Exemples d'utilisation du logiciel de traitement interactif MUSTIG.

Joël LIENARD - Gérard LEJEUNE

CEPHAG, INPG/IEG, URA 346 CNRS, BP 46, 38402 St Martin d'Hères Cedex

RÉSUME

Mustig est un logiciel interactif à entrée graphique pour le traitement des signaux à une ou plusieurs dimensions. Un traitement y est décrit par un "langage" graphique basé sur la notion de graphe de dépendance. Le logiciel possède un éditeur graphique simple et puissant pour construire et modifier le graphe du traitement. A l'aide de divers exemples nous montrons les différentes façons d'utiliser le langage pour définir le traitement : diagramme de dépendance, diagramme de flux, relations mathématiques entre fonctions ou entre vecteurs ou matrices, ainsi que son aptitude à décrire des traitements de signaux multidimensionnels.

SUMMARY

Mustig is a graphical and interactive software for the analysis of signals, either mono or multidimensional. A given treatment is implemented using a graphical "language" based on concept of dependency graph. The software includes a graphical editor, simple and powerful, for building and modifying the graph. Using several exemples, we show haw to use the language to define the operations (dependency diagram, flux diagram, mathematical relations between functions or between vectors and matrices) and to extend the treatment to multidimentional signals.

Introduction

Le principe de ce logiciel à été présenté lors du précédent colloque GRETSI (J.Liénard. Langage conversationnel pour le traitement des multi-signaux). Rappelons ses caractéristiques principales :

Le traitement est décrit par un **diagramme de dépendance** entre signaux. L'utilisateur n'a pas a se préoccuper de la gestion de la mémoire de travail.

Il est adapté au traitement **interactif**. Après une modification du graphe, les résultats des calculs précédents sont réutilisés, seuls les calculs nécessités par la modification sont exécutés. Ceci assure un fonctionnement **rapide**. De plus, les résultats affichés (courbes ou valeurs numériques) correspondent toujours au graphe actuel, indépendamment de l'historique de sa mise au point. Ceci assure un fonctionnement **sûr.**

C'est un logiciel **ouvert** : il n'est pas limité à l'utilisation de méthodes prédéfinies. Tout nouveau traitement peut être défini facilement de façon interactive. Des sous-programmes externes peuvent être appelés.

Le langage est prévu pour le traitement de signaux **multidimensionnels**. Les signaux manipulés peuvent être des fonctions d'un nombre quelconque de variables (temps "t", espace "x,y",temps-espace "t,x,y"...). La syntaxe permet une extension très rapide à plusieurs dimensions d'un traitement prévu pour un signal à une seule dimension.

Le traitement est décrit de façon **graphique** : le diagramme de dépendance est véritablement dessiné sur l'écran d'un micro-ordinateur. Pour cela, le logiciel possède un éditeur graphique simple et puissant pour construire le graphe du traitement, le modifier, le déformer, créer des macro-fonctions, examiner le contenu des macro-fonctions, éditer les valeurs numériques ou du texte de commentaire. Il possède également un dispositif de mise en page qui permet de réaliser très rapidement des feuilles de résultats regroupant des parties du graphe de description du traitement et les résultats associées.

Le logiciel fonctionne actuellement sur un micro-ordinateur MAC II, seul ou relié à un calculateur plus puissant auquel il sous-traite les calculs.

Nous nous proposons ici,après avoir exposé les principes du langage graphique, de démontrer ses possibilités à l'aide d'exemple .

2.Le langage graphique de MUSTIG

Le "langage" utilisé sous MUSTIG peut être appréhendé sous plusieurs formes. La forme de base est celle d'un diagramme de dépendance, complété par une syntaxe permettant de ne décrire qu'un seul motif d'un diagramme périodique en formant des "paquets" et de règles permettant d'allèger la

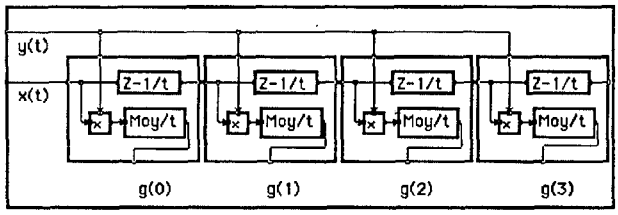
représentation. Des sous-ensembles du graphe peuvent être compactés en "macro". Dès que l'on possède un ensemble suffisant de ces macros (celle de la bibliothèque standard par exemple), il est possible d'oublier la forme de base pour utiliser une forme "fonctionnelle" beaucoup plus proche de l'écriture mathématique. La définition de la macro "retard" (z^{-1}) permet de décrire les structures de filtres sous la forme diagramme de flux. Enfin, il est toujours possible concevoir un traitement sous forme séquentielle, plus classique, utile en particulier pour des applications temps réel.

