintf('[b:a];'));
5241                 end
5242                 eval(sprintf('b_old=a;'));
5243             end
5244         else
5245             eval(sprintf('b_old=0;'));
5246         end
5247     else
5248         set(fig.textWait,...
5249             'String','You must enter a valid numeric value.',...
5250             'fontSize',19,...
5251             'Visible','on');
5252         pause(1.5)
5253         set(fig.textWait,...
5254             'String','Please wait...',...
5255             'fontSize',35,...
5256             'Visible','off');
5257     % El vector final el posarem a 0 perquè no
5258     % processi
5259     vect=[];
5260     return % Farem que surti del bucle
5261 end
5262
5263 end
5264
5265 k=k+1;
5266
5267 while remain~=0
5268     if remain(1)==' '
5269         eval(sprintf('[P_%d remain]=strtok(remain, ' ', '');',k));
5270         eval(sprintf('f=findstr(P_%d, '-');',k));
5271         if isempty(f) % Si no és un interval
5272             eval(sprintf('[a]=(P_%d);',k));
5273             eval(sprintf('a=str2num(a);'));
5274
5275             vect=eval(sprintf('[vect a];'));
5276             if isscalar(a) && mod(a,1)==0

```

```

5277         if a>b_old && a>=1
5278             eval(sprintf('b_old=a;'));
5279         end
5280     else
5281         set(fig.textWait,...
5282             'String','You must enter a numeric value.',...
5283             'fontSize',19,...
5284             'Visible','on');
5285         pause(1.5)
5286         set(fig.textWait,...
5287             'String','Please wait... ',...
5288             'fontSize',35,...
5289             'Visible','off');
5290         % El vector final el posarem a 0 perquè no
5291         % processi
5292         vect=[];
5293         return % Farem que surti del bucle
5294     end
5295
5296     else % si és un interval
5297         eval(sprintf('[a remainGuio]=strtok(P_%d, '-');',k));
5298         eval(sprintf('[b remainGuio]=strtok(remainGuio, '-');'));
5299         eval(sprintf('a=str2num(a);'));
5300         eval(sprintf('b=str2num(b);'));
5301         if isscalar(a) && isscalar(b) && mod(a,1)==0 && mod(b,1)==0
5302             if b>=1 && a>=1
5303                 if b>a
5304                     vect=eval(sprintf('[vect a:b];'));
5305                     eval(sprintf('b_old=b;'));
5306                 else
5307                     if b==a
5308                         vect=eval(sprintf('[vect b];'));
5309                     else
5310                         vect=eval(sprintf('[vect b:a];'));
5311                     end
5312                     eval(sprintf('b_old=a;'));
5313                 end
5314             end
5315         else
5316             set(fig.textWait,...
5317                 'String','You must enter a numeric value.',...
5318                 'fontSize',19,...
5319                 'Visible','on');
5320             pause(1.5)
5321             set(fig.textWait,...
5322                 'String','Please wait... ',...
5323                 'fontSize',35,...
5324                 'Visible','off');
5325             % El vector final el posarem a 0 perquè no
5326             % processi
5327             vect=[];
5328             return % Farem que surti del bucle
5329         end
5330     end
5331     end
5332     k=k+1;
5333 end
5334
5335 if ~isempty(vect)
5336     vect=sort(vect);
5337     pM=vect;

```

```

5338     % Recollim els camps majors a la matriu:
5339     pM=pM(ismember(pM,1:size(fileGlobal.xsignals,1))==0);
5340     % Suprimim els camps majors a la matriu:
5341     vect(ismember(vect,1:size(fileGlobal.xsignals,1))==0)=[];
5342
5343     j=length(vect);
5344     for i=1:j
5345         if j>=i % evitem repeticions
5346             k = findstr(vect, vect(:,i));
5347             while length(k)>1
5348                 if j>=i
5349                     vect(:,k(end))=[];
5350                     k(:,(end))=[];
5351                     j=j-1;
5352                 end
5353             end
5354         end
5355     end
5356 end
5357 if ~isempty(pM) % Si la dada entrada supera la mida de la matriu...
5358     str={'Maximum Range: 1-' ...
5359         num2str(size(fileGlobal.Bspaces,1)) '. Vector changed'};
5360
5361     set(fig.textWait,...
5362         'String',str,...
5363         'fontSize',19,...
5364         'Visible','on');
5365
5366     pause(0.5)
5367     str=['1-' num2str(size(fileGlobal.Bspaces,1))];
5368     set(fig.editSelect,...
5369         'string',str);
5370     set(fig.textWait,...
5371         'String','Please Wait',...
5372         'fontSize',35,...
5373         'Visible','off');
5374 end
5375 end
5376
5377
5378 function FetElProcess (varargin)
5379 fig=guidata(gcbo); % llegeix guidata.
5380 global fileGlobal
5381
5382 check=get(fig.menuWaveletPT,'Checked');
5383 if strcmp(check,'on') % Si venim de Wavelet Packet Transform
5384
5385     % Inicialitzem la variable dels subespais
5386     if isfield(fileGlobal,'subspaceReconstr')
5387         fileGlobal = rmfield(fileGlobal, 'subspaceReconstr');
5388     end
5389     set(fig.listboxJKL,'string','', 'Value',1);
5390
5391     nmostres=(size(fileGlobal.BCoefitients,1));
5392
5393     j=1:log2(nmostres)+1;
5394     k=1:2^(j(1)-1);
5395     l=1:nmostres/(k(1));
5396
5397     set([fig.textJPop fig.popupJ ...

```

```

5398         fig.textKPop fig.popupK ...
5399         fig.textLPop fig.popupL], 'Enable', 'on');
5400     set(fig.popupJ, ...
5401         'String', j-1, ... % Volem que mostri 0 com a senyal original
5402         'Value', 1, ...    % 1 serà el senyal original
5403         'Enable', 'on');
5404     set(fig.popupK, ...
5405         'String', k-1, ... % Volem que mostri 0 com a senyal original
5406         'Value', 1, ...    % 1 serà el senyal original
5407         'Enable', 'on');
5408     set(fig.popupL, ...
5409         'String', l-1, ...
5410         'Value', 1, ...
5411         'Enable', 'on');
5412
5413     % Fem invisible el panel Wavelet Packet Transf.
5414     set([fig.uipanelWaveletPT fig.uipanelDiscrimAnalysis], ...
5415         'Visible', 'off');
5416     set([fig.menuWaveletPT fig.menuLDB], 'Checked', 'off');
5417
5418     set(fig.panelTaula, 'Visible', 'off');
5419
5420     drawnow
5421
5422     set(fig.buttonAddToReconstr, 'Enable', 'on')
5423     set([fig.buttonReconsCustDict ...
5424         fig.botoRemoveAllToReconstruct ...
5425         fig.botoRemoveToReconstruct], ...
5426         'enable', 'off');
5427
5428     set(fig.uipanelReconstructCustDict, ...
5429         'Visible', 'on');
5430
5431     drawnow
5432
5433     else % Si venim de Discriminant Analysis -> LDB
5434         a=fileGlobal.D_Ranking(:,1);
5435         set(fig.popupLibrary, 'String', a, 'Enable', 'on', 'Value', 1);
5436
5437         set(fig.radioSelect, 'Enable', 'on', 'Value', 1);
5438         set(fig.editSelect, 'Enable', 'on');
5439
5440         p=get(fig.textExempleSeleccio, 'Position');
5441         str={'Write numbers and intervals separated by comma. Ex: 1,3,5-12,14. Max
5442             range: 1-' ...
5443             num2str(size(fileGlobal.Bspaces,1))}};
5444         [outstring pos] = textwrap(fig.textExempleSeleccio, str);
5445         set(fig.textExempleSeleccio, ...
5446             'String', outstring, ...
5447             'Enable', 'on', ...
5448             'Position', [p(1)      ...
5449                         p(2)      ...
5450                         p(3)      ...
5451                         pos(4)]);
5452
5453         set(fig.buttonViewRec, 'Enable', 'on');
5454         set(fig.buttonRemoveRec, 'Enable', 'on');
5455
5456         set(fig.uipanelDiscrimAnalysis, 'Visible', 'off');
5457         set(fig.menuLDB, 'Checked', 'off');

```



```

5458
5459     set(fig.uipanelReconstruct,'visible','on');
5460
5461     % Ara modificarem el text d'exemple que ens mostra el panel de
5462     % reconstrucció per dir el rang màxim:
5463     set(fig.textLibraryPop,'Enable','on');
5464 end
5465
5466 if isfield(fileGlobal,'BCoefitients') && isfield(fileGlobal,'sJ')
5467     set([fig.textCoefficients fig.varCoefficients],'Enable','on');
5468     set(fig.varCoefficients, ...
5469         'String', ... % num mostres * (nivells + node(0,0))
5470         num2str(size(fileGlobal.xsignals,2)*(fileGlobal.sJ+1)));
5471 end
5472
5473 if isfield(fileGlobal,'xsignals')
5474     AxesResetInvisible();
5475
5476 end
5477
5478 % -----%
5479
5480 guidata(gcbo,fig);           % Desem els canvis fets
5481 end
5482
5483
5484
5485 function FuncPlot (figura,titol,varargin)
5486 % Arguments: axis,titol,xsignals,ranswer, (opcional:xlabel)
5487 global ContourOrProfile
5488 senyal=varargin{1};
5489 ranswer=varargin{2};
5490 if nargin==5
5491     labelX=varargin{3};
5492 else
5493     labelX='';
5494 end
5495 set(figura,'Visible','on');
5496 axes(figura);
5497
5498 % Array de caràcters amb els colors de línia:
5499 % El blanc '-w' no li posarem perquè no es veuria amb el fons
5500 % No hi posem el vermell perquè l'usarem per altres coses
5501 linia=['-b';'-g';'-c';'-m';'-y';'-k'];
5502 j=1; % índex per llegir aquest array
5503
5504 % per què sàpiga quan arribem al final del vector ranswer:
5505 fi=ranswer(:,end);
5506
5507 hold on
5508 if strcmp(ContourOrProfile,'Contour')
5509     y=0:360/(size(senyal,1)-1):360;
5510 else
5511     y=0:(size(senyal,1))/(size(senyal,1)-1):(size(senyal,1));
5512 end
5513
5514 for i=1:size(senyal,2)
5515
5516     plot(y,senyal(:,i),linia(j,:));
5517     if ranswer(:,i)~=fi % Si no estem al final

```

```

5518         % Compararem la posició de ranswer amb la següent per
5519         % saber si canvia
5520         if ranswer(:,i)<ranswer(:,i+1)
5521             j=j+1; % hi afegeix 1 per agafar el següent color
5522         end
5523     end
5524 end
5525
5526 title(titol, 'Color', 'w');
5527 xlabel(labelX, 'Color', 'w');
5528
5529 if strcmp(ContourOrProfile, 'Contour')
5530     set(figura, 'XTick', 0:360/12:360) % Parts en que dividirà l'eix
5531 else
5532     set(gca, 'XTick', 0:(size(senyal,1))/8:(size(senyal,1)))
5533 end
5534
5535 grid on
5536 zoom reset;
5537 axis tight
5538
5539 clear y
5540 end
5541
5542
5543 function FuncPolar (figura,titol,varargin)
5544
5545 senyal=varargin{1};
5546 ranswer=varargin{2};
5547 if nargin==3
5548     labelX=varargin{3};
5549 else
5550     labelX='';
5551 end
5552 set(figura, 'Visible', 'on');
5553 axes(figura);
5554
5555 % Array de caràcters amb els colors de línia:
5556 % El blanc '-w' no li posarem perquè no es veuria amb el fons
5557 % No hi posem el vermell perquè l'usarem per altres coses
5558 linia=['-b'; '-g'; '-c'; '-m'; '-y'; '-k'];
5559 j=1; % índex per llegir aquest array
5560
5561 % per què sàpiga quan arribem al final del vector ranswer:
5562 fi=ranswer(:,end);
5563
5564
5565 angle=0:2*pi/(size(senyal,1)-1):2*pi;
5566 % Si no canviéssim la diecció dels eixos hauriem de
5567 % canviar la direcció de l'angle: angle=-angle.
5568
5569 polar(angle', senyal(:,1), linia(j,:));
5570 hold on
5571 % Si ens trobéssim amb només dues mostres hauria de canviar ja
5572 % de color:
5573 if ranswer(:,1)~=fi % Si no tenim una sola mostra
5574     % Compararem la posició de ranswer amb la següent per
5575     % saber si canvia
5576     if ranswer(:,1)<ranswer(:,2)
5577         j=j+1; % hi afegeix 1 per agafar el següent color

```

```

5578     end
5579 end
5580
5581 for i=1:size(senyal,2)
5582     polar(angle',senyal(:,i),linia(j,:));
5583
5584     if ranswer(:,i)~=fi % Si no estem al final
5585         % Compararem la posició de ranswer amb la següent per
5586         % saber si canvia
5587         if ranswer(:,i)<ranswer(:,i+1)
5588             j=j+1; % hi afegeix 1 per agafar el següent color
5589         end
5590     end
5591 end
5592
5593 set(figura,'YDir','reverse') % canviem la direcció dels eixos
5594
5595 title(titol,'Color','w');
5596 xlabel(labelX,'Color','w');
5597 grid on
5598
5599 zoom reset;
5600
5601 end
5602
5603
5604 function CheckMostraJKL (varargin)
5605 fig=guidata(gcbo);
5606 global fileGlobal
5607 for j=1:13
5608     strCH=strcat('checkMostraJKL',num2str(j));
5609     set(fig.(char(strCH)), 'Value', 0);
5610 end
5611
5612 set(gcbo,'Value',1); % Tornem a marcar el quin hem marcat
5613
5614 posCaixa=get(fig.mostraJKL,'Position');
5615 posCaixaDespres=get(gcbo,'Position');
5616
5617 % Reposicionem la caixa
5618 set(fig.mostraJKL,...
5619     'Position',[posCaixa(1) ...
5620                posCaixaDespres(2) ...
5621                posCaixa(3) ...
5622                posCaixa(4)]);
5623
5624 % Hi posem el text
5625 strTag=get(gcbo,'Tag');
5626 filaCaixes=strtok(strTag,'checkMostraJKL');
5627 posLlibreria=['rankLibrary',filaCaixes];
5628 nomLlibreria=char(get(fig.(char(posLlibreria)), 'String'));
5629 fila=strmatch(nomLlibreria,fileGlobal.D_Ranking(:,1),'exact');
5630
5631 j=num2str(fileGlobal.Bspaces(fila,2));
5632 k=num2str(fileGlobal.Bspaces(fila,3));
5633 l=num2str(fileGlobal.Bspaces(fila,4));
5634 str=[j,',',k,',',l];
5635 set(fig.mostraJKL,'String',str,'Visible','on');
5636
5637 % Per fer córrer els checks a mida que fem anar l'slider:

```

```

5638 posFiles=['files',filaCaixes];
5639 fig.chkckStrFiles=get(fig.(char(posFiles)), 'String');
5640
5641 guidata(gcbo,fig)
5642 end
5643
5644
5645 function InicialitzaTaula (varargin)
5646 fig=guidata(gcbo); % llegeix guidata.
5647
5648 global fileGlobal
5649
5650 set(fig.panelTaula, 'Position', fig.midaPanelTaula);
5651 set(fig.titol1, 'Position', fig.midaTitol, 'Visible', 'on');
5652
5653 set(fig.titol2, 'Position', [fig.midaTitol(1)+fig.midaTitol(3) ...
5654     fig.midaTitol(2)     ...
5655     fig.midaTitol(3)     ...
5656     fig.midaTitol(4)     ], ...
5657     'Visible', 'on');
5658 fig.titol3=icontrol(fig.panelTaula, ...
5659     'Style', 'edit', ...
5660     'Enable', 'inactive', ...
5661     'String', 'j,k,l', ...
5662     'Position', [fig.midaPanelTaula(3)-2*fig.midaPanelTaula(3)*0.09 ...
5663     fig.midaTitol(2)     ...
5664     2*fig.midaPanelTaula(3)*0.09     ...
5665     fig.midaTitol(4)     ]);
5666
5667 set(fig.slideTaula, ...
5668     'Position', [fig.midaPanelTaula(3)- ...
5669     (2*0.09+0.09/2+0.05)*fig.midaPanelTaula(3) ...
5670     0 ...
5671     fig.midaPanelTaula(3)*0.09/2 ...
5672     fig.midaPanelTaula(4) ]);
5673
5674
5675 if length(fileGlobal.D_Ranking)<=13
5676     % posem els valors
5677     v=13;
5678     for i=1:size(fileGlobal.D_Ranking,1)
5679         strRL=strcat('rankLibrary', num2str(i));
5680         set(fig.(char(strRL)), ...
5681             'String', fileGlobal.D_Ranking(i,1), ...
5682             'Position', [fig.midaTitol(1) ...
5683             (fig.midaPanelTaula(4)/14*(v-1)) ...
5684             fig.midaTitol(3)     ...
5685             fig.midaTitol(4)     ], ...
5686             'visible', 'on');
5687         strRPD=strcat('rankPowDisc', num2str(i));
5688         set(fig.(char(strRPD)), ...
5689             'String', fileGlobal.D_Ranking(i,2), ...
5690             'Position', [fig.midaTitol(1)+fig.midaTitol(3) ...
5691             (fig.midaPanelTaula(4)/14*(v-1)) ...
5692             fig.midaTitol(3)     ...
5693             fig.midaTitol(4)     ], ...
5694             'visible', 'on');
5695         strF=strcat('files', num2str(i));
5696         set(fig.(char(strF)), ...
5697             'String', i, ...

```

```

5698         'visible','on');
5699     strCH=strcat('checkMostraJKL',num2str(i));
5700     set(fig.(char(strCH)),...
5701         'position',[fig.midaPanelTaula(3)-...
5702                 (2*0.09+0.05)*fig.midaPanelTaula(3) ...
5703                 (fig.midaPanelTaula(4)/14*(v-1)) ...
5704                 fig.midaPanelTaula(3)*0.05 ...
5705                 fig.midaTitol(4)],...
5706         'visible','on');
5707     v=v-1;
5708 end
5709 % Fem invisibles la resta de caixes
5710 for j=size(fileGlobal.D_Ranking,1)+1:13
5711     % 13 són el número de caixes creades
5712     strRL=strcat('rankLibrary',num2str(j));
5713     set(fig.(char(strRL)),...
5714         'Position',[fig.midaTitol(1) ...
5715                 (fig.midaPanelTaula(4)/14*(v-1)) ...
5716                 fig.midaTitol(3) ...
5717                 fig.midaTitol(4) ],...
5718         'visible','off');
5719     strRPD=strcat('rankPowDisc',num2str(j));
5720     set(fig.(char(strRPD)),...
5721         'Position',[fig.midaTitol(1)+fig.midaTitol(3) ...
5722                 (fig.midaPanelTaula(4)/14*(v-1)) ...
5723                 fig.midaTitol(3) ...
5724                 fig.midaTitol(4) ],...
5725         'visible','off');
5726     strF=strcat('files',num2str(j));
5727     set(fig.(char(strF)),...
5728         'visible','off');
5729     set(fig.slideTaula,...
5730         'visible','off');
5731     strCH=strcat('checkMostraJKL',num2str(j));
5732     set(fig.(char(strCH)),...
5733         'position',[fig.midaPanelTaula(3)-...
5734                 (2*0.09+0.05)*fig.midaPanelTaula(3) ...
5735                 (fig.midaPanelTaula(4)/14*(v-1)) ...
5736                 fig.midaPanelTaula(3)*0.05 ...
5737                 fig.midaTitol(4)],...
5738         'visible','off');
5739     v=v-1;
5740 end
5741 set(fig.slideTaula,...
5742     'visible','off');
5743
5744 else
5745     % Que mostri els 13 primers i faci visible l'slider, que serà
5746     % l'encarregat de fer-los correr
5747     v=13;
5748     for i=1:13
5749         strRL=strcat('rankLibrary',num2str(i));
5750         set(fig.(char(strRL)),...
5751             'String',fileGlobal.D_Ranking(i,1),...
5752             'Position',[fig.midaTitol(1) ...
5753                     (fig.midaPanelTaula(4)/14*(v-1)) ...
5754                     fig.midaTitol(3) ...
5755                     fig.midaTitol(4) ],...
5756             'visible','on');
5757         strRPD=strcat('rankPowDisc',num2str(i));
5758         set(fig.(char(strRPD)),...

```

```

5759         'String',fileGlobal.D_Ranking(i,2),...
5760         'Position',[fig.midaTitol(1)+fig.midaTitol(3) ...
5761                 (fig.midaPanelTaula(4)/14*(v-1)) ...
5762                 fig.midaTitol(3) ...
5763                 fig.midaTitol(4) ],...
5764         'visible','on');
5765     strF=strcat('files',num2str(i));
5766     set(fig.(char(strF)),...
5767         'String',i,...
5768         'visible','on');
5769     strCH=strcat('checkMostraJKL',num2str(i));
5770     set(fig.(char(strCH)),...
5771         'position',[fig.midaPanelTaula(3)-...
5772                 (2*0.09+0.05)*fig.midaPanelTaula(3) ...
5773                 (fig.midaPanelTaula(4)/14*(v-1)) ...
5774                 fig.midaPanelTaula(3)*0.05 ...
5775                 fig.midaTitol(4)],...
5776         'visible','on');
5777     v=v-1;
5778 end
5779 resta=size(fileGlobal.D_Ranking,1)-13;
5780 set(fig.slideTaula,...
5781     'Max',resta+1,...
5782     'Value',resta+1,...
5783     'SliderStep',[1/resta ...
5784                 1/resta]);
5785 fig.posicioSlide_previ=size(fileGlobal.D_Ranking,1)-13;
5786 set(fig.slideTaula,...
5787     'visible','on');
5788 end
5789 set(fig.mostraJKL,...
5790     'Position',[fig.midaPanelTaula(3)-...
5791                 2*fig.midaPanelTaula(3)*0.09 ...
5792                 (fig.midaPanelTaula(4)/14*12) ...
5793                 2*fig.midaPanelTaula(3)*0.09 ...
5794                 fig.midaTitol(4) ]);
5795 % EXTENT
5796 extRL=get(fig.rankLibrary1,'Extent');
5797 antRL=get(fig.rankLibrary1,'position');
5798
5799 if extRL(3)>antRL(3)
5800     Extent(extRL,antRL);
5801 end
5802
5803 extRPD=get(fig.rankPowDiscl,'Extent');
5804 antRPD=get(fig.rankPowDiscl,'position');
5805 if extRPD(3)>antRPD(3)
5806     ExtentTaula(extRPD,antRPD);
5807 end
5808
5809 % Desmarquem tots els Checks
5810 for j=1:13
5811     strCH=strcat('checkMostraJKL',num2str(j));
5812     set(fig.(char(strCH)),'Value',0);
5813 end
5814 % Marquem el primer i hi posem les seves dades
5815 set(fig.checkMostraJKL1,'Value',1);
5816
5817 j=num2str(fileGlobal.Bspaces(1,2));
5818 k=num2str(fileGlobal.Bspaces(1,3));
5819 l=num2str(fileGlobal.Bspaces(1,4));

```

```

5820 str=[j, ',', k, ',', l];
5821 set(fig.mostraJKL, 'String', str);
5822
5823 % Per fer córrer els checks a mida que fem anar l'slider:
5824 fig.chkckStrFiles='1';
5825
5826 set(fig.panelTaula, 'Visible', 'on');
5827
5828 clear v j k l str
5829 guidata(gcbo, fig) % Desem els canvis fets
5830 end
5831
5832
5833 function ExtentTaula (varargin)
5834 fig=guidata(gcbo);
5835 extRL=varargin{1};
5836 antRL=varargin{2};
5837
5838 it=fieldnames(fig);
5839 p=get(fig.panelTaula, 'position');
5840 set(fig.panelTaula, ...
5841     'position', [p(1)-(extRL(3)-antRL(3)) ...
5842                 p(2) ...
5843                 p(3)+2*(extRL(3)-antRL(3)) ...
5844                 p(4)]);
5845 p=get(fig.panelTaula, 'position');
5846 pSlide=get(fig.slideTaula, 'Position');
5847 pCh=get(fig.checkMostraJKL1, 'position');
5848 pmJKL=get(fig.mostraJKL, 'position');
5849 for i=1:13
5850     strCH=strcat('checkMostraJKL', num2str(i));
5851     set(fig.(char(strCH)), ...
5852         'position', [p(3)-pmJKL(3)-pCh(3) ...
5853                     pCh(2) ...
5854                     pCh(3) ...
5855                     pCh(4)]);
5856 end
5857 set(fig.mostraJKL, 'position', [p(3)-pmJKL(3) ...
5858                                 pmJKL(2) ...
5859                                 pmJKL(3) ...
5860                                 pmJKL(4)]);
5861 set(fig.slideTaula, ...
5862     'Position', [p(3)-pCh(3)-pmJKL(3)-pSlide(3) ...
5863                 pSlide(2) ...
5864                 pSlide(3) ...
5865                 pSlide(4)]);
5866 mf=get(fig.files1, 'position');
5867
5868 for i=1:size(it,1)
5869     if (~isempty(findstr(char(it(i,:)), 'rankLibrary'))) || ...
5870         (~isempty(findstr(char(it(i,:)), 'titoll1')))
5871         colum=get(fig.(char(it(i,:))), 'Position');
5872         set(fig.(char(it(i,:))), ...
5873             'Position', [mf(3) ...
5874                         colum(2) ...
5875                         colum(3)+(extRL(3)-antRL(3)) ...
5876                         colum(4)]);
5877
5878
5879 end

```

```

5880         if (~isempty(findstr(char(it(i,:)), 'rankPowDisc')) || ...
5881             (~isempty(findstr(char(it(i,:)), 'titol2'))))
5882             colum=get(fig.(char(it(i,:))), 'Position');
5883             set(fig.(char(it(i,:))), ...
5884                 'Position', [mf(3)+extRL(3) ...
5885                             colum(2) ...
5886                             colum(3)+(extRL(3)-antRL(3)) ...
5887                             colum(4)]);
5888
5889         end
5890     end
5891     guidata(gcbo, fig)
5892 end
5893
5894
5895 function SlideTaula (varargin)
5896 % Aquí arribem amb més de 13 elements i els volem fer córrer amunt i
5897 % avall amb l'slider
5898 fig=guidata(gcbo); % llegeix guidata.
5899 global fileGlobal
5900
5901 val = get(gcbo, 'Value');
5902
5903 min=get(fig.slideTaula, 'Min');
5904 max=get(fig.slideTaula, 'Max');
5905 numTaules=size(fileGlobal.D_Ranking,1)-max+1;
5906 v=round((size(fileGlobal.D_Ranking,1)-val)+1);
5907
5908 if ~isfield(fig, 'chckStrFiles')
5909     % vol dir que acabàvem d'iniciar la taula:
5910     fig.chckStrFiles='1';
5911 end
5912
5913 for i=1:numTaules
5914     strRL=strcat('rankLibrary', num2str(i));
5915     set(fig.(char(strRL)), ...
5916         'String', fileGlobal.D_Ranking((i+v-numTaules),1), ...
5917         'visible', 'on');
5918     strRPD=strcat('rankPowDisc', num2str(i));
5919     set(fig.(char(strRPD)), ...
5920         'String', fileGlobal.D_Ranking((i+v-numTaules),2), ...
5921         'visible', 'on');
5922     strF=strcat('files', num2str(i));
5923     set(fig.(char(strF)), ...
5924         'String', v+i-numTaules, ...
5925         'visible', 'on');
5926
5927     strCH=strcat('checkMostraJKL', num2str(i));
5928     set(fig.(char(strCH)), 'Value', 0);
5929     % Per fer córrer els checks a mida que fem anar l'slider:
5930     if strcmp(get(fig.(char(strF)), ...
5931         'String'), num2str(fig.chckStrFiles))
5932         coincideix=1;
5933         nomCH_nou=strCH;
5934     end
5935     set(fig.mostraJKL, 'visible', 'off');
5936 end
5937 % Si l'string guardat a fig.chckStrFiles coincideix amb l'string
5938 % de fila marcarem el check i farem visible la CaixaJKL, si no ens
5939 % esperarem a fer córrer més l'slider
5940 if exist('coincideix', 'var')

```



```

5941     set(fig.(char(nomCH_nou)), 'Value', 1);
5942     posCaixa=get(fig.mostraJKL, 'Position');
5943     posCaixaDespres=get(fig.(char(nomCH_nou)), 'Position');
5944     set(fig.mostraJKL, ...
5945         'Position', [posCaixa(1) ...
5946                     posCaixaDespres(2) ...
5947                     posCaixa(3) ...
5948                     posCaixa(4)], ...
5949         'Visible', 'on');
5950 end
5951
5952 if v==size(fileGlobal.D_Ranking,1)
5953     set(fig.slideTaula, 'Value', min);
5954 elseif v==numTaules
5955     set(fig.slideTaula, 'Value', max);
5956 end
5957
5958 guidata(gcbo, fig)           % Desem els canvis fets
5959 end
5960
5961
5962 function DibuixaBspaces (varargin)
5963 fig=guidata(gcbo); % llegeix guidata.
5964 global fileGlobal
5965
5966 AxesResetInvisible();
5967
5968 max=get(fig.slideTaula, 'Max');
5969
5970 axes(fig.axesSignals_VSxAuxProfile);
5971
5972 hold all % Sosté el plot, el color de línia actual i l'estil de
5973 % línia perquè les ordres següents no recomposin aquestes
5974 % propietats. El color de línia seguirà el cicle pels
5975 % colors predefinits.
5976 x=round(length(fileGlobal.xsignals)*0.2); % Per representar agafarem
5977 % el 20% de la longitud del senyal, les
5978 % x files més significatives, i
5979 % arrodoneix.
5980
5981 long=size(fileGlobal.Bspaces, 3);
5982 if size(fileGlobal.Bspaces, 3)>1 % Si té més d'una capa
5983     for i=1:(size(fileGlobal.Bspaces, 3))
5984         plot(fileGlobal.Bspaces(1:x, 1, i));
5985     end
5986 else
5987     plot(fileGlobal.Bspaces(1:x, 1));
5988 end
5989
5990 % Agafa els noms dels millors diccionaris i en fa una llegenda,
5991 % però només si no es passa d'una mida determinada aproximada:
5992 if size(fileGlobal.D_Ranking, 1)<=max
5993     str=cellstr(fileGlobal.D_Ranking(1:long));
5994
5995     % Farà que es vegi correctament el caràcter _:
5996     str1 = strrep(str, '_', '\_');
5997     legend(str1, 'Location', 'Best')
5998
5999     str2=[num2str(x) ' of ' ...
6000         num2str(length(fileGlobal.xsignals)) ' bases'];

```

```

6001     xlabel(str2,'Color','w');
6002 end
6003 hold off % Desactivem el hold
6004
6005 title(fig.axesSignals_VSxAuxProfile,...
6006     'Power Discrimination',...
6007     'Color','w');
6008
6009 axis tight
6010 grid on;
6011 zoom reset;
6012
6013 set(fig.axesSignals_VSxAuxProfile,'visible','on');
6014 guidata(gcbo,fig) % Desem els canvis fets
6015 end
6016
6017
6018 function AxesResetInvisible (varargin)
6019 fig=guidata(gcbo); % llegeix guidata.
6020 if nargin==1
6021     fig=varargin{1};
6022 end
6023 cla(fig.axesPlot,'reset');
6024 cla(fig.axesFix,'reset');
6025 cla(fig.axesPolar,'reset');
6026 cla(fig.axesPolarTots,'reset');
6027 cla(fig.axesSignals_VSxAuxProfile,'reset');
6028 cla(fig.axesxAux_VSsignalsProfile,'reset');
6029 cla(fig.axesSignals_VSxAux,'reset');
6030 cla(fig.axesxAux_VSsignals,'reset');
6031 cla(fig.axesSignals_VSxAuxProfileRec,'reset');
6032 cla(fig.axesSignals_VSxAuxRec,'reset');
6033
6034 set([fig.axesPlot fig.axesFix fig.axesPolar fig.axesPolarTots ...
6035     fig.axesSignals_VSxAuxProfile fig.axesxAux_VSsignalsProfile...
6036     fig.axesSignals_VSxAux fig.axesxAux_VSsignals...
6037     fig.axesSignals_VSxAuxProfileRec fig.axesSignals_VSxAuxRec],...
6038     'Visible','off');
6039
6040 guidata(gcbo,fig) % Desem els canvis fets
6041 end

```

# **ANNEX**

## **B**

**Assistent,  
funció LDB\_Wizard**

```

1  function LDB_Wizard (varargin)
2  %
3  %  Wizard per LDB_GUI
4
5  %-----
6  %  Tasques d'inicialització i
7  %-----
8
9  wiz.file=struct;
10
11  wiz.fig=varargin{1}; % Llegim les dades de la figura LDB.
12
13  %-----
14  %  Construcció de components
15  %-----
16
17
18  set(0,'Units','pixels') % ens assegurem que està en pixels
19  wiz.scnsize = get(0,'Screensize'); % agafem la mida de la pantalla
20
21  wiz.bdwidth = 5; % 5 píxels per incloure als costats de la finestra
22  wiz.topbdwidth = wiz.scnsize(3)/4; % Centrada a la pantalla
23
24  finestra = [wiz.bdwidth,                ... %[left      ...
25              0*wiz.scnsize(4) + wiz.topbdwidth, ... % bottom  ...
26              wiz.scnsize(3)*4/10,        ... % width    ...
27              wiz.scnsize(4)/2];         % height  ]
28
29  wiz.wizardLDB = figure(...
30      'MenuBar','none',... % amaga el menú estàndard
31      'Name','Wizard LDB',...% el nom de la figura serà Wizard LDB
32      'NumberTitle','off',...% treu el número de figura (figure: 1)
33      'Position',finestra,...%la posiciona en la posició calculada abans
34      'Resize','off',... % no ens deixa canviar-li el tamany
35      'Units','Pixels',... % la mesurarà en píxels
36      'WindowStyle','modal',...% Finestra modal: sempre visible
37      'Visible','off',... % La farem visible més tard
38      'CloseRequestFcn',@LDB_Wizard_CloseRequestFcn,...
39      'HandleVisibility','callback');
40
41  set(gcf,'Color',[0.5 0.5 0.5]); % Color de fons
42
43  % Mesures absolutes dels panels
44  % Tindrem 3 panels petits: panelIndicaLoad, panelIndicaConf i
45  % panelIndicaDepict
46  numPanelsXics=3;
47  wiz.panelsXicsAmplada = 70;
48  wiz.alcadaPanelBotons = 50;
49  wiz.panelsXicsAlcada = (finestra(4)-wiz.alcadaPanelBotons) / ...
50                        numPanelsXics;
51  % L'alçada de tots els panels serà de tota la figura menys
52  % els panels dels botons
53  wiz.panelsAlcada=(finestra(4)-wiz.alcadaPanelBotons);
54  wiz.midaPanelsGrans=[wiz.panelsXicsAmplada                ...
55                      wiz.alcadaPanelBotons                ...
56                      (finestra(3)-wiz.panelsXicsAmplada) ...
57                      wiz.panelsAlcada                      ];
58  % Panels petits
59  aT=18;
60  posIndicaLoad=[0

```

```

61         (wiz.panelsXicsAlcada*(numPanelsXics-1) +      ...
62         wiz.alcadaPanelBotons) ...
63         wiz.panelsXicsAmplada ...
64         wiz.panelsXicsAlcada];
65 wiz.panelIndicaLoad=uipanel('Parent',wiz.wizardLDB,...
66     'Units','Pixels',...
67     'ButtonDownFcn',@IndicaLoad_ButtonDownFcn,...
68     'Position',posIndicaLoad,...
69     'BorderType','etchedout');
70 wiz.txtIndicaLoad=icontrol(wiz.panelIndicaLoad,...
71     'Style','text',...
72     'String','Load',...
73     'FontWeight','bold',...
74     'FontSize',10,...
75     'ButtonDownFcn',@IndicaLoad_ButtonDownFcn,...
76     'Position',[2 ...
77         (wiz.panelsXicsAlcada/2-aT/2) ...
78         65 ...
79         aT ...
80         ]);
81 wiz.txtIndicaLoadBuit=icontrol(wiz.panelIndicaLoad,...
82     'Style','text',...
83     'Position',[posIndicaLoad(3)-3 ...
84         0.5 ...
85         4 ...
86         posIndicaLoad(4)-4],...
87     'Visible','on');
88 wiz.panelIndicaConf=uipanel('Parent',wiz.wizardLDB,...
89     'Units','Pixels',...
90     'Position',[0 ...
91         (wiz.panelsXicsAlcada*(numPanelsXics-2) +      ...
92         wiz.alcadaPanelBotons) ...
93         wiz.panelsXicsAmplada ...
94         wiz.panelsXicsAlcada ...
95         'BorderType','etchedout',...
96         'BackgroundColor',[0.5 0.5 0.5]);
97 wiz.txtIndicaConf=icontrol(wiz.panelIndicaConf,...
98     'Style','text',...
99     'String','Configure',...
100    'FontWeight','bold',...
101    'FontSize',10,...
102    'Position',[2 ...
103        (wiz.panelsXicsAlcada/2-aT/2) ...
104        65 ...
105        aT ...
106        ],...
107    'BackgroundColor',[0.5 0.5 0.5]);
108 % (panelsXicsAlcada/2-9: 9 és la meitat de l'alcada del text)
109 wiz.txtIndicaConfBuit=icontrol(wiz.panelIndicaConf,...
110    'Style','text',...
111    'Position',[posIndicaLoad(3)-3 ...
112        0.5 ...
113        4 ...
114        posIndicaLoad(4)-4],...
115    'Visible','off');
116 wiz.panelIndicaDepict=uipanel('Parent',wiz.wizardLDB,...
117    'Units','Pixels',...
118    'Position',[0 ...
119        (wiz.panelsXicsAlcada*(numPanelsXics-3) +      ...
120        wiz.alcadaPanelBotons) ...
121        wiz.panelsXicsAmplada ...
122        wiz.panelsXicsAlcada],...

```