Toutes ces formes ne sont en fait que des approches différentes du langage. Le code de calcul fabriqué par le compilateur en partant de l'une ou de l'autre forme sera souvent le même. Il n'y a aucun inconvénient à mélanger les formes dans un même traitement.

3.Exemples d'utilisation.

3.1 Corrélateur.

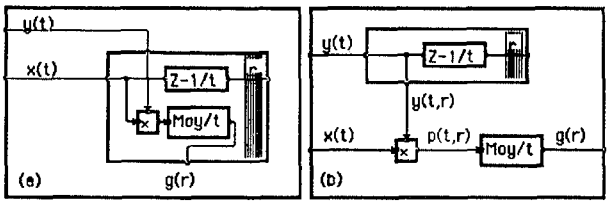
On sait que l'estimation de la fonction d'intercorrélation g(r) de deux signaux x(t) et y(t) peut se faire par moyennage sur une durée T du produit x(t).y(t-r). On obtient y(t-r) en retardant y(t) de la quantité r. En système échantillonné, le retard élémentaire est un retard d'un pas de temps noté souvent "z^{-1}". Nous obtenons alors le schéma suivant d'un corrélateur, calqué directement sur une structure électronique, estimant la fonction g(r) pour quatre valeurs de r.

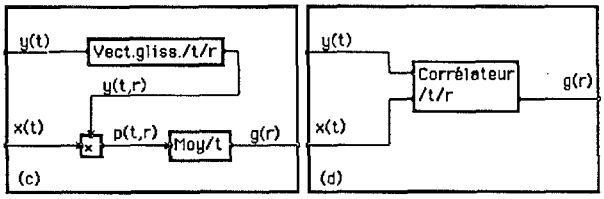


Il suffira de dessiner une seule cellule retard-produit-moyenne et d'indiquer sa répétition par la syntaxe "paquet" (fig. (a) ci-dessous). Pour comprendre la notation, il faut imaginer que tous les motifs identiques du graphe ont étés dessinés sur des rectangles de carton, puis que tous ces motifs ont été rassemblés en un paquet comme des cartes à jouer. Le trait épaissi représente alors schématiquement l'épaisseur du paquet.

Ce diagramme est équivalent à (b) ou les opérateurs produit et moyenne ont étés sortis du paquet. En effet la liaison qui arrive à la borne supérieure de l'opérateur produit représente l'ensemble des signaux de chacun des motifs du paquet, indicés par "r". C'est donc une fonction de "t" et de "r". Les opérateurs "produit" et "moyenne" ne connaissant pas cette variable "r" vont opérer sur chacune de ses valeurs.

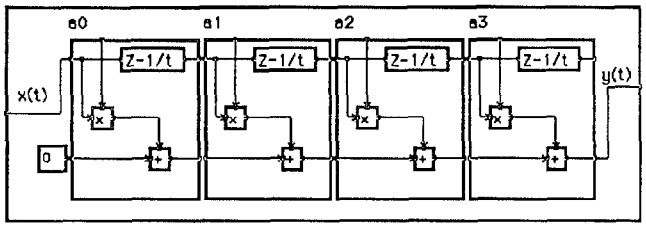
Il ne reste plus dans le paquet qu'un opérateur "z^{-1}". Ce paquet représente un registre à décalage. On peut considérer sa sortie comme un vecteur glissant le long du signal y(t) et comportant les N dernières valeurs de y(t). En condensant ce paquet en une macro, on obtient en (c) un diagramme uniquement fonctionnel. Enfin, ce diagramme peut lui-même être compacté en une macro "corrélateur" (d).



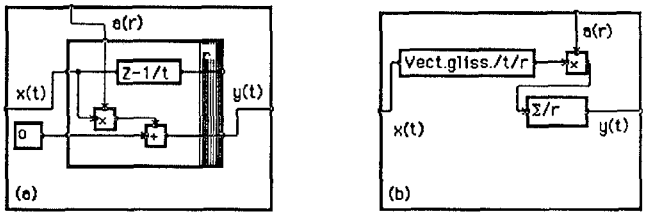


3.2 Filtre transversal.

Partons de la forme "diagramme de flux" bien connue du filtre, dessinée ici pour un ordre 4.