```

122     'BorderType','etchedout',...
123     'BackgroundColor',[0.5 0.5 0.5]);
124
125 % str='Depict';
126 wiz.txtIndicaDepict=icontrol(wiz.panelIndicaDepict,...
127     'Style','text',...
128     'String','Analysis',...
129     'FontWeight','bold',...
130     'FontSize',10,...
131     'Position',[2
132         (wiz.panelsXicsAlcada/2-aT/2) ...
133         65 ...
134         aT ... ],...
135     'BackgroundColor',[0.5 0.5 0.5]);
136 % (panelsXicsAlcada/2-9: 9 és la meitat de l'alçada del text)
137 wiz.txtIndicaDepictBuit=icontrol(wiz.panelIndicaDepict,...
138     'Style','text',...
139     'Position',[posIndicaLoad(3)-3 ...
140         0.5 ...
141         4 ...
142         posIndicaLoad(4)-4],...
143     'Visible','off');
144 clear aT numPanelsXics posIndicaLoad
145
146 % Panels grans
147 wiz.panelLoad=uipanel('Parent',wiz.wizardLDB,...
148     'Units','Pixels',...
149     'Position',wiz.midaPanelsGrans,...
150     'BorderType','etchedout');
151
152 wiz.panelConf=uipanel('Parent',wiz.wizardLDB,...
153     'Units','Pixels',...
154     'Position',wiz.midaPanelsGrans,...
155     'Visible','off',...
156     'BorderType','etchedout');
157
158 wiz.panelDepict=uipanel('Parent',wiz.wizardLDB,...
159     'Units','Pixels',...
160     'Position',wiz.midaPanelsGrans,...
161     'Visible','off',...
162     'BorderType','etchedout');
163
164 % Panel de botons
165 midaPanelBotons=[wiz.panelsXicsAmplada ...
166     0 ...
167     (finestra(3)-wiz.panelsXicsAmplada) ...
168     wiz.alcadaPanelBotons ... ];
169 wiz.panelBotons=uipanel('Parent',wiz.wizardLDB,...
170     'Units','Pixels',...
171     'Position',midaPanelBotons,...
172     'BorderType','etchedout');
173
174 ab=22; % Alçada dels botons
175 mee=10; % Marge entre elements interns
176 mde=15; % Marge dret, esquerre, dalt i baix entre botons i panel.
177 bc=(finestra(3)-wiz.panelsXicsAmplada)/3-mde-mee; % Amplada botoCancel
178
179 wiz.buttonNext = icontrol('Parent',wiz.panelBotons,...
180     'style','pushbutton',...
181     'String','Next',...
182     'position',[(finestra(3)-wiz.panelsXicsAmplada-mde-bc/2) ...

```



```

244     'Units','Pixels',...
245     'Position',[10           ...
246             midaPanelSignal(4)/6*4   ...
247             (midaPanelSignal(3)-20) ...
248             midaPanelSignal(4)/6-10],...
249     'BorderType','line');
250 % [marge (alçadaPanel/(NumPanels+2)*(NumPanels-1) ...]
251 wiz.panel2Wiz=uipanel('Parent',wiz.uipanelSignalWiz,...
252     'Units','Pixels',...
253     'Position',[10           ...
254             midaPanelSignal(4)/6*3   ...
255             (midaPanelSignal(3)-20) ...
256             midaPanelSignal(4)/6-10],...
257     'BorderType','line');
258 % [marge (alçadaPanel/(NumPanels+2)*(NumPanels-3) ...]
259 wiz.panel4Wiz =uipanel('Parent',wiz.uipanelSignalWiz,...
260     'Units','Pixels',...
261     'Position',[10           ...
262             midaPanelSignal(4)/6*2   ...
263             (midaPanelSignal(3)-20) ...
264             midaPanelSignal(4)/6-10],...
265     'BorderType','line');
266 % [marge (alçadaPanel/(NumPanels+2)*(NumPanels-4) ...]
267 wiz.panel5Wiz =uipanel('Parent',wiz.uipanelSignalWiz,...
268     'Units','Pixels',...
269     'Position',[10           ...
270             midaPanelSignal(4)/6*1   ...
271             (midaPanelSignal(3)-20) ...
272             midaPanelSignal(4)/6-10],...
273     'BorderType','line');
274 % Elements dins d'aquets panels
275 midaPanelsPropietats=[10           ...
276             midaPanelSignal(4)/7*5   ...
277             (midaPanelSignal(3)-20) ...
278             midaPanelSignal(4)/6-10];
279 midaPaneletsText=[7           ...
280             (midaPanelsPropietats(4)-14)/2 ...
281             midaPanelsPropietats(3)/2-7 ...
282             14];
283 midaPaneletsVar=[midaPanelsPropietats(3)/2+2 ...
284             (midaPanelsPropietats(4)-14)/2 ...
285             midaPanelsPropietats(3)/2-7 ...
286             14];
287 wiz.textSignalNameWiz=uicontrol(wiz.panell1Wiz,...
288     'Style','text',...
289     'String','Signal Name: ',...
290     'HorizontalAlignment','Left',...
291     'Position',midaPaneletsText,...
292     'Enable','off');
293 wiz.textNumberSignalsWiz=uicontrol(wiz.panel2Wiz,...
294     'Style','text',...
295     'String','Number of Signals: ',...
296     'HorizontalAlignment','Left',...
297     'Position',midaPaneletsText,...
298     'Enable','off');
299 wiz.textVectorClassWiz=uicontrol(wiz.panel4Wiz,...
300     'Style','text',...
301     'String','Vector Class: ',...
302     'HorizontalAlignment','Left',...
303     'Position',midaPaneletsText,...
304     'Enable','off');

```



```

305 wiz.textLevelsWiz=icontrol(wiz.panel5Wiz,...
306     'Style','text',...
307     'String','Levels: ',...
308     'HorizontalAlignment','Left',...
309     'Position',midaPaneletsText,...
310     'Enable','off');
311 wiz.varSignalNameWiz=icontrol(wiz.panel1Wiz,...
312     'Style','text',...
313     'Position',midaPaneletsVar,...
314     'Enable','off');
315 wiz.varNumberSignalsWiz=icontrol(wiz.panel2Wiz,...
316     'Style','text',...
317     'Position',midaPaneletsVar,...
318     'Enable','off');
319 wiz.varVectorClassWiz=icontrol(wiz.panel4Wiz,...
320     'Style','text',...
321     'Position',midaPaneletsVar,...
322     'Enable','off');
323 wiz.varLevelsWiz=icontrol(wiz.panel5Wiz,...
324     'Style','text',...
325     'Position',midaPaneletsVar,...
326     'Enable','off');
327
328 wiz.radioContour=icontrol(wiz.panelLoad,...
329     'Style','radiobutton',...
330     'String','Load a contour data',...
331     'Value',1,...
332     'Position',[ (wiz.midaPanelsGrans(3)*9/25) ...
333                 (wiz.panelsAlcada)*4/4.5 ...
334                 (wiz.midaPanelsGrans(3)*9/25) ...
335                 (wiz.panelsAlcada/3)/5 ],...
336     'Callback',@RadioContourWiz);
337
338 wiz.ContourOrProfile='Contour';
339
340 wiz.radioProfile=icontrol(wiz.panelLoad,...
341     'Style','radiobutton',...
342     'String','Load a profile',...
343     'Value',0,...
344     'Position',[ (wiz.midaPanelsGrans(3)*9/25) ...
345                 (wiz.panelsAlcada)*4/4.9 ...
346                 (wiz.midaPanelsGrans(3)*9/25) ...
347                 (wiz.panelsAlcada/3)/5 ],...
348     'Callback',@RadioProfileWiz);
349
350 wiz.buttonLoad=icontrol(wiz.panelLoad,...
351     'Style','pushbutton',...
352     'String','Load',...
353     'Position',[ (wiz.midaPanelsGrans(3)/3) ...
354                 (wiz.panelsAlcada)*4/5.5 ...
355                 (wiz.midaPanelsGrans(3)/3) ...
356                 (wiz.panelsAlcada/3)/5 ],...
357     'Callback',@ButtonLoadWiz);
358
359 % panelWaveletMenu
360 midaPanelWaveletMenu=[15 ...
361     15 ...
362     (wiz.midaPanelsGrans(3)-30) ...
363     wiz.panelsAlcada*2/3];
364 wiz.panelWaveletMenuWiz=uipanel('Parent',wiz.panelConf,...
365     'TitlePosition','centertop ',...

```

```

366     'Title','Descriptors',...
367     'Units','Pixels',...
368     'Position',midaPanelWaveletMenu,...
369     'BorderType','beveledout');
370
371 % Elements del WaveletMenu
372 pos=[5      ...
373     wiz.panelsAlcada*4/5  ...
374     (wiz.midaPanelsGrans(3)-10)  ...
375     (wiz.panelsAlcada/5-10)];
376 wiz.txtLibTypeWiz=icontrol(wiz.panelConf,...
377     'Style','text',...
378     'FontSize',16,...
379     'FontWeight','bold',...
380     'Position',[pos(1) pos(2) pos(3) pos(4)]);
381 str={'Library Type:' ; 'Standard'};
382 % Si el text és major que la caixa de text 'textwrap' el partirà en
383 %cell arrays i ens retornarà la posició i l'string nou:
384 [outstring,newpos] = textwrap(wiz.txtLibTypeWiz,str);
385 set(wiz.txtLibTypeWiz,...
386     'String',outstring,...
387     'Position',[pos(1) (pos(2)-newpos(4)/3) pos(3) newpos(4)])
388
389 wiz.panelListboxesWiz=uipanel('Parent',wiz.panelWaveletMenuWiz,...
390     'Units','Pixels',...
391     'BorderType','beveledin',...
392     'Position',[12      ...
393                 12      ...
394                 (midaPanelWaveletMenu(3)-24)  ...
395                 (wiz.panelsAlcada*2/3-36)]);
396 % (36 és 12*2+12, ja que si no se superposa amb el títol)
397
398 % Obtenim les mides i posició de panelListboxesWiz
399 p=get(wiz.panelListboxesWiz,'Position');
400
401 mee=3; % Marge entre elements interns: ListBoxes-mee-botons
402 mde=6; % Marge dret, esquerre, dalt i baix entre listboxes i panel.
403 mbp=16; % Amplada del botó petit, botoAfegeix i botoTreu
404
405 % Fem una petita equació per trobar l'amplada dels ListBoxes:
406 % midaPanelAmplada=mde+midaListBoxes+mee+mbp+mee+midaListBoxes+mde
407 % midaPanelAmplada=2*mde+2*midaListBoxes+2*mee+mbp (...)
408 % midaListBoxes=(midaPanelAmplada-mbp-2*(mde+mee))/2
409 ampLB=(p(3)-mbp-2*(mde+mee))/2; % Amplada de ListBoxes
410
411 abT=14; % Alçada dels botons botoTots i botoCap
412 alcLB=(p(4)-2*mde-abT-mee); % Alçada dels Listboxs
413
414 wiz.listBoxEntradaWiz=icontrol(wiz.panelListboxesWiz,...
415     'Style','listbox',...
416     'Position',[mde (mde+abT+mee) ampLB alcLB],...
417     'BackgroundColor','w');
418 wiz.listBoxSortidaWiz=icontrol(wiz.panelListboxesWiz,...
419     'Style','listbox',...
420     'Position',[(mde+ampLB+2*mee+mbp) (mde+abT+mee) ampLB alcLB],...
421     'BackgroundColor','w');
422
423 wiz.botoTots=icontrol(wiz.panelListboxesWiz,...
424     'Style','pushbutton',...
425     'String','All',...
426     'Position',[mde mde ampLB abT],...

```

```

427     'Callback',@BotoTotsWiz);
428 wiz.botoCap=icontrol(wiz.panelListboxesWiz,...
429     'Style','pushbutton',...
430     'String','None',...
431     'Position',[(mde+ampLB+2*mee+mbp) mde ampLB abT],...
432     'Callback',@BotoCapWiz);
433
434 wiz.botoAfegeix=icontrol(wiz.panelListboxesWiz,...
435     'Style','pushbutton',...
436     'String','>',...
437     'FontWeight','bold',...
438     'Position',[(mde+ampLB+mee)           ...
439                 ((p(4)+mee+abT)/2+mee/2) ...
440                 mbp                       ...
441                 alcLB/3                   ],...
442     'Callback',@BotoAfegeixWiz);
443 wiz.botoTreu=icontrol(wiz.panelListboxesWiz,...
444     'Style','pushbutton',...
445     'String','<',...
446     'FontWeight','bold',...
447     'Position',[(mde+ampLB+mee)           ...
448                 ((p(4)+mee+abT)/2-alcLB/3-mee/2) ...
449                 mbp                       ...
450                 alcLB/3                   ],...
451     'Enable','off',...
452     'Callback',@BotoTreuWiz);
453
454 % Tercer clic -----
455 m=((wiz.panelsAlcada/3) - ((wiz.panelsAlcada/3)*2/3)) / 2;
456 midaPanelRanking=[15 ...
457                 (wiz.midaPanelsGrans(4)*2/3-m/4) ...
458                 (wiz.midaPanelsGrans(3)/2-15) ...
459                 (wiz.panelsAlcada/3)           ];
460 wiz.panelRanking=uipanel('Parent',wiz.panelDepict,...
461     'TitlePosition','centertop ',...
462     'Title','Most discriminant dictionaries',...
463     'Units','Pixels',...
464     'Position',midaPanelRanking,...
465     'BorderStyle','beveledout');
466
467
468 wiz.txtRanking=icontrol(wiz.panelRanking,...
469     'Style','text',...
470     'FontSize',10,...
471     'HorizontalAlignment','left',...
472     'TooltipString',...
473     'Ranking of the top five dictionaries prosecuted',...
474     'Position',[15/2           ...
475                 m/4           ...
476                 (midaPanelRanking(3)/2-22.5+15/2) ...
477                 (wiz.panelsAlcada/3-m)           ]);
478
479 % Power Discriminatory
480 str='Power Discriminatory of the top five dictionaries prosecuted';
481 wiz.txtPowDisc=icontrol(wiz.panelRanking,...
482     'Style','text',...
483     'FontSize',10,...
484     'HorizontalAlignment','left',...
485     'TooltipString',str,...
486     'Position',[(midaPanelRanking(3)/2+7.5) ...
487                 m/4           ...

```

```

488             (midaPanelRanking(3)/2-22.5+7.5) ...
489             (wiz.panelsAlcada/3-m)      ]);
490 clear newpos pos str outstring m
491
492 % Axes
493 midaAxis=[0 0 (wiz.midaPanelsGrans(3)) (wiz.panelsAlcada*2/3-15)];
494 wiz.axes2=axes('Parent',wiz.panelDepict,...
495             'Units','Pixels',...
496             'OuterPosition',midaAxis,...
497             'HandleVisibility','callback',...
498             'Visible','on');
499
500 % ---Panel Reconstruct
501 m=((wiz.panelsAlcada/3) - ((wiz.panelsAlcada/3)*2/3)) / 2;
502
503 midaPanelReconstruct=[wiz.midaPanelsGrans(3)/2 ...
504                     (wiz.midaPanelsGrans(4)*2/3)-m/4 ...
505                     (wiz.midaPanelsGrans(3)-30)*1/2 ...
506                     (wiz.panelsAlcada/3)      ]);
507
508 wiz.uipanelReconstruct=uipanel('Parent',wiz.panelDepict,...
509             'TitlePosition','centertop',...
510             'Title','Reconstruction menu',...
511             'Units','Pixels',...
512             'Position',midaPanelReconstruct,...
513             'BorderStyle','beveledout',...
514             'Visible','on');
515 % Elements del panelReconstruct
516 % Segona fila
517 nf=4;      % Numero files que volem
518 atGN=50;   % Amplada dels radios i text
519 mee=7;     % Marge entre elements
520 apGN= midaPanelReconstruct(3)-atGN; % Amplada del editText
521 mitjana=(midaPanelReconstruct(3)-atGN-mee-apGN*2/3)/2;
522
523 wiz.txtLibraryPop=icontrol(wiz.uipanelReconstruct,...
524             'Style','text',...
525             'String','Descript.: ',...
526             'FontWeight','bold',...
527             'Enable','on',...
528             'Position', [(midaPanelReconstruct(3))/8-atGN/4 ...
529                     (midaPanelReconstruct(4)/(nf+2)*(nf-0)+16/5) ...
530                     atGN ...
531                     16 ...
532                     ]);
533
534 wiz.popupLibrary=icontrol(wiz.uipanelReconstruct,...
535             'Style','popupmenu',...
536             'String','LCos',...
537             'Position', [(mitjana+mee+atGN) ...
538                     (midaPanelReconstruct(4)/(nf+2)*(nf-0)+16/3) ...
539                     apGN*2/3 ...
540                     16 ...
541                     ], ...
542             'BackgroundColor','w',...
543             'enable','on');
544
545 % Primera fila
546
547 wiz.radioButton=icontrol(wiz.uipanelReconstruct,...
548             'Style','radiobutton',...
549             'String','Select',...
550             'Enable','on',...
551             'Position', [(midaPanelReconstruct(3))/8-atGN/4 ...

```

```

549             (midaPanelReconstruct(4)/(nf+2)*(nf-2)+16/2) ...
550             atGN ...
551             16 ],...
552     'Enable','on',...
553     'Callback',@RadioSelectWiz);
554
555     wiz.editSelect=icontrol(wiz.uipanelReconstruct,...
556     'Style','edit',...
557     'String','',...
558     'TooltipString','Example: 1,3,5-12,14',...
559     'Position', [(midaPanelReconstruct(3))/8+atGN+2 ...
560                 (midaPanelReconstruct(4)/(nf+2)*(nf-2)+16/2) ...
561                 (midaPanelReconstruct(3)- ...
562                 (midaPanelReconstruct(3)/5+atGN+2+mee)) ...
563                 16 ], ...
564     'BackgroundColor','w',...
565     'Enable','on');
566
567     % Tercera fila
568     margeP=5; % Marge a l'exterior
569     midaAmpPanInt=midaPanelReconstruct(3)-margeP*2; % amplada interior
570
571     p=get(wiz.editSelect,'Position');
572
573     str={'Write numbers and intervals separated by comma.'};
574     wiz.txtExempleSeleccio=icontrol('Parent',wiz.uipanelReconstruct,...
575     'Style','text',...
576     'String',str,...
577     'Units','Pixels',...
578     'Position',[margeP ((p(2)-44)/2) midaAmpPanInt 44]);
579
580     [outstring pos] = textwrap(wiz.txtExempleSeleccio,str);
581     set(wiz.txtExempleSeleccio,'String',outstring,...
582     'Position',[margeP ...
583                 pos(2)+margeP ...
584                 midaAmpPanInt ...
585                 pos(4)],...
586     'Enable','on');
587
588
589     clear abT alcLB ampLB apGN atGN fletxa* m matgeP mb mbp mde mee ...
590     midaAmpPanInt mitjana nf outstring p pos str
591
592     m=wiz.midaPanelsGrans(3) + wiz.panelsXicsAmplada;
593     finestra(1)=(wiz.scnsize(3)-m) / 2;
594
595     % Fem ara visible la finestra, ja que té tots els components formats
596     set(wiz.wizardLDB,'Position',finestra,'Visible','on')
597
598     % Mida de la finestra inicial: panels xics + panelconf:
599     wiz.finestraInic=finestra;
600
601     % Comptador pels botons d'endavant i endarrere
602     wiz.comptaPosicio=0;
603
604     clear mida* finestra m
605
606     %-----
607     % Podem carregar ja les llibreries
608     load Complete_dLibrary;

```

```

609 % Mirem de no carregar 'LSin' ni 'LCos'
610 wiz.Complete_dLibrary=Complete_dLibrary;
611 wiz.Complete_dLibrary((ismember(wiz.Complete_dLibrary, 'LCos')==1))=[];
612 wiz.Complete_dLibrary((ismember(wiz.Complete_dLibrary, 'LSin')==1))=[];
613
614 wiz.Complete_dLibrary=sort(wiz.Complete_dLibrary); % Ordre alfabètic
615 %-----
616
617 wiz.file={}; %Inicialitzem la variable 'wiz.file'
618
619 guidata(gcf,wiz); % emmagatzema l'estructura
620
621 end
622
623
624 %-----
625 % Callbacks per LDB_Wizard
626 %-----
627
628 function LDB_Wizard_CloseRequestFcn (varargin)
629
630 ButtonExitWiz();
631
632 end
633
634
635 function ButtonFinishWiz (varargin)
636 wiz=guidata(gcbo); % llegeix guidata.
637 global fileGlobal fileNameGlob filePathNameGlob ...
638     ContourOrProfile loadOk
639
640 val=get(wiz.popupLibrary, 'Value'); %Obtenim el nom de la llibreria
641 str=char(wiz.file.D_Ranking(val,1));
642 cadena=get(wiz.editSelect, 'String');
643 vect=ComparaCadenaWiz(cadena);
644
645 if ~isempty(vect)
646 set([wiz.buttonFinish wiz.buttonPrev wiz.buttonExit ...
647     wiz.radioSelect wiz.editSelect wiz.popupLibrary],...
648     'Enable','off');
649 drawnow
650
651 if isfield(fileGlobal, 'xAux_Signal')
652     fileGlobal = rmfield(fileGlobal, 'xAux_Signal');
653 end
654
655 wiz.finish=1; % Variable que usarem a RadioStdFig
656
657
658 % [LO_D,HI_D,LO_R,HI_R] = WFILTERS('wname') computes four
659 % filters associated with the orthogonal or biorthogonal
660 % wavelet named in the string 'wname'.
661 % LO_R, the reconstruction low-pass filter
662 % HI_R, the reconstruction high-pass filter
663 [LO_R,HI_R]=wfilters(str, 'r'); % 'r': (Reconstruction filters)
664
665
666 wiz.file.xAux_Signal=fIWPT_RecTree(...
667     wiz.file.BCoefitients(vect, :, val), ...

```

```

668         wiz.file.Bspaces (vect,2:4,val), ...
669         size(wiz.file.xsignals,1), ...
670         LO_R,HI_R);
671     end
672
673
674     clear val str vect ext data colnoms pAx pW
675
676     if isfield(wiz.file,'xAux_Signal')
677         guidata(gcf,wiz); % emmagatzema l'estructura
678         FinishedWiz(gcbo);
679
680         set(wiz.fig.radioButton,'Value',1);
681         set(wiz.fig.editSelect,'Enable','on','String',cadena);
682
683         fileGlobal=wiz.file;
684         fileNameGlob=wiz.fileName;
685         filePathNameGlob=wiz.filePathName;
686         ContourOrProfile=wiz.ContourOrProfile;
687
688         loadOk=1; % Càrrega no satisfactòria
689
690         uiresume(wiz.wizardLDB);
691         delete(wiz.wizardLDB);
692     end
693 end
694
695
696 function ButtonNextWiz (varargin)
697 wiz=guidata(gcbo); % llegeix guidata.
698
699 wiz.comptaPosicio=wiz.comptaPosicio+1;
700
701 if wiz.comptaPosicio==1
702     if isfield(wiz.file,'xsignals') && isfield(wiz.file,'ranswer')...
703         && isfield(wiz.file,'sJ')
704
705         guidata(gcbo,wiz); % Desem els canvis fets
706         CarregaLliberies
707         wiz=guidata(gcbo); % Actualitza fig
708
709         %---Visible
710
711         set(wiz.panelIndicaLoad,'BackgroundColor',[0.5 0.5 0.5]);
712         set(wiz.panelIndicaConf,'BackgroundColor',...
713             get(0,'defaultUicontrolBackgroundColor'));
714         set(wiz.panelIndicaDepict,'BackgroundColor',[0.5 0.5 0.5]);
715
716         set(wiz.txtIndicaLoad,'BackgroundColor',[0.5 0.5 0.5]);
717         set(wiz.txtIndicaConf,'BackgroundColor',...
718             get(0,'defaultUicontrolBackgroundColor'));
719         set(wiz.txtIndicaDepict,'BackgroundColor',[0.5 0.5 0.5]);
720
721         set(wiz.txtIndicaLoadBuit, 'Visible','off');
722         set(wiz.txtIndicaConfBuit, 'Visible','on');
723         set(wiz.txtIndicaDepictBuit,'Visible','off');
724
725         % Panel dels Botons:
726         set(wiz.buttonNext,...
727             'TooltipString','Process',...

```

```

728         'Enable','off');
729
730     set(wiz.buttonPrev,...
731         'Visible','on',...
732         'Enable','on');
733
734     set(wiz.botoAfegeix,'enable','on');
735     set(wiz.botoTreu,'enable','off');
736     % Amaguem el panel Load:
737     set(wiz.panelLoad,'Visible','off');
738
739     % Fem visible el panel de configuració:
740     set(wiz.panelConf,'Visible','on');
741
742 else
743     % Restem la posició que havíem sumat, ja que no seguirem
744     wiz.comptaPosicio=wiz.comptaPosicio-1;
745     msgbox('The uploaded file is invalid.',...
746         ' File Type Error ','Error')
747 end
748 end
749
750 if wiz.comptaPosicio==2 % Si estem al panel del mig i volem processar
751
752     set(wiz.textWaitWiz,'Units','Pixels',...
753         'Enable','on',...
754         'Visible','on');
755     set([wiz.buttonNext wiz.buttonPrev wiz.buttonExit ...
756         wiz.botoTots wiz.botoCap wiz.botoAfegeix wiz.botoTreu],...
757         'Enable','off');
758     drawnow % Regenerem la figura
759
760     %---Feina
761
762     % Creem variables
763     for k=1:116
764         % Fins a 1 menys que Complete_dLibrary, ja que si són tots
765         % anirà a 'All'
766         eval(sprintf('X_%d=[];',k));
767     end
768     for k=1:length(wiz.dreta)
769         eval(sprintf('X_%d=char(wiz.dreta(k,1));',k));
770     end
771
772     % Si hem seleccionat tots els diccionaris entrarà l'string 'All'
773     if length(wiz.dreta)==length(wiz.Complete_dLibrary)
774         [wiz.file.Bspaces,wiz.file.BCoefitients,...
775         wiz.file.D_Ranking]=f_mlldb2(wiz.file.xsignals,...
776         wiz.file.ranswer,wiz.file.sJ,'Std','All');
777     else
778     % La funció found_lib de f_mlldb2 compara per string amb el cell
779     % array Complete_dLibrary. Li he de passar string per string
780     [wiz.file.Bspaces,wiz.file.BCoefitients,...
781     wiz.file.D_Ranking]=f_mlldb2(wiz.file.xsignals,...
782     wiz.file.ranswer,wiz.file.sJ,'Std',...
783     X_1,X_2,X_3,X_4,X_5,X_6,X_7,X_8,X_9,X_10,...
784     X_11,X_12,X_13,X_14,X_15,X_16,X_17,X_18,X_19,X_20,...
785     X_21,X_22,X_23,X_24,X_25,X_26,X_27,X_28,X_29,X_30,...
786     X_31,X_32,X_33,X_34,X_35,X_36,X_37,X_38,X_39,X_40,...
787     X_41,X_42,X_43,X_44,X_45,X_46,X_47,X_48,X_49,X_50,...

```



```

788     X_51,X_52,X_53,X_54,X_55,X_56,X_57,X_58,X_59,X_60,...
789     X_61,X_62,X_63,X_64,X_65,X_66,X_67,X_68,X_69,X_70,...
790     X_71,X_72,X_73,X_74,X_75,X_76,X_77,X_78,X_79,X_80,...
791     X_81,X_82,X_83,X_84,X_85,X_86,X_87,X_88,X_89,X_90,...
792     X_91,X_92,X_93,X_94,X_95,X_96,X_97,X_98,X_99,X_100,...
793     X_101,X_102,X_103,X_104,X_105,X_106,X_107,X_108,X_109,X_110,...
794     X_111,X_112,X_113,X_114,X_115,X_116);
795 end
796 clear X_* k
797
798 %---Visible
799
800 % Posem les dades obtingudes al panel que s'obrirà
801 % Si només volem indicar el nom:
802 j={'1.- '; '2.- '; '3.- '; '4.- '; '5.- '};
803 if size(wiz.file.D_Ranking,1)<5
804     str1=cellstr(wiz.file.D_Ranking(1:end,1));
805     string=cell(1,(size(wiz.file.D_Ranking,1))); %Reservem memòria
806     for i=1:(size(wiz.file.D_Ranking,1)) % Concatenem la posició
807         string(1,i)=strcat(j(i),str1(i)); % i el nom
808     end
809 else
810     str1=cellstr(wiz.file.D_Ranking(1:5,1));
811     string=cell(1,5); % Reservem memòria
812     for i=1:5 % Concatenem la posició i el nom
813         string(1,i)=strcat(j(i),str1(i));
814     end
815 end
816
817 string=string';
818 string=strjust(string, 'left');
819
820 set(wiz.txtRanking, 'String', string);
821 long=size(string,1);
822 % Agafem el PD de la base:
823 stringPowDisc=(wiz.file.D_Ranking(1:long,2));
824 stringPowDisc=stringPowDisc';
825 set(wiz.txtPowDisc, 'String', stringPowDisc);
826
827 set(wiz.axes2, 'Visible', 'on');
828 axes(wiz.axes2);
829
830 % Fem un reset a axes2
831 cla(wiz.axes2, 'reset')
832
833 hold all % Sosté el plot, el color de línia actual i l'estil de
834         % línia perquè les ordres següents no recomposin aquestes
835
836
837         % propietats. El color de línia seguirà el cicle pels
838         % colors predefinitos.
839 x=round(length(wiz.file.xsignals)*0.2); % Per representar agafarem
840         % el 20% de la longitud del senyal, les
841         % x files més significatives, i
842         % arrodoneix.
843 if size(wiz.file.Bspaces,3)>1
844     for i=1:long % Volem les 5 millors, agafarem 5 capes enrere
845         plot(wiz.file.Bspaces(1:x,1,i));
846     end
847 else

```

```

848         plot(wiz.file.Bspaces(1:x,1));
849     end
850     % Agafa els noms dels 5 millors diccionaris:
851     str=cellstr(wiz.file.D_Ranking(1:long));
852     legend(str, 'Location', 'Best')
853     str=[num2str(x) ' of ' ...
854         num2str(length(wiz.file.xsignals)) ' bases'];
855     xlabel(str);
856     hold off % Desactivem el hold
857
858     title(wiz.axes2, 'Descriptor discrimination profiles');
859
860     axis tight
861     zoom reset;
862
863     clear i j long x str*
864
865     % Panels xics
866     set(wiz.panelIndicaLoad, ...
867         'BackgroundColor', [0.5 0.5 0.5]);
868     set(wiz.panelIndicaConf, ...
869         'BackgroundColor', [0.5 0.5 0.5]);
870     set(wiz.panelIndicaDepict, ...
871         'BackgroundColor', get(0, 'defaultUicontrolBackgroundColor'));
872
873     set(wiz.txtIndicaLoad, ...
874         'BackgroundColor', [0.5 0.5 0.5]);
875     set(wiz.txtIndicaConf, ...
876         'BackgroundColor', [0.5 0.5 0.5]);
877     set(wiz.txtIndicaDepict, ...
878         'BackgroundColor', get(0, 'defaultUicontrolBackgroundColor'));
879
880     set(wiz.txtIndicaLoadBuit, 'Visible', 'off');
881     set(wiz.txtIndicaConfBuit, 'Visible', 'off');
882     set(wiz.txtIndicaDepictBuit, 'Visible', 'on');
883     % Amaguem la barra wait
884     set(wiz.textWaitWiz, 'Visible', 'off');
885
886     % Panel dels Botons:
887     set(wiz.buttonNext, ...
888         'TooltipString', '', ...
889         'Visible', 'off');
890
891     set(wiz.buttonPrev, ...
892         'Visible', 'on', ...
893         'Enable', 'on');
894     set([wiz.buttonNext wiz.buttonPrev wiz.buttonExit ...
895         wiz.botoTots wiz.botoCap wiz.botoAfegeix wiz.botoTreu], ...
896         'Enable', 'on');
897     drawnow % Regenera la figura i fa més sensible la GUI.
898
899     % Afegim el boto finish
900
901     set(wiz.buttonFinish, ...
902         'Enable', 'on', ...
903         'Visible', 'on');
904     set(wiz.botoTots, 'enable', 'off');
905     set(wiz.botoCap, 'enable', 'off');
906     set(wiz.botoAfegeix, 'enable', 'off');
907     set(wiz.botoTreu, 'enable', 'off');

```

```

908
909     % Panel de reconstrucció
910
911     a=wiz.file.D_Ranking(:,1);
912     set(wiz.popupLibrary,'String',a,'Enable','on');
913     set(wiz.radioSelect,'Enable','on','Value',1);
914
915     set(wiz.editSelect,'enable','on','String','');
916
917     % Ara modificarem el text d'exemple que ens mostra el panel de
918     % reconstrucció per dir el rang màxim:
919
920     set(wiz.editSelect,'ToolTipString',...
921         ['Example: 1,3,5-12,14. Maximum range: 1-' ...
922         num2str(size(wiz.file.Bspaces,1))]);
923
924     set(wiz.txtExempleSeleccio,'Enable','on');
925     set(wiz.txtLibraryPop,'Enable','on');
926
927     % Panel Depict
928     set(wiz.panelConf,'Visible','off');
929     set(wiz.panelDepict,'Visible','on');
930
931     clear mee p pPrev a
932
933 end
934
935 guidata(gcbo,wiz);           % Desa l'estructura.
936 end
937
938
939 function ButtonPrevWiz (varargin)
940 wiz=guidata(gcbo); % llegeix guidata.
941 wiz.comptaPosicio=wiz.comptaPosicio-1;
942
943 if wiz.comptaPosicio>=0 % Mentre no estem al panel principal...
944     if wiz.comptaPosicio==0 % Hem arribat al panel principal
945
946         set(wiz.buttonNext,...
947             'ToolTipString','',...
948             'Enable','on');
949
950         set(wiz.buttonPrev,...
951             'Visible','off');
952
953         % Panels xics
954         set(wiz.panelIndicaLoad,...
955             'BackgroundColor',...
956             get(0,'defaultUicontrolBackgroundColor'));
957         set(wiz.panelIndicaConf,...
958             'BackgroundColor',[0.5 0.5 0.5]);
959         set(wiz.panelIndicaDepict,...
960             'BackgroundColor',[0.5 0.5 0.5]);
961
962         set(wiz.txtIndicaLoad,...
963             'BackgroundColor',...
964             get(0,'defaultUicontrolBackgroundColor'));
965         set(wiz.txtIndicaConf,...
966             'BackgroundColor',[0.5 0.5 0.5]);
967         set(wiz.txtIndicaDepict,...

```

```

968         'BackgroundColor',[0.5 0.5 0.5]);
969
970     set(wiz.txtIndicaLoadBuit, 'Visible','on');
971     set(wiz.txtIndicaConfBuit, 'Visible','off');
972     set(wiz.txtIndicaDepictBuit,'Visible','off');
973
974     % Panels grans
975     set(wiz.panelConf,...
976         'Visible','off');
977     set(wiz.panelLoad,...
978         'Visible','on');
979
980 end
981 if wiz.comptaPosicio==1 % Del panel depict al panelConf
982
983     %---Feina
984     guidata(gcbo,wiz); % Desem els canvis fets
985     CarregaLliberies
986     wiz=guidata(gcbo); % Actualitza fig
987
988     set(wiz.botoTots,'enable','on');
989     set(wiz.botoCap,'enable','off');
990     set(wiz.botoAfegeix,'enable','on');
991     set(wiz.botoTreu,'enable','off');
992
993     %---Visible
994     % Botons
995
996     set(wiz.buttonNext,...
997         'ToolTipString','Process',...
998         'Enable','off',...
999         'Visible','on');
1000     set(wiz.buttonFinish,...
1001         'Visible','off');
1002
1003     % Panels xics
1004
1005     set(wiz.panelIndicaLoad,...
1006         'BackgroundColor',[0.5 0.5 0.5]);
1007     set(wiz.panelIndicaConf,...
1008         'BackgroundColor',...
1009         get(0,'defaultUicontrolBackgroundColor'));
1010     set(wiz.panelIndicaDepict,...
1011         'BackgroundColor',[0.5 0.5 0.5]);
1012
1013     set(wiz.txtIndicaLoad,...
1014         'BackgroundColor',[0.5 0.5 0.5]);
1015     set(wiz.txtIndicaConf,...
1016         'BackgroundColor',...
1017         get(0,'defaultUicontrolBackgroundColor'));
1018     set(wiz.txtIndicaDepict,...
1019         'BackgroundColor',[0.5 0.5 0.5]);
1020
1021     set(wiz.txtIndicaLoadBuit, 'Visible','off');
1022     set(wiz.txtIndicaConfBuit, 'Visible','on');
1023     set(wiz.txtIndicaDepictBuit,'Visible','off');
1024
1025     % Panels grans
1026     set(wiz.panelDepict,'Visible','off');
1027     set(wiz.panelConf,'Visible','on');

```

```

1028
1029         clear LCos LSin i j p
1030     end
1031 end
1032
1033 guidata(gcbo,wiz);           % Desa l'estructura.
1034 end
1035
1036
1037 function ButtonLoadWiz (varargin)
1038 wiz=guidata(gcbo);
1039 % Obrim una finestra a amb tots els fitxers a la vista i retorna el
1040 % directori del fitxer a path i el nom del fitxer a file:
1041
1042 ind=0; % variable que usarem per la detecció d'errors
1043 ok=1;  % variable que usarem per saber si hem fet 'Cancel·lar'
1044
1045 while ind==0 && ok~=0
1046     % Si les extensions no són correctes i no hem fet cancel·lar
1047     [fileName, path, ok] = uigetfile(...
1048         {'*.mat','MAT-files (*.mat)';... % 1
1049         '*.txt','TXT-files (*.txt)'};... % i 2 ok possibles
1050         'Open file');
1051
1052     % Detecció d'errors sobre *.* (no totes les versions de Matlab
1053     % en tenen)
1054     mat=findstr('.mat',fileName);
1055     % Busca la cadena '.mat' al nom del fitxer
1056     if isempty(mat)~=1
1057         if mat(end)+3==length(fileName)
1058             % Si en 3 posicions més el nom del fitxer s'acaba
1059             % l'extensió és correcta i no tornara a entrar al while
1060             ind=1;
1061         end
1062     end
1063     txt=findstr('.txt',fileName);
1064     if isempty(txt)~=1
1065         if txt(end)+3==length(fileName)
1066             % l'extensió és correcta i no tornara a entrar al while
1067             ind=1;
1068         end
1069     end
1070 end % end while
1071 if ind==1 && ok<=2
1072     % Si la càrrega ha estat correcta en extensió
1073     % i no hem fet 'Cancel·lar'...
1074
1075     % Netegem les dades anteriors, tot i quedaran substituïdes
1076     wiz.file=[]; wiz.fileName=[]; wiz.filePathName=[];
1077
1078     % Assignem el nom del fitxer a wiz.fileName
1079     wiz.fileName=fileName;
1080
1081     % Abans de netejar la interfície desarem algunes variables:
1082
1083
1084     % % Netegem la interfície
1085     EnableDisableWiz(gcbo,'off',1)
1086     wiz.comptaPosicio=0; % posem el comptador a 0
1087

```

```

1088     % Escriu tota la ruta del fitxer amb el nom inclòs:
1089     % D:\MATLAB7\work\Nom_del_fitxer.txt
1090     wiz.filePathName=strcat(path,wiz.fileName);
1091
1092     wiz.file=load(wiz.filePathName);
1093
1094     nom=['Wizard LDB - ' wiz.fileName];
1095     set(wiz.wizardLDB,'Name',nom);
1096
1097     %-----% Modifiquem la interfície inicial
1098     guidata(gcbo,wiz);
1099     wiz=Modif_interf_inicial_Wiz(gcbo);
1100
1101     %-----%
1102
1103
1104     clear ind ok txt mat path fileName;
1105     guidata(gcbo,wiz);
1106 end
1107 end
1108
1109
1110 function ButtonExitWiz (varargin)
1111 wiz=guidata(gcbo);
1112 global fileGlobal fileNameGlob ContourOrProfile loadOk
1113 % Les variables globals encara no les hem modificat. No ho fem fins
1114 % que no fem finish. Només les modifica el botó finish i la interfície
1115 % principal
1116
1117 set([wiz.buttonLoad wiz.buttonExit wiz.buttonNext wiz.buttonPrev ...
1118     wiz.radioContour wiz.radioProfile ...
1119     wiz.botoTots wiz.botoCap wiz.botoAfegeix wiz.botoTreu ...
1120     wiz.radioSelect wiz.editSelect wiz.popupLibrary ...
1121     wiz.buttonFinish],...
1122     'Enable','off');
1123
1124 wiz.finish=0; % Variable que usarem a RadioStdFig
1125
1126 cla(wiz.fig.axesFix,'reset');% és on hem carregat la vista temporal
1127 % % del fitxer temporal de l'assistent
1128 cla(wiz.fig.axesPlot,'reset');
1129 cla(wiz.fig.axesPolar,'reset');
1130 cla(wiz.fig.axesPolarTots,'reset');
1131 cla(wiz.fig.axesSignals_VSxAuxProfile,'reset');
1132 cla(wiz.fig.axesxAux_VSsignalsProfile,'reset');
1133 cla(wiz.fig.axesSignals_VSxAux,'reset');
1134 cla(wiz.fig.axesxAux_VSsignals,'reset');
1135 cla(wiz.fig.axesSignals_VSxAuxProfileRec,'reset');
1136 cla(wiz.fig.axesSignals_VSxAuxRec,'reset');
1137 set([wiz.fig.axesPlot wiz.fig.axesFix wiz.fig.axesPolar ...
1138     wiz.fig.axesPolarTots ...
1139     wiz.fig.axesSignals_VSxAuxProfile ...
1140     wiz.fig.axesxAux_VSsignalsProfile ...
1141     wiz.fig.axesSignals_VSxAux ...
1142     wiz.fig.axesxAux_VSsignals ...
1143     wiz.fig.axesSignals_VSxAuxProfileRec ...
1144     wiz.fig.axesSignals_VSxAuxRec],...
1145     'Visible','off');
1146 if ~isempty(fileNameGlob) % Si hem carregat algun fitxer...
1147     % Modifiquem la interfície de LDB

```

```

1148 set(wiz.fig.textSignalName, 'enable', 'on')
1149 set(wiz.fig.varSignalName, 'enable', 'on')
1150 % Posem el nom del senyal que hi teniem:
1151 set(wiz.fig.varSignalName, 'String', num2str(fileNameGlob));
1152
1153 if isfield(fileGlobal, 'xsignals')
1154     set(wiz.fig.textNumberSignals, 'enable', 'on');
1155     set(wiz.fig.varNumberSignals, 'enable', 'on');
1156     set(wiz.fig.varNumberSignals, ...
1157         'String', num2str(size(fileGlobal.xsignals,2)));
1158 end
1159 if isfield(fileGlobal, 'sJ')
1160     %si haviem carregat un fitxer a LDB ja teniem aquesta variable
1161     % calculada o venia amb les dades
1162     set(wiz.fig.textLevels, 'enable', 'on');
1163     set(wiz.fig.varLevels, 'enable', 'on');
1164     set(wiz.fig.varLevels, 'String', num2str(fileGlobal.sJ));
1165 end
1166
1167 if isfield(fileGlobal, 'ranswer') ...
1168     && isfield(fileGlobal, 'xsignals') ...
1169     && isfield(fileGlobal, 'sJ')
1170
1171     if isfield(fileGlobal, 'BCoefitients')
1172         set(wiz.fig.textCoefficients, 'enable', 'on');
1173         set(wiz.fig.varCoefficients, 'enable', 'on');
1174         set(wiz.fig.varCoefficients, 'String', ...
1175             num2str(size(fileGlobal.xsignals,2)*(fileGlobal.sJ+1)));
1176     end
1177
1178     set(wiz.fig.textVectorClass, 'enable', 'on');
1179     set(wiz.fig.varVectorClass, 'enable', 'on');
1180     % 'i' serà un vector que valdrà 1,2,3... fins al valor final
1181     % de ranswer
1182     i=1:fileGlobal.ranswer(end);
1183     set(wiz.fig.varVectorClass, 'String', num2str(i(end)));
1184
1185     set(wiz.fig.textGroupNumber, 'enable', 'on');
1186     set(wiz.fig.popupGroupNumber, 'enable', 'on');
1187
1188     set(wiz.fig.popupGroupNumber, 'String', i);
1189
1190     set(wiz.fig.textFirst, 'enable', 'on');
1191     set(wiz.fig.editFirst, 'enable', 'on');
1192     set(wiz.fig.textLast, 'enable', 'on');
1193     set(wiz.fig.editLast, 'enable', 'on');
1194
1195     i=get(wiz.fig.popupGroupNumber, 'Value');
1196     ipos = find(fileGlobal.ranswer==i);
1197
1198     % Posa el primer element de ipos a editFirst:
1199     set(wiz.fig.editFirst, 'String', num2str(ipos(1)));
1200     % Posa l'últim element de ipos a editLast:
1201     set(wiz.fig.editLast, 'String', num2str(ipos(end)));
1202
1203     set(wiz.fig.buttonView, 'Enable', 'on');
1204
1205     set(wiz.fig.menuPolar, 'Checked', 'off');
1206     ContourOrProfile=wiz.fig.contourOrProfiletmp;
1207     if strcmp(ContourOrProfile, 'Profile')

```

```

1208         set(wiz.fig.menuPolar, 'checked', 'off');
1209     else
1210         set(wiz.fig.menuPolar, 'checked', 'on');
1211     end
1212     set(wiz.fig.menuShowHidePlot, 'Label', 'Hide'); % Canvia el nom
1213 end
1214 if isfield(fileGlobal, 'xsignals') ...
1215     && isfield(fileGlobal, 'answer') ...
1216     && isfield(fileGlobal, 'sJ')
1217     set(wiz.fig.textLibType, 'enable', 'on');
1218
1219     %Panel Discriminant
1220     % Modifiquem el valor màxim de l'slider del nivell de
1221     % descomposició el valor que mostrarà i el pas,
1222     % en percentatge
1223     set(wiz.fig.sliderNivellDescomp, ...
1224         'Max', fileGlobal.sJ, ...
1225         'Value', fileGlobal.sJ, ...
1226         'SliderStep', [1/(fileGlobal.sJ-1) 1/(fileGlobal.sJ-1)]);
1227     set(wiz.fig.entraNivellDescomp, ...
1228         'String', fileGlobal.sJ);
1229
1230     % Cridem la funció RadioStdFig per ensenyar directament els
1231     % seus listboxs i el seu contingut correctament
1232     guidata(gcbo, wiz) % Desem els canvis fets fins ara
1233
1234     if strcmp(wiz.fig.StdOrCust, 'Std')
1235         RadioStdFig
1236     else
1237         RadioCustFig
1238         if ~isnan(wiz.fig.N)
1239             set(wiz.fig.entraN, 'String', wiz.fig.N)
1240         end
1241         if ~isnan(wiz.fig.K)
1242             set(wiz.fig.entraK, 'String', wiz.fig.K)
1243         end
1244         if isfield(wiz.fig, 'cust_V')
1245             set(wiz.fig.listboxEntrada, 'String', wiz.fig.cust_V)
1246             set(wiz.fig.buttonProcess, 'enable', 'on');
1247         end
1248     end
1249     set(wiz.fig.uipanelDiscrimAnalysis, ...
1250         'Visible', 'off');
1251     set(wiz.fig.menuLDB, 'Checked', 'off');
1252     wiz=guidata(gcbo); % llegeix els canvis acabats de fer
1253
1254     % Panel WaveletPT
1255     % Modifiquem el valor màxim de l'slider del nivell de
1256     % descomposició, el valor que mostrarà i el pas,
1257     % en percentatge
1258     set(wiz.fig.sliderNivellDescompWPT, ...
1259         'Max', fileGlobal.sJ, ...
1260         'Value', fileGlobal.sJ, ...
1261         'SliderStep', [1/(fileGlobal.sJ-1) 1/(fileGlobal.sJ-1)]);
1262     set(wiz.fig.entraNivellDescompWPT, ...
1263         'String', fileGlobal.sJ);
1264     % i posem la cadena wiz.Complete_dLibrary al seu listbox
1265     set(wiz.fig.listboxFEWPT, 'String', wiz.Complete_dLibrary);
1266     set(wiz.fig.buttonProcessWPT, 'enable', 'on');
1267 end
1268

```



```

1269     if isfield(fileGlobal,'xsignals') && ...
1270         isfield(wiz.file,'xsignals')
1271         % Si hem carregat un fitxer a l'assistent
1272         set(wiz.fig.checkGrid,'Value',1,'Enable','on')
1273         set(wiz.fig.checkPan,'Value',0,'Enable','on');
1274         set(wiz.fig.checkCursor,'Value',0,'Enable','on');
1275         set(wiz.fig.checkZoom,'Value',0,'Enable','on');
1276
1277         set(wiz.fig.uipanelPlot,...
1278             'Visible','on',... % i farà visible el uipanelPlot
1279             'Position',[wiz.fig.panelsX ...
1280             (wiz.fig.panelPlotY-wiz.fig.panelSignalAlcada) ...
1281             wiz.fig.panelsAmplada ...
1282             wiz.fig.panelPlotAlcada]);
1283
1284     end
1285     if isfield(fileGlobal,'xsignals')
1286         % si hem carregat un fitxer a la interfície principal
1287         set(wiz.fig.checkGrid,'Value',1,'Enable','on')
1288         set(wiz.fig.checkPan,'Value',0,'Enable','on');
1289         set(wiz.fig.checkCursor,'Value',0,'Enable','on');
1290         set(wiz.fig.checkZoom,'Value',0,'Enable','on');
1291
1292         % Farà que es vegi correctament el caràcter _ :
1293         str = strrep(fileNameGlob, '_', '\_');
1294         str=[str ' - All bases']; % Hi afegim 'Totes les parts'
1295         if strcmp(ContourOrProfile,'Profile')
1296             FuncPlotWiz(wiz.fig.axesFix,...
1297                 str,...
1298                 ContourOrProfile,...
1299                 fileGlobal.xsignals,fileGlobal.ranswer,...
1300                 'Samples');
1301         else
1302             FuncPolarWiz(wiz.fig.axesFix,...
1303                 str,...
1304                 fileGlobal.xsignals,fileGlobal.ranswer)
1305             set(wiz.fig.buttonRemove,'Enable','off');
1306         end
1307     end
1308 end
1309
1310 loadOk=0; % Càrrega no satisfactòria
1311
1312 uiresume(wiz.wizardLDB);
1313 delete(wiz.wizardLDB); % Tanca la figura
1314 end
1315
1316 function BotoTotsWiz (varargin)
1317 wiz=guidata(gcbo); % llegeix guidata.
1318
1319 wiz.ultimaposicioDreta=length(wiz.Complete_dLibrary);
1320 wiz.ultimaposicioEsquerra=0;
1321 wiz.dreta=wiz.Complete_dLibrary;
1322 wiz.esquerra={};
1323 set(wiz.listboxSortidaWiz,...
1324     'String',wiz.dreta,...
1325     'Value',1);
1326 set(wiz.listboxEntradaWiz,...
1327     'String',wiz.esquerra,...
1328     'Value',1);

```

```

1330
1331 if (isempty(wiz.esquerra))% Si només quedava un element a la llista...
1332     % Deshabilitarem el boto d'afegir-ne més
1333     set(wiz.botoAfegeix, 'enable', 'off');
1334     set(wiz.botoTots, 'enable', 'off');
1335     % Habilitem el boto de treure'n més
1336     set(wiz.botoTreu, 'enable', 'on');
1337     set(wiz.botoCap, 'enable', 'on');
1338
1339     set(wiz.buttonNext, 'enable', 'on');
1340 end
1341
1342 guidata(gcbo,wiz)
1343 end
1344
1345
1346 function BotoCapWiz (varargin)
1347 wiz=guidata(gcbo); % llegeix guidata.
1348
1349 wiz.ultimaposicioEsquerra=length(wiz.Complete_dLibrary);
1350 wiz.ultimaposicioDreta=0;
1351 wiz.dreta={};
1352 wiz.esquerra=wiz.Complete_dLibrary;
1353 set(wiz.listboxSortidaWiz, 'String',wiz.dreta, 'Value',1);
1354 set(wiz.listboxEntradaWiz, 'String',wiz.esquerra, 'Value',1);
1355
1356 if (isempty(wiz.dreta))% Si només quedava un element a la llista...
1357     % habilitarem el boto d'afegir-ne més
1358     set(wiz.botoAfegeix, 'enable', 'on');
1359     % deshabilitarem el boto de treure'n més
1360     set(wiz.botoTreu, 'enable', 'off');
1361     set(wiz.botoCap, 'enable', 'off');
1362     set(wiz.botoTots, 'enable', 'on');
1363
1364     set(wiz.buttonNext, 'enable', 'off');
1365 end
1366
1367 guidata(gcbo,wiz)
1368 end
1369
1370
1371 function BotoAfegeixWiz (varargin)
1372 wiz=guidata(gcbo); % llegeix guidata.
1373
1374 val = get(wiz.listboxEntradaWiz, 'Value');
1375 tmp=wiz.esquerra{val}; % converteix del cell array a un temporal
1376
1377 % Creem un cell array nou concatenat amb els escollits anteriorment
1378 wiz.dreta(length(wiz.dreta)+1,1)={tmp};
1379
1380 % Augmenta en 1 posició la última posició
1381 wiz.ultimaposicioDreta=wiz.ultimaposicioDreta+1;
1382
1383 wiz.dreta=sort(wiz.dreta); % endrecem
1384
1385 % Suprimim el camp afegit al listBoxDreta del listBoxEsquerra
1386 wiz.esquerra(val)=[];
1387 wiz.ultimaposicioEsquerra=wiz.ultimaposicioEsquerra-1;
1388
1389 % Si hem agafat l'últim element hem de posar 'Value' al penúltim:

```

```

1390 if val>length(wiz.esquerra)
1391     if wiz.ultimaposicioEsquerra>0
1392         set(wiz.listboxEntradaWiz, 'Value',wiz.ultimaposicioEsquerra);
1393     else
1394         set(wiz.botoTots, 'enable', 'off');
1395         % no pot prendre el valor 0, per tant el posem a 1:
1396         set(wiz.listboxEntradaWiz, 'Value',1);
1397     end
1398 end
1399
1400 if (isempty(wiz.esquerra))% Si només quedava un element a la llista...
1401     % Deshabilitarem el boto d'afegir-ne més
1402     set(wiz.botoAfegeix, 'enable', 'off');
1403     % Habilitem el boto de treure'n més
1404     set(wiz.botoTreu, 'enable', 'on');
1405     set(wiz.buttonNext, 'enable', 'on');
1406 else
1407     if (~isempty(wiz.dreta))
1408         set(wiz.botoTreu, 'enable', 'on');
1409         set(wiz.botoCap, 'enable', 'on');
1410         set(wiz.buttonNext, 'enable', 'on');
1411     else
1412         set(wiz.buttonNext, 'enable', 'off');
1413     end
1414     % habilitarem el boto d'afegir-ne més
1415     set(wiz.botoAfegeix, 'enable', 'on');
1416 end
1417
1418 set(wiz.listboxSortidaWiz, 'Value',1) % Ens assegurem que té un valor>0
1419 %Ho desem a les variables
1420 set(wiz.listboxSortidaWiz, 'String',wiz.dreta);
1421 set(wiz.listboxEntradaWiz, 'String',wiz.esquerra);
1422
1423 % Netegem les variables que ja no necessitarem
1424 clear val tmp
1425
1426 guidata(gcbo,wiz)
1427 end
1428
1429
1430 function BotoTreuWiz (varargin)
1431 wiz=guidata(gcbo); % llegeix guidata.
1432
1433 val = get(wiz.listboxSortidaWiz, 'Value');
1434 tmp=wiz.dreta{val}; % converteix del cell array a un temporal
1435
1436 wiz.dreta(val)=[];
1437 wiz.ultimaposicioDreta=wiz.ultimaposicioDreta-1;
1438
1439 % Afegim el camp suprimit del listBox2 al listBox1
1440 wiz.esquerra(length(wiz.esquerra)+1,1)={tmp};
1441 wiz.ultimaposicioEsquerra=wiz.ultimaposicioEsquerra+1;
1442 wiz.esquerra=sort(wiz.esquerra);
1443
1444 % Si hem agafat l'últim element hem de posar 'Value' al penúltim.
1445 if val>length(wiz.dreta)
1446     if wiz.ultimaposicioDreta>0
1447         set(wiz.listboxSortidaWiz, 'Value',wiz.ultimaposicioDreta);
1448     else
1449         set(wiz.botoAfegeix, 'enable', 'off');

```

```

1450         set(wiz.botoCap, 'enable', 'off');
1451         % no pot prendre el valor 0, per tant el posem a 1
1452         set(wiz.listboxSortidaWiz, 'Value', 1);
1453     end
1454 end
1455
1456 if (isempty(wiz.dreta))% Si només quedava un element a la llista...
1457     % habilitarem el boto d'afegir-ne més
1458     set(wiz.botoAfegeix, 'enable', 'on');
1459     % deshabilitarem el boto de treure'n més
1460     set(wiz.botoTreu, 'enable', 'off');
1461     set(wiz.buttonNext, 'enable', 'off');
1462 else
1463     if (~isempty(wiz.dreta))
1464         set(wiz.botoAfegeix, 'enable', 'on');
1465         set(wiz.botoTots, 'enable', 'on');
1466         set(wiz.buttonNext, 'enable', 'on');
1467     else
1468         set(wiz.buttonNext, 'enable', 'off');
1469     end
1470     % habilitarem el boto de treure'n més
1471     set(wiz.botoTreu, 'enable', 'on');
1472 end
1473
1474 set(wiz.listboxEntradaWiz, 'Value', 1) % Ens assegurem que té un valor>0
1475 % Ho desem a les variables
1476 set(wiz.listboxSortidaWiz, 'String', wiz.dreta);
1477 set(wiz.listboxEntradaWiz, 'String', wiz.esquerra);
1478
1479 % Netegem les variables que ja no necessitarem
1480 clear val tmp
1481
1482 guidata(gcbo, wiz)
1483 end
1484
1485
1486 function RadioContourWiz (varargin)
1487 wiz=guidata(gcbo); % llegeix guidata.
1488
1489 set(wiz.radioContour, 'Value', 1)
1490 set(wiz.radioProfile, 'Value', 0)
1491
1492 wiz.ContourOrProfile='Contour';
1493
1494 if isfield(wiz, 'signals')
1495     set([wiz.buttonLoad wiz.buttonExit wiz.buttonNext...
1496         wiz.radioContour wiz.radioProfile], 'Enable', 'off');
1497
1498     % Si ja haviem carregat un fitxer, que torni a dibuixar
1499     set(wiz.wizardLDB, 'WindowStyle', 'normal');
1500
1501     cla(wiz.fig.axesFix, 'reset')
1502     set(wiz.fig.axesFix, 'visible', 'off')
1503
1504     cla(wiz.fig.axesSignals_VSxAuxProfile, 'reset')
1505     set(wiz.fig.axesSignals_VSxAuxProfile, 'visible', 'off')
1506     cla(wiz.fig.axesxAux_VSsignalsProfile, 'reset')
1507     set(wiz.fig.axesxAux_VSsignalsProfile, 'visible', 'off')
1508     cla(wiz.fig.axesSignals_VSxAuxProfileRec, 'reset');
1509     set(wiz.fig.axesSignals_VSxAuxProfileRec, 'visible', 'off');

```

```

1510     cla(wiz.fig.axesSignals_VSxAuxRec, 'reset');
1511     set(wiz.fig.axesSignals_VSxAuxRec, 'Visible', 'off');
1512
1513     % Títol:
1514     % Farà que es vegi correctament el caràcter _:
1515     str = strrep(wiz.fileName, '_', '\_');
1516     str=[str ' - All bases']; % Hi afegim 'Totes les parts'
1517
1518     str1=[str ' - Polar'];
1519     FuncPolarWiz(wiz.fig.axesSignals_VSxAuxProfile,...
1520         str1,...
1521         wiz.file.xsignals,wiz.file.ranswer);
1522
1523     str2=[str ' - Plot'];
1524     FuncPlotWiz(wiz.fig.axesxAux_VSsignalsProfile,...
1525         str2,...
1526         'Contour',...
1527         wiz.file.xsignals,wiz.file.ranswer);
1528
1529     set([wiz.buttonLoad wiz.buttonExit wiz.buttonNext...
1530         wiz.radioContour wiz.radioProfile], 'Enable', 'on');
1531
1532
1533     uistack(wiz.wizardLDB, 'top');
1534
1535     set(wiz.wizardLDB, 'WindowStyle', 'modal');
1536
1537     drawnow
1538 end
1539 guidata(gcbo,wiz)
1540 end
1541
1542
1543 function RadioProfileWiz (varargin)
1544 wiz=guidata(gcbo); % llegeix guidata.
1545
1546 set(wiz.radioContour, 'Value', 0)
1547 set(wiz.radioProfile, 'Value', 1)
1548
1549 wiz.ContourOrProfile='Profile';
1550
1551 if isfield(wiz.file, 'xsignals')
1552     set([wiz.buttonLoad wiz.buttonExit wiz.buttonNext...
1553         wiz.radioContour wiz.radioProfile], 'Enable', 'off');
1554     drawnow
1555
1556     % Si ja havíem carregat un fitxer, que torni a dibuixar
1557     set(wiz.wizardLDB, 'WindowStyle', 'normal');
1558
1559     cla(wiz.fig.axesSignals_VSxAuxProfile, 'reset')
1560     set(wiz.fig.axesSignals_VSxAuxProfile, 'visible', 'off')
1561     cla(wiz.fig.axesxAux_VSsignalsProfile, 'reset')
1562     set(wiz.fig.axesxAux_VSsignalsProfile, 'visible', 'off')
1563     cla(wiz.fig.axesSignals_VSxAuxProfileRec, 'reset');
1564     set(wiz.fig.axesSignals_VSxAuxProfileRec, 'visible', 'off');
1565     cla(wiz.fig.axesSignals_VSxAuxRec, 'reset');
1566     set(wiz.fig.axesSignals_VSxAuxRec, 'Visible', 'off');
1567
1568     cla(wiz.fig.axesFix, 'reset')
1569     set(wiz.fig.axesFix, 'visible', 'off')

```

```

1570
1571 % Títol:
1572 % Farà que es vegi correctament el caràcter _:
1573 str = strrep(wiz.fileName, '_', '\_');
1574 str=['Radial view (' str ')']; % Hi afegim 'Totes les parts'
1575
1576 FuncPlotWiz(wiz.fig.axesFix,...
1577     str,...
1578     'Profile',...
1579     wiz.file.xsignals,wiz.file.ranswer)
1580
1581 set([wiz.buttonLoad wiz.buttonExit wiz.buttonNext...
1582     wiz.radioContour wiz.radioProfile], 'Enable', 'on');
1583
1584 uistack(wiz.wizardLDB, 'top');
1585 set(wiz.wizardLDB, 'WindowStyle', 'modal');
1586
1587 drawnow
1588 end
1589 guidata(gcbo,wiz)
1590 end
1591
1592
1593 function RadioSelectWiz (varargin)
1594 fig=guidata(gcbo); % llegeix guidata.
1595
1596 set(fig.radioSelect, 'Value', 1)
1597 % set(fig.radioAll, 'Value', 0)
1598
1599 set(fig.editSelect, 'Enable', 'on');
1600
1601 guidata(gcbo, fig);
1602 end
1603
1604
1605 %-----
1606 % Funcions d'utilitat per LDB_Wizard
1607 %-----
1608
1609 function CarregaLlibreries (varargin)
1610 wiz=guidata(gcbo); % llegeix guidata.
1611
1612 %---Feina
1613
1614 % Actualitzem el listbox del segon panel
1615
1616 wiz.esquerra=wiz.Complete_dLibrary; %Ho carreguem a 'esquerra'
1617 wiz.dreta={}; % Buidem 'dreta'
1618 % Última posició del listbox esquerre (màxims elements):
1619 wiz.ultimaposicioEsquerra=length(wiz.esquerra);
1620 % Última posició del listbox dret (màxims elements):
1621 wiz.ultimaposicioDreta=0;
1622
1623 set(wiz.listboxEntradaWiz, 'String', wiz.Complete_dLibrary);
1624 set(wiz.listboxSortidaWiz, 'String', wiz.dreta);
1625
1626 guidata(gcbo,wiz); % Desa l'estructura.
1627 end
1628

```

```

1629
1630 %---Funció per modificar la interfície quan es carreguen les variables
1631 function wiz=Modif_interf_inicial_Wiz (gcbo)
1632 wiz=guidata(gcbo); % llegeix guidata.
1633 % Modifiquem la interfície
1634 set([wiz.buttonLoad wiz.buttonExit ...
1635      wiz.radioContour wiz.radioProfile], 'Enable', 'off');
1636
1637 set(wiz.textSignalNameWiz, 'enable', 'on')
1638 set(wiz.varSignalNameWiz, 'enable', 'on')
1639 % Posem el nom del senyal
1640 set(wiz.varSignalNameWiz, 'String', num2str(wiz.fileName));
1641
1642 % Canviem la variable xdata a xsignals, si existeix:
1643 % F = ISFIELD(S,'camp') torna true si 'camp' és el nom d'un camp en
1644 % l'estructura S:
1645 if isfield(wiz.file, 'xdata') % xdata és el mateix que xsignals
1646     wiz.file.xsignals=wiz.file.xdata; % Guardem xdata a xsignals
1647     wiz.file = rmfield(wiz.file, 'xdata'); %Esborrem xdata de wiz.file
1648 end
1649 if isfield(wiz.file, 'xsignals')
1650     set(wiz.textNumberSignalsWiz, 'enable', 'on');
1651     set(wiz.varNumberSignalsWiz, 'enable', 'on');
1652     set(wiz.varNumberSignalsWiz, ...
1653         'String', num2str(size(wiz.file.xsignals,2)));
1654 end
1655 % Si no existeix sJ la creem:
1656 if ~isfield(wiz.file, 'sJ')
1657     if isfield(wiz.file, 'xsignals')
1658         wiz.file.sJ=log2(size(wiz.file.xsignals,1));
1659     end
1660 end
1661 if isfield(wiz.file, 'sJ')
1662     % i posem a enable les caixes de text corresponents
1663     set(wiz.textLevelsWiz, 'enable', 'on');
1664     set(wiz.varLevelsWiz, 'enable', 'on');
1665     set(wiz.varLevelsWiz, 'String', num2str(wiz.file.sJ));
1666 end
1667 % Canviem la variable rlabel a ranswer, si existeix:
1668 if isfield(wiz.file, 'rlabel') % rlabel és el mateix que ranswer
1669     wiz.file.ranswer=wiz.file.rlabel; % Guardem rlabel a ranswer
1670     wiz.file = rmfield(wiz.file, 'rlabel'); % Esborrem rlabel
1671 end
1672 if isfield(wiz.file, 'ranswer')
1673     set(wiz.textVectorClassWiz, 'enable', 'on');
1674     set(wiz.varVectorClassWiz, 'enable', 'on');
1675     % 'i' serà un vector que valdrà 1,2,3...
1676     % fins al valor final de ranswer
1677     i=1:wiz.file.ranswer(end);
1678     set(wiz.varVectorClassWiz, 'String', num2str(i(end)));
1679 end
1680 if isfield(wiz.file, 'xsignals') && ...
1681     isfield(wiz.file, 'sJ') && ...
1682     isfield(wiz.file, 'ranswer')
1683     set(wiz.wizardLDB, 'WindowStyle', 'normal');
1684
1685 % Títol:
1686 % Farà que es vegi correctament el caràcter _:
1687 str = strrep(wiz.fileName, '_', '\_');
1688 str=['(' str ')']; % Hi afegim 'Totes les parts'
1689

```

```

1690     if strcmp(wiz.ContourOrProfile, 'Profile');
1691
1692         FuncPlotWiz(wiz.fig.axesFix, ...
1693             str, ...
1694             'Profile', ...
1695             wiz.file.xsignals, wiz.file.ranswer)
1696
1697     else
1698         str1=['Contour view ' str];
1699         FuncPolarWiz(wiz.fig.axesSignals_VSxAuxProfile, ...
1700             str1, ...
1701             wiz.file.xsignals, wiz.file.ranswer);
1702
1703         str2=['Radial view ' str];
1704         FuncPlotWiz(wiz.fig.axesxAux_VSsignalsProfile, ...
1705             str2, ...
1706             'Contour', ...
1707             wiz.file.xsignals, wiz.file.ranswer);
1708     end
1709     set([wiz.buttonLoad wiz.buttonExit ...
1710         wiz.radioContour wiz.radioProfile], 'Enable', 'on');
1711
1712     uistack(wiz.wizardLDB, 'top');
1713     % Finestra modal: sempre visible:
1714     set(wiz.wizardLDB, 'WindowStyle', 'modal');
1715
1716     set(wiz.buttonNext, 'Enable', 'on')
1717
1718     drawnow
1719
1720 end
1721
1722 guidata(gcbo, wiz)
1723 end
1724
1725
1726 function EnableDisableWiz (gcbo, able, inici)
1727 wiz=guidata(gcbo); % llegeix guidata.
1728 % Depenent de si inici==1 (si obrim un fitxer):
1729 % Si able=off els posa en enable=off
1730 % Si able=on els posa en enable=on
1731
1732 mod=get(wiz.wizardLDB, 'WindowStyle');
1733 if strcmp(mod, 'modal') % Per evitar els beeps la farem invisible
1734     esmod=1; % Variable temporal per indicar que era modal
1735     set(wiz.wizardLDB, 'Visible', 'off');
1736 else % Si no és modal només posarem la variable que ho indica a 0
1737     esmod=0;
1738 end
1739 it=fieldnames(wiz);
1740 for i=1:size(it,1)
1741     if (~isempty(findstr(char(it(i,:)), 'text')))
1742         set(wiz.(char(it(i,:))), 'Enable', able);
1743         if (~isempty(findstr(char(it(i,:)), 'textWaitWiz')))
1744             set(wiz.(char(it(i,:))), 'Visible', 'off');
1745         end
1746     end
1747     if (~isempty(findstr(char(it(i,:)), 'var')))
1748         if inici==1
1749             set(wiz.(char(it(i,:))), 'Enable', 'off', 'String', '');

```



```

1750         end
1751     end
1752     if (~isempty(findstr(char(it(i,:)), 'button')))
1753         set(wiz.(char(it(i,:))), 'Enable', able);
1754     end
1755 end
1756 clear it
1757
1758 it=fieldnames(wiz.fig);
1759 for i=1:size(it,1)
1760     if (~isempty(findstr(char(it(i,:)), 'edit')))
1761         set(wiz.fig.(char(it(i,:))), 'Enable', able);
1762     end
1763     if (~isempty(findstr(char(it(i,:)), 'text')))
1764         set(wiz.fig.(char(it(i,:))), 'Enable', able);
1765         if (~isempty(findstr(char(it(i,:)), 'textWait')))
1766             set(wiz.fig.(char(it(i,:))), 'Visible', 'off');
1767         end
1768     end
1769     if (~isempty(findstr(char(it(i,:)), 'var')))
1770         if inici==1
1771             set(wiz.fig.(char(it(i,:))), 'Enable', 'off');
1772         end
1773     end
1774     if (~isempty(findstr(char(it(i,:)), 'popup')))
1775         set(wiz.fig.(char(it(i,:))), 'Enable', able);
1776     end
1777     if (~isempty(findstr(char(it(i,:)), 'button')))
1778         set(wiz.fig.(char(it(i,:))), 'Enable', able);
1779         if inici==1
1780             set(wiz.fig.buttonProcess, 'Visible', 'off')
1781         end
1782     end
1783     if (~isempty(findstr(char(it(i,:)), 'radio')))
1784         if inici==1
1785             set(wiz.fig.(char(it(i,:))), 'Enable', 'off', 'Value', 0);
1786         else
1787             set(wiz.fig.(char(it(i,:))), 'Enable', able)
1788         end
1789     end
1790     if (~isempty(findstr(char(it(i,:)), 'check')))
1791         if inici==1
1792             set(wiz.fig.(char(it(i,:))), 'Value', 0, 'Enable', able);
1793         else
1794             set(wiz.fig.(char(it(i,:))), 'Enable', able);
1795         end
1796     end
1797     if (~isempty(findstr(char(it(i,:)), 'entra')))
1798         if inici~=1
1799             set(wiz.fig.(char(it(i,:))), 'Enable', able);
1800         else
1801             set(wiz.fig.(char(it(i,:))), 'String', '');
1802         end
1803     end
1804     if (~isempty(findstr(char(it(i,:)), 'listbox')))
1805         if inici==1
1806             set(wiz.fig.(char(it(i,:))), 'string', '', 'Value', 1);
1807         end
1808     end
1809 end
1810

```

```

1811  if inici==1
1812
1813      cla(wiz.fig.axesPlot, 'reset');
1814      cla(wiz.fig.axesFix, 'reset');
1815      cla(wiz.fig.axesPolar, 'reset');
1816      cla(wiz.fig.axesPolarTots, 'reset');
1817      cla(wiz.fig.axesSignals_VSxAuxProfile, 'reset');
1818      cla(wiz.fig.axesxAux_VSsignalsProfile, 'reset');
1819      cla(wiz.fig.axesSignals_VSxAux, 'reset');
1820      cla(wiz.fig.axesxAux_VSsignals, 'reset');
1821      cla(wiz.fig.axesSignals_VSxAuxProfileRec, 'reset');
1822
1823      set([wiz.fig.axesPlot wiz.fig.axesFix wiz.fig.axesPolar ...
1824          wiz.fig.axesPolarTots wiz.fig.axesSignals_VSxAuxProfile ...
1825          wiz.fig.axesxAux_VSsignalsProfile ...
1826          wiz.fig.axesSignals_VSxAux wiz.fig.axesxAux_VSsignals...
1827          wiz.fig.axesSignals_VSxAuxRec ...
1828          wiz.fig.axesSignals_VSxAuxProfileRec], ...
1829          'Visible', 'off');
1830
1831      set(wiz.fig.panelTaula, 'Visible', 'off')
1832
1833      % Deshabilitem les opcions del gràfic
1834      zoom reset;
1835      datacursormode off
1836      zoom off
1837      pan off
1838      % Panels que s'amaguen:
1839      set([wiz.fig.uipanelDiscrimAnalysis wiz.fig.uipanelListboxes ...
1840          wiz.fig.uipanelCustNK wiz.fig.uipanelPlot ...
1841          wiz.fig.uipanelReconstruct wiz.fig.uipanelWaveletPT...
1842          wiz.fig.uipanelReconstructCustDict], ...
1843          'Visible', 'off');
1844
1845      % Les marques dels seus menús
1846      set([wiz.fig.menuLDB ...
1847          wiz.fig.menuWaveletPT], ...
1848          'Checked', 'off');
1849      set(wiz.fig.menuShowHidePlot, 'Label', 'Show plot menu');
1850
1851      if isfield(wiz.file, 'xsignals')
1852          wiz.file = rmfield(wiz.file, 'xsignals');
1853      end
1854      if isfield(wiz.file, 'ranswer')
1855          wiz.file = rmfield(wiz.file, 'ranswer');
1856      end
1857      if isfield(wiz.file, 'sJ')
1858          wiz.file = rmfield(wiz.file, 'sJ');
1859      end
1860
1861      set(wiz.wizardLDB, 'Position', wiz.finestraInic);
1862
1863      % Panel dels Botons:
1864      set(wiz.panelBotons, ...
1865          'Position', [wiz.panelsXicsAmplada ...
1866                      0 ...
1867                      (wiz.finestraInic(3)-wiz.panelsXicsAmplada) ...
1868                      wiz.alcadaPanelBotons]);
1869
1870      % Panel dels Botons:

```

```

1871 set(wiz.buttonPrev,...
1872     'Visible','off',...
1873     'Enable','on');
1874 set(wiz.buttonExit,...
1875     'Enable','on');
1876 set(wiz.buttonLoad,'Enable','on');
1877
1878 % Panel d'indicació
1879 set(wiz.panelIndicaLoad,'BackgroundColor',...
1880     get(0,'defaultUicontrolBackgroundColor'));
1881 set(wiz.panelIndicaConf,'BackgroundColor',...
1882     [0.5 0.5 0.5]);
1883 set(wiz.panelIndicaDepict,'BackgroundColor',...
1884     [0.5 0.5 0.5]);
1885
1886 set(wiz.txtIndicaLoad,'BackgroundColor',...
1887     get(0,'defaultUicontrolBackgroundColor'));
1888 set(wiz.txtIndicaConf,'BackgroundColor',...
1889     [0.5 0.5 0.5]);
1890 set(wiz.txtIndicaDepict,'BackgroundColor',...
1891     [0.5 0.5 0.5]);
1892
1893 set(wiz.txtIndicaLoadBuit, 'Visible','on');
1894 set(wiz.txtIndicaConfBuit, 'Visible','off');
1895 set(wiz.txtIndicaDepictBuit,'Visible','off');
1896
1897 % Fem un reset a axes2
1898 cla(wiz.axes2,'reset')
1899
1900 set(wiz.botoTots,'enable','on');
1901 set(wiz.botoCap,'enable','off');
1902 set(wiz.botoAfegeix,'enable','on');
1903 set(wiz.botoTreu,'enable','off');
1904
1905 wiz.esquerra=wiz.Complete_dLibrary; % Ho carreguem a 'esquerra'
1906 wiz.dreta={}; % Buidem 'dreta'
1907 % Última posició del listbox esquerre (màxims elements):
1908 wiz.ultimaposicioEsquerra=length(wiz.esquerra);
1909 % Última posició del listbox dret (màxims elements):
1910 wiz.ultimaposicioDreta=0;
1911
1912 set(wiz.listboxEntradaWiz,'String',wiz.Complete_dLibrary);
1913 set(wiz.listboxSortidaWiz,'String','');
1914
1915 % Inicialitzem la variable dels subespais
1916 fileGlobal.subspaceReconstr=(zeros(1,3));
1917
1918 end
1919 if esmod==1 % Si la finestra era modal i l'hem canviat
1920     set(wiz.wizardLDB,'Visible','on');
1921 else % si la finestra no era modal
1922     % Posa la finestra wiz.wizardLDB en primer terme
1923     uistack(wiz.wizardLDB, 'top');
1924 end
1925
1926 clear able bc i inici it mb mod esmod
1927 guidata(gcbo,wiz);
1928 end
1929
1930

```

```

1931 function FinishedWiz (varargin)
1932 global fileNameGlob
1933 % fileNameGlob la utilitzarem per saber si abans havíem carregat algun
1934 % fitxer o és la primera vegada que se'n carrega algun
1935 wiz=guidata(gcbo); % llegeix guidata.
1936 % A EnableDisableWiz() esborrarem els següents camps, així que els
1937 % desem en variables per utilitzar-los a continuació
1938 if isfield(wiz.file,'xsignals')
1939     xsignals = wiz.file.xsignals;
1940 end
1941 if isfield(wiz.file,'ranswer')
1942     ranswer = wiz.file.ranswer;
1943 end
1944 if isfield(wiz.file,'sJ')
1945     sJ = wiz.file.sJ;
1946 end
1947 % Netegem la interfície cridant la funció EnableDisableWiz.
1948 EnableDisableWiz(gcbo,'off',1) % Li enviem l'objecte que ha fet la
1949 % crida, l'string 'off' perquè sàpiga quin tipus de 'Visible' o
1950 % 'Enable' volem
1951
1952 % -----%
1953 % Modificació de la interfície inicial
1954 wiz=guidata(gcbo); % llegeix guidata.
1955
1956 set(wiz.fig.textWait,...
1957     'String','Please wait...',...
1958     'enable','on',...
1959     'Visible','on');
1960
1961 set(wiz.wizardLDB,'Visible','off');
1962 % Modifiquem la interfície
1963 set(wiz.fig.textSignalName,'enable','on')
1964 set(wiz.fig.varSignalName,'enable','on')
1965 % Posem el nom del senyal:
1966 set(wiz.fig.varSignalName, 'String', num2str(wiz.fileName));
1967
1968 % Canviem la variable xdata a xsignals, si existeix:
1969 % F = ISFIELD(S,'camp') torna true si 'camp' és el nom d'un camp en
1970 % l'estructura S:
1971 if exist('xsignals','var')
1972     set(wiz.fig.textNumberSignals,'enable','on');
1973     set(wiz.fig.varNumberSignals,'enable','on');
1974     set(wiz.fig.varNumberSignals, ...
1975         'String', num2str(size(xsignals,2)));
1976 end
1977
1978 if exist('sJ','var')
1979     % i posem a enable les caixes de text corresponents
1980     set(wiz.fig.textLevels,'enable','on');
1981     set(wiz.fig.varLevels,'enable','on');
1982     set(wiz.fig.varLevels, 'String', num2str(sJ));
1983 end
1984
1985 if isfield(wiz.file,'BCoefitients')
1986     % i posem a enable les caixes de text corresponents
1987     set(wiz.fig.textCoefficients,'enable','on');
1988     set(wiz.fig.varCoefficients,'enable','on');
1989     set(wiz.fig.varCoefficients,'String',...
1990         num2str(size(xsignals,2)*(sJ+1)));
1991 end

```

```

1992
1993 if exist('ranswer','var') ...
1994     && exist('xsignals','var') ...
1995     && exist('sJ','var')
1996     set(wiz.fig.textVectorClass,'enable','on');
1997     set(wiz.fig.varVectorClass,'enable','on');
1998     % 'i' serà un vector que valdrà 1,2,3... fins al valor final
1999     % de ranswer
2000     i=1:ranswer(end);
2001     set(wiz.fig.varVectorClass,'String',num2str(i(end)));
2002
2003     set(wiz.fig.textGroupName,'enable','on');
2004     set(wiz.fig.popupGroupName,'enable','on');
2005
2006     set(wiz.fig.popupGroupName,'String',i);
2007
2008     set(wiz.fig.textFirst,'enable','on');
2009     set(wiz.fig.editFirst,'enable','on');
2010     set(wiz.fig.textLast,'enable','on');
2011     set(wiz.fig.editLast,'enable','on');
2012
2013     i=get(wiz.fig.popupGroupName,'Value');
2014     ipos = find(ranswer==i);
2015
2016     % Posa el primer element de ipos a editFirst:
2017     set(wiz.fig.editFirst,'String',num2str(ipos(1)));
2018     % Posa l'últim element de ipos a editLast:
2019     set(wiz.fig.editLast,'String',num2str(ipos(end)));
2020
2021     set(wiz.fig.buttonView,'Enable','on');
2022 end
2023 if exist('xsignals','var') ...
2024     && exist('ranswer','var') ...
2025     && exist('sJ','var')
2026     set(wiz.fig.textLibType,'enable','on');
2027     set(wiz.fig.radioStd,'enable','on');
2028     set(wiz.fig.radioCust,'enable','on');
2029     p=get(wiz.fig.textExempleSeleccio,'Position');
2030     str=['Write numbers and intervals separated by comma. Ex: 1,3,5-
2031         12,14. Max range: 1-' ...
2032         num2str(size(wiz.file.Bspaces,1))]];
2033     [outstring pos] = textwrap(wiz.fig.textExempleSeleccio,str);
2034     set(wiz.fig.textExempleSeleccio,...
2035         'String',outstring,...
2036         'Enable','on',...
2037         'Position',[p(1)    ...
2038                     p(2)    ...
2039                     p(3)    ...
2040                     pos(4)]);
2041 end
2042
2043 % Panel Discriminant
2044 % Modifiquem el valor màxim de l'slider del nivell de descomposició
2045 % el valor que mostrarà i el pas ( en percentatge)
2046 set(wiz.fig.sliderNivellDescomp,...
2047     'Max',sJ,...
2048     'Value',sJ,...
2049     'SliderStep',[1/(sJ-1) 1/(sJ-1)]);
2050 set(wiz.fig.entraNivellDescomp,...
2051     'String',sJ);
2052 % Cridem la funció RadioStdFig o RadioCustFig per ensenyar directament

```

```

2053 % els seus listboxs i el seu contingut correctament
2054 if ~exist('fileNameGlob','var')
2055     % Si no existeix la variable és que no hem carregat cap fitxer
2056     % i assignem 'Std' a fig.StdOrCust
2057     wiz.fig.StdOrCust='Std';
2058     wiz.fig.esquerra=wiz.esquerra;
2059     wiz.fig.Complete_dLibrary=wiz.Complete_dLibrary;
2060 end
2061 if strcmp(wiz.fig.StdOrCust,'Std')
2062     guidata(gcbo,wiz) % Desem els canvis fets fins ara
2063     RadioStdFig
2064     wiz=guidata(gcbo); % llegeix els canvis acabats de fer
2065 else
2066     guidata(gcbo,wiz) % Desem els canvis fets fins ara
2067     RadioCustFig
2068     wiz=guidata(gcbo); % llegeix els canvis acabats de fer
2069 end
2070 set(wiz.fig.uipanelDiscrimAnalysis,'Visible','off');
2071 set(wiz.fig.menuLDB,'Checked','off');
2072 % Panel WaveletPT
2073 % Modifiquem el valor màxim de l'slider del nivell de descomposició
2074 % el valor que mostrarà i el pas ( en percentatge)
2075 set(wiz.fig.sliderNivellDescompWPT,...
2076     'Max',sJ,...
2077     'Value',sJ,...
2078     'SliderStep',[1/(sJ-1) 1/(sJ-1)]);
2079 set(wiz.fig.entraNivellDescompWPT,...
2080     'String',sJ);
2081 % i posem la cadena wiz.Complete_dLibrary al seu listBox
2082 set(wiz.fig.listBoxFEWPT,'String',wiz.Complete_dLibrary);
2083 set(wiz.fig.buttonProcessWPT,'enable','on');
2084
2085 % Panel Plot reconstruct
2086
2087 a=wiz.file.D_Ranking(:,1);
2088 val=get(wiz.popupLibrary,'Value'); %Obtenim el nom de la llibreria
2089 set(wiz.fig.popupLibrary,'String',a,'Enable','on','Value',val);
2090
2091 set(wiz.fig.radioSelect,'Enable','on');
2092
2093 set(wiz.fig.textExempleSeleccio,'Enable','on');
2094 set(wiz.fig.buttonViewRec,'Enable','on');
2095 set(wiz.fig.buttonRemoveRec,'Enable','on');
2096 set(wiz.fig.textLibraryPop,'Enable','on');
2097
2098 set(wiz.fig.uipanelReconstruct,...
2099     'Visible','on',... % i farà visible el uipanelReconstruct
2100     'Position',[wiz.fig.panelsX ...
2101     (wiz.fig.panelPlotY-wiz.fig.panelSignalAlcada) ...
2102     wiz.fig.panelsAmplada ...
2103     wiz.fig.panelPlotAlcada]);
2104
2105
2106 if exist('xsignals','var')
2107     if strcmp(wiz.ContourOrProfile,'Profile')
2108         set(wiz.fig.menuPolar,'checked','off');
2109
2110         FuncPlotWiz(wiz.fig.axesSignals_VSxAuxProfile,...
2111             'Original signal',...
2112             'Profile',...
2113             xsignals,answer)

```

```

2114         FuncPlotWiz(wiz.fig.axesxAux_VSsignalsProfile,...
2115                     'Signal reconstructed',...
2116                     'Profile',...
2117                     wiz.file.xAux_Signal,answer)
2118     else
2119         set(wiz.fig.menuPolar,'checked','on');
2120         % Farà que es vegi correctament el caràcter _:
2121         str = strrep(wiz.fileName, '_', '\_');
2122         str=[str ' - All bases']; % Hi afegim 'Totes les parts'
2123
2124         FuncPolarWiz(wiz.fig.axesPolarTots,...
2125                     str,...
2126                     xsignals,answer)
2127
2128         FuncPlotWiz(wiz.fig.axesSignals_VSxAux,...
2129                     'Original signal',...
2130                     'Contour',...
2131                     xsignals,answer)
2132         FuncPlotWiz(wiz.fig.axesxAux_VSsignals,...
2133                     'Signal reconstructed',...
2134                     'Contour',...
2135                     wiz.file.xAux_Signal,answer)
2136     end
2137 end
2138
2139 % -----%
2140 set(wiz.fig.textWait,...
2141     'String','Please wait...',...
2142     'Visible','off')
2143
2144 clear str
2145
2146 guidata(gcbo,wiz);
2147 end
2148
2149
2150 function vect=ComparaCadenaWiz (varargin)
2151 wiz=guidata(gcbo); % llegeix guidata.
2152 vect=[];
2153 pM=[];
2154 k=1;
2155 % Cas i=1, quan comencem;
2156 % Desem en una variable P_1 la cadena fins a la primera coma i
2157 % la resta a remain
2158 eval(sprintf(['P_%d remain']=strtok(varargin{1},',','');',k));
2159 % busquem dins de la variable P_1 si hi ha un guió
2160 eval(sprintf(['f=findstr(P_%d,\'-\');',k]));
2161
2162 if isempty(f) % Si no és un interval (un interval: 1-5)
2163     eval(sprintf(['a]=(P_%d);',k));
2164     eval(sprintf(['a=str2num(a);']));
2165     if isscalar(a) && mod(a,1)==0
2166         if a>=1
2167             vect=eval(sprintf(['a;']));
2168             eval(sprintf(['b_old=a;']));
2169         end
2170     else
2171         set(wiz.textWaitWiz,...
2172             'String','Enter a valid numeric value.',...
2173             'fontSize',10,...
2174             'Visible','on');

```

```

2175         pause(1.5)
2176         set(wiz.textWaitWiz,...
2177             'String','Wait...',...
2178             'fontSize',20,...
2179             'Visible','off');
2180         % El vector final el posarem a 0 perquè no
2181         % processi
2182         vect=[];
2183         return % Farem que surti del bucle
2184     end
2185
2186 else % Si és un interval
2187     eval(sprintf(['a remainGuio]=strtok(P_%d,'-');',k));
2188     eval(sprintf(['b remainGuio]=strtok(remainGuio,'-');']));
2189     eval(sprintf('a=str2num(a);'));
2190     eval(sprintf('b=str2num(b);'));
2191     if isscalar(a) && isscalar(b) && mod(a,1)==0 && mod(b,1)==0
2192         if b>=1 && a>=1
2193             if b>a
2194                 vect=eval(sprintf(['a:b;']));
2195                 eval(sprintf('b_old=b;'));
2196             else
2197                 if b==a
2198                     vect=eval(sprintf(['b;']));
2199                 else
2200                     vect=eval(sprintf(['b:a;']));
2201                 end
2202                 eval(sprintf('b_old=a;'));
2203             end
2204         else
2205             eval(sprintf('b_old=0;'));
2206         end
2207     else
2208         set(wiz.textWaitWiz,...
2209             'String','Enter a valid numeric value.',...
2210             'fontSize',10,...
2211             'Visible','on');
2212         pause(1.5)
2213         set(wiz.textWaitWiz,...
2214             'String','Wait...',...
2215             'fontSize',20,...
2216             'Visible','off');
2217         % El vector final el posarem a 0 perquè no
2218         % processi
2219         vect=[];
2220         return % Farem que surti del bucle
2221     end
2222
2223 end
2224
2225 k=k+1;
2226
2227 while remain~=0
2228     if remain(1)==' '
2229         eval(sprintf(['P_%d remain]=strtok(remain,' ','');',k));
2230         eval(sprintf(['f=findstr(P_%d,'-');',k]));
2231         if isempty(f) % Si no és un interval
2232             eval(sprintf(['a]=(P_%d);',k));
2233             eval(sprintf('a=str2num(a);'));
2234
2235             vect=eval(sprintf(['vect a;']));

```



```

2236         if isscalar(a) && mod(a,1)==0
2237             if a>b_old && a>=1
2238                 eval(sprintf('b_old=a;'));
2239             end
2240         else
2241             set(wiz.textWaitWiz,...
2242                 'String','Enter a numeric value.',...
2243                 'fontSize',10,...
2244                 'Visible','on');
2245             pause(1.5)
2246             set(wiz.textWaitWiz,...
2247                 'String','Wait... ',...
2248                 'fontSize',20,...
2249                 'Visible','off');
2250             % El vector final el posarem a 0 perquè no
2251             % processi
2252             vect=[];
2253             return % Farem que surti del bucle
2254         end
2255     else % si és un interval
2256         eval(sprintf(['a remainGuio]=strtok(P_%d, '-');',k));
2257         eval(sprintf(['b remainGuio]=strtok(remainGuio, '-');']);
2258         eval(sprintf('a=str2num(a);'));
2259         eval(sprintf('b=str2num(b);'));
2260         if isscalar(a) && isscalar(b) && mod(a,1)==0 && mod(b,1)==0
2261             if b>=1 && a>=1
2262                 if b>a
2263                     vect=eval(sprintf(['vect a:b;']));
2264                     eval(sprintf('b_old=b;'));
2265                 else
2266                     if b==a
2267                         vect=eval(sprintf(['vect b;']));
2268                     else
2269                         vect=eval(sprintf(['vect b:a;']));
2270                     end
2271                     eval(sprintf('b_old=a;'));
2272                 end
2273             end
2274         end
2275     else
2276         set(wiz.textWaitWiz,...
2277             'String','Enter a numeric value.',...
2278             'fontSize',10,...
2279             'Visible','on');
2280         pause(1.5)
2281         set(wiz.textWaitWiz,...
2282             'String','Wait... ',...
2283             'fontSize',20,...
2284             'Visible','off');
2285         % El vector final el posarem a 0 perquè no
2286         % processi
2287         vect=[];
2288         return % Farem que surti del bucle
2289     end
2290 end
2291 end
2292 k=k+1;
2293 end
2294
2295 if ~isempty(vect)
2296     vect=sort(vect);

```

```

2297     pM=vect;
2298     % Recollim els camps majors a la matriu:
2299     pM=pM(ismember(pM,1:size(wiz.file.xsignals,1))==0);
2300     % Suprimim els camps majors a la matriu:
2301     vect(ismember(vect,1:size(wiz.file.xsignals,1))==0)=[];
2302
2303     j=length(vect);
2304     for i=1:j
2305         if j>=i % evitem repeticions
2306             k = findstr(vect, vect(:,i));
2307             while length(k)>1
2308                 if j>=i
2309                     vect(:,k(end))=[];
2310                     k(:,(end))=[];
2311                     j=j-1;
2312                 end
2313             end
2314         end
2315     end
2316 end
2317
2318 if ~isempty(pM) % Si la dada entrada supera la mida de la matriu...
2319     str={'Maximum Range: 1-' ...
2320         num2str(size(wiz.file.Bspaces,1)) ' .Vector changed'}};
2321
2322     set(wiz.textWaitWiz,...
2323         'String',str,...
2324         'fontSize',10,...
2325         'Visible','on');
2326
2327     pause(0.5)
2328     str=['1-' num2str(size(wiz.file.Bspaces,1))];
2329     set(wiz.fig.editSelect,...
2330         'string',str);
2331     set(wiz.textWaitWiz,...
2332         'String','Wait',...
2333         'fontSize',20,...
2334         'Visible','off');
2335 end
2336 end
2337
2338
2339 function FuncPlotWiz (figura,titol,contour,varargin)
2340 wiz=guidata(gcbo); % llegeix guidata.
2341
2342 senyal=varargin{1};
2343 ranswer=varargin{2};
2344
2345 if nargin==6
2346     labelX=varargin{3};
2347     contour='Profile';
2348 else
2349     labelX='';
2350 end
2351 set(figura,'Visible','on');
2352 axes(figura);
2353
2354 % Array de caràcters amb els colors de línia:
2355 % El blanc '-w' no li posarem perquè no es veuria amb el fons
2356 % No hi posem el vermell perquè l'usarem per altres coses

```

```

2357 linia=['-b';'-g';'-c';'-m';'-y';'-k'];
2358 j=1; % índex per llegir aquest array
2359
2360 % per què sàpiga quan arribem al final del vector ranswer:
2361 fi=ranswer(:,end);
2362
2363 hold on
2364
2365 if strcmp(contour,'Contour')
2366 y=0:360/(size(senyal,1)-1):360;
2367 else
2368 y=0:(size(senyal,1))/(size(senyal,1)-1):(size(senyal,1));
2369 end
2370
2371 for i=1:size(senyal,2)
2372
2373     plot(y,senyal(:,i),linia(j,:));
2374     if ranswer(:,i)~=fi % Si no estem al final
2375         % Compararem la posició de ranswer amb la següent per
2376         % saber si canvia
2377         if ranswer(:,i)<ranswer(:,i+1)
2378             j=j+1; % hi afegeix 1 per agafar el següent color
2379         end
2380     end
2381 end
2382
2383 title(titol,'Color','w');
2384 xlabel(labelX,'Color','w');
2385
2386 if strcmp(contour,'Contour')
2387     set(figura,'XTick',0:360/12:360) % Parts en que dividirà l'eix
2388 else
2389     set(gca,'XTick',0:(size(senyal,1))/8:(size(senyal,1)))
2390 end
2391
2392 grid on
2393 zoom reset;
2394 axis tight
2395
2396 guidata(gcbo,wiz);
2397 end
2398
2399
2400 function FuncPolarWiz (figura,titol,varargin)
2401
2402 senyal=varargin{1};
2403 ranswer=varargin{2};
2404 if nargin==3
2405     labelX=varargin{3};
2406 else
2407     labelX='';
2408 end
2409 set(figura,'Visible','on');
2410 axes(figura);
2411
2412 % Array de caràcters amb els colors de línia:
2413 % El blanc '-w' no li posarem perquè no es veuria amb el fons
2414 % No hi posem el vermell perquè l'usarem per altres coses
2415 linia=['-b';'-g';'-c';'-m';'-y';'-k'];
2416 j=1; % índex per llegir aquest array

```

```

2417
2418 % per què sàpiga quan arribem al final del vector ranswer:
2419 fi=ranswer(:,end);
2420
2421
2422 angle=0:2*pi/(size(senyal,1)-1):2*pi;
2423 % angle=-angle; % Si no canviéssim la diecció dels eixos hauríem de
2424 % canviar la direcció de l'angle.
2425
2426 polar(angle',senyal(:,1),linia(j,:));
2427 hold on
2428 % Si ens trobéssim amb només dues mostres hauria de canviar ja
2429 % de color:
2430 if ranswer(:,1)~=fi % Si no tenim una sola mostra
2431     % Compararem la posició de ranswer amb la següent per
2432     % saber si canvia
2433     if ranswer(:,1)<ranswer(:,2)
2434         j=j+1; % hi afegeix 1 per agafar el següent color
2435     end
2436 end
2437
2438 for i=1:size(senyal,2)
2439     polar(angle',senyal(:,i),linia(j,:));
2440
2441     if ranswer(:,i)~=fi % Si no estem al final
2442         % Compararem la posició de ranswer amb la següent per
2443         % saber si canvia
2444         if ranswer(:,i)<ranswer(:,i+1)
2445             j=j+1; % hi afegeix 1 per agafar el següent color
2446         end
2447     end
2448 end
2449
2450 title(titol,'Color','w');
2451 xlabel(labelX,'Color','w');
2452
2453 set(figura,'YDir','reverse') % canviem la direcció dels eixos
2454
2455 grid on
2456
2457 zoom reset;
2458
2459 end
2460
2461 function RadioStdFig (varargin)
2462 global fileGlobal
2463 wiz=guidata(gcbo); % llegeix guidata.
2464
2465 % Indiquem quin radioButton està actiu
2466 set(wiz.fig.radioStd,'enable','on','Value',1)
2467 set(wiz.fig.radioCust,'enable','on','Value',0)
2468
2469 wiz.fig.StdOrCust='Std'; % Desem en una variable el valor 'Std'
2470
2471
2472 % Obtenim les mides i posició de uipanelCustNK i de uipanelListboxes
2473 p=get(wiz.fig.uipanelCustNK,'Position');
2474 % u=get(wiz.fig.uipanelListboxes,'Position');
2475 u=wiz.fig.panelDiscrimAnalysisMenuAlcada/2-10; % Alçada
2476 % Alçada del panel uipanelListboxes:

```

```

2477 v=u+p(4)+10; % alçada uipanelCustNK + de uipanelListboxes + marge
2478
2479 set(wiz.fig.uipanelNivellDescomp, 'Visible', 'on');
2480
2481 set(wiz.fig.uipanelCustNK, 'Visible', 'off')
2482 set(wiz.fig.uipanelListboxes, ...
2483     'Visible', 'on', ...
2484     'Position', [wiz.fig.margeP ...
2485                 (p(2)-wiz.fig.panelDiscrimAnalysisMenuAlcada/2) ...
2486                 wiz.fig.midaAmpPanInt ...
2487                 v ...
2488                 ])
2489 % Obtenim les mides i posició de uipanelListboxes:
2490 p=get(wiz.fig.uipanelListboxes, 'Position');
2491
2492 mee=3; % Marge entre elements interns: ListBoxes-mee-botons
2493 mde=6; % Marge dret, esquerre, dalt i baix entre listboxes i panel.
2494 mbp=16; % Amplada del botó petit, botoAfegeix i botoTreu
2495 % Fent una petita equació per trobar l'amplada dels Listboxes:
2496 % midaPanelAmplada=mde+midaListBoxes+mee+mbp+mee+midaListBoxes+mde
2497 % midaPanelAmplada=2*mde+2*midaListBoxes+2*mee+mbp ...
2498 % midaListBoxes=(midaPanelAmplada-mbp-2*(mde+mee))/2
2499 ampLB=(p(3)-mbp-2*(mde+mee))/2; % Amplada de ListBoxes
2500
2501 abT=14; % Alçada dels botons botoTots i botoCap
2502 alcLB=(p(4)-2*mde-abT-mee*4); % Alçada dels Listboxs
2503
2504 set(wiz.fig.listboxEntrada, ...
2505     'Position', [mde ...
2506                 (mde+abT+mee) ...
2507                 ampLB ...
2508                 alcLB ], ...
2509     'String', wiz.fig.esquerra, ... % Mostrarem el que hi havia
2510     'Visible', 'on')
2511 set(wiz.fig.listboxSortida, ...
2512     'Position', [(mde+ampLB+2*mee+mbp) ...
2513                 (mde+abT+mee) ...
2514                 ampLB ...
2515                 alcLB ], ...
2516     'String', wiz.fig.dreta, ...
2517     'Visible', 'on')
2518
2519 % Pels Listboxes del Filter Menu
2520
2521 set(wiz.fig.botoTots, ...
2522     'Position', [mde ...
2523                 mde ...
2524                 ampLB ...
2525                 abT ], ...
2526     'Visible', 'on');
2527 set(wiz.fig.botoCap, ...
2528     'Position', [(mde+ampLB+2*mee+mbp) ...
2529                 mde ...
2530                 ampLB ...
2531                 abT ], ...
2532     'Visible', 'on');
2533 set(wiz.fig.botoAfegeix, ...
2534     'Position', [(mde+ampLB+mee) ...
2535                 ((p(4)+mde+abT)/2+mee/2) ...
2536                 mbp ...
2537                 alcLB/3 ], ...

```

```

2538     'Visible','on');
2539 set(wiz.fig.botoTreu,...
2540     'Position',[mde+ampLB+mee           ...
2541     ((p(4)+mee+abT)/2-alcLB/3-mee/2) ...
2542     mbp           ...
2543     alcLB/3       ],...
2544     'Visible','on');
2545
2546 if ~isempty(wiz.fig.esquerra)
2547     set(wiz.fig.botoTots,...
2548         'Enable','on');
2549     set(wiz.fig.botoAfegeix,...
2550         'Enable','on');
2551 else
2552     set(wiz.fig.botoTots,...
2553         'Enable','off');
2554     set(wiz.fig.botoAfegeix,...
2555         'Enable','off');
2556 end
2557
2558 if isempty(wiz.fig.dreta)
2559     set(wiz.fig.botoCap,...
2560         'Enable','off');
2561     set(wiz.fig.botoTreu,...
2562         'Enable','off');
2563
2564     set(wiz.fig.buttonProcess,...
2565         'Visible','on',...
2566         'Enable','off')
2567 else
2568     set(wiz.fig.botoCap,...
2569         'Enable','on');
2570     set(wiz.fig.botoTreu,...
2571         'Enable','on');
2572
2573     set(wiz.fig.buttonProcess,...
2574         'Visible','on',...
2575         'Enable','on')
2576 end
2577
2578
2579 if isfield(fileGlobal,'xAux_Signal') % Si ja haviem processat...
2580     set(wiz.fig.uipanelDiscrimAnalysis,...
2581         'Visible','on');
2582     set(wiz.fig.menuLDB,'Checked','on');
2583 end
2584
2585
2586 % Netegem les variables que ja no necessitarem
2587 clear str c Complete_dLibrary
2588
2589 guidata(gcbo,wiz)           % Desem els canvis fets
2590 end
2591
2592
2593 function RadioCustFig (varargin)
2594
2595 wiz=guidata(gcbo); % llegeix guidata.
2596
2597 set(wiz.fig.radioStd,'enable','on','Value',0)

```

```

2598 set(wiz.fig.radioCust,'enable','on','Value',1)
2599
2600 wiz.fig.StdOrCust='Cust';
2601
2602 % Pels Listboxes del Filter Menu
2603
2604 set(wiz.fig.uipanelCustNK,...
2605     'Visible','on')
2606 % Obtenim les mides i posició de uipanelCustNK
2607 p=get(wiz.fig.uipanelCustNK,'Position');
2608 set(wiz.fig.uipanelListboxes,...
2609     'Visible','on',...
2610     'Position',[wiz.fig.margeP
2611                 (p(2)-wiz.fig.panelDiscrimAnalysisMenuAlcada/2) ...
2612                 wiz.fig.midaAmpPanInt
2613                 wiz.fig.panelDiscrimAnalysisMenuAlcada/2-10    ])
2614 set(wiz.fig.buttonProcess,'Visible','on')
2615 set(wiz.fig.entraN,'Enable','on');
2616 set(wiz.fig.entraK,'Enable','on');
2617 wiz.fig.esquerra={};
2618 wiz.fig.dreta={};
2619 % Obtenim les mides i posició de uipanelListboxes
2620 p=get(wiz.fig.uipanelListboxes,'Position');
2621
2622 mde=6; % Marge dret, esquerre, dalt i baix entre listboxes i panel.
2623
2624 alcLB=(p(4)-2*mde); % Alçada dels Listboxs
2625 ampLBCust=p(3)-2*mde;
2626
2627 set(wiz.fig.listboxEntrada,...
2628     'Position',[mde
2629                 mde
2630                 ampLBCust
2631                 alcLB ],...
2632     'String',wiz.fig.esquerra)% No mostrem
2633 set(wiz.fig.listboxSortida,'Visible','off'); % No mostrem
2634 set(wiz.fig.botoTots,'Visible','off'); % No mostrem
2635 set(wiz.fig.botoCap,'Visible','off'); % No mostrem
2636 set(wiz.fig.botoAfegeix,'Visible','off'); % No mostrem
2637 set(wiz.fig.botoTreu,'Visible','off'); % No mostrem
2638 set(wiz.fig.buttonProcess,'enable','off')
2639
2640 % Agafem les dades del panel entraN i entraK
2641 wiz.fig.N=str2double(get(wiz.fig.entraN,'String'));
2642 wiz.fig.K=str2double(get(wiz.fig.entraK,'String'));
2643
2644 % Ja podem activar el botó buttonCreateComb
2645 set(wiz.fig.buttonCreateComb,'enable','on');
2646 set(wiz.fig.botoAfegeix,'enable','off'); %Aquests els desactivem
2647 set(wiz.fig.botoTots,'enable','off');
2648 set(wiz.fig.botoCap,'enable','off');
2649
2650 % Fem visible el panel uipanelNivellDescomp
2651 set(wiz.fig.uipanelNivellDescomp,'visible','on');
2652 set(wiz.fig.uipanelDiscrimAnalysis,...
2653     'Visible','on');
2654 set(wiz.fig.menuLDB,'Checked','on');
2655
2656 guidata(gcbo,wiz) % Desem els canvis fets
2657 end

```