Comme précédemment, nous pouvons former un paquet (a), puis faire apparaître les opérateurs "vecteur glissant" et "somme" (b).



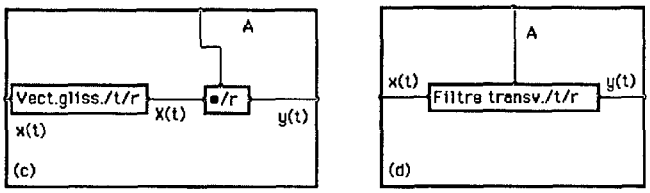
Nous obtenons une forme qui traduit directement l'équation du filtre:

$$y(t)=\Sigma\ x(t-r).a(r)$$

En appelant $\mathbf{X}(t)$ le vecteur glissant sur x(t) et $\mathbf{A}$ le vecteur des coefficients on peut écrire cette équation sous forme d'un produit scalaire de deux vecteurs :

$$y(t)=\mathbf{X}(t)\cdot\mathbf{A}$$

On peut se rapprocher de cette écriture en compactant les modules "produit" et "somme" en une macro "produit scalaire"(c).

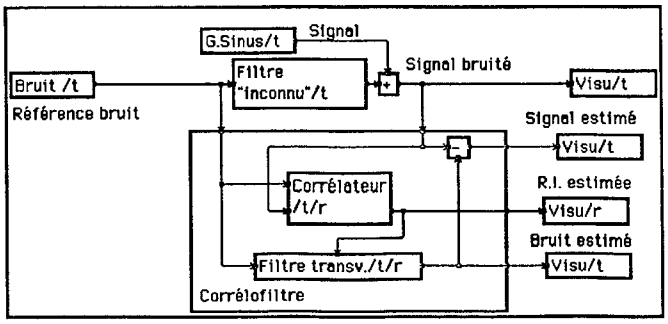


L'un quelconque des graphes a,b ou c peut être compacté en une macro "filtre transversal" (d).

3.3 Corrélo-filtre.

a) en deux passes

On appelle corrélo-filtre l'assemblage d'un corrélateur, qui identifie la réponse impulsionnelle d'un filtre inconnu par intercorrélation entrée-sortie, et d'un filtre transversal, qui utilise comme coefficients le réponse estimée par le corrélateur et réalise donc un filtre identique au filtre inconnu (aux erreurs d'estimation près). Ceci permet d'estimer un bruit perturbateur puis, par soustraction, le signal perturbé par le bruit. Nous pouvons utiliser sans aucune modification les deux macros "corrélateur" et "filtre transversal" précédentes pour former le corrélofiltre. Nous le mettons en oeuvre à l'aide des macros générateurs de signaux et visualisations disponibles en bibliothèque.



Notons que le calcul s'effectue en deux étapes. L'estimation de la fonction de corrélation se fait par intégration sur toute la durée du signal (indiqué par la définition de la variable "t"), puis le signal est ensuite filtré. Mais cette chronologie n'est pas indiquée explicitement : le compilateur la déduit automatiquement du diagramme de dépendance.

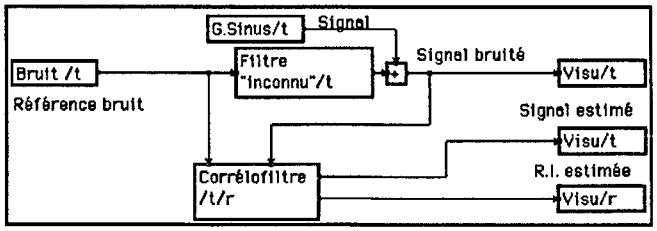
b) à moyenne exponentielle

La structure précédente ne convient que pour des signaux à durée limitée et stationnaires. Pour traiter des signaux non stationnaires, il faut réaliser un moyennage sur une fenêtre glissante. Le plus simple est de choisir une fenêtre exponentielle, en remplaçant le moyenneur du corrélateur par un filtre passe-bas de grande constante de temps. Aucune autre modification n'est nécessaire pour transformer le diagramme du corrélo-filtre en deux passes en corrélo-filtre à moyennage exponentiel. Le filtre utilise l'estimation évolutive de la corrélation, le filtrage peut alors être simultané à l'estimation de la corrélation. Le compilateur génére une seule boucle de calcul pour ces deux fonctions. Le fonctionnement temps réel est possible.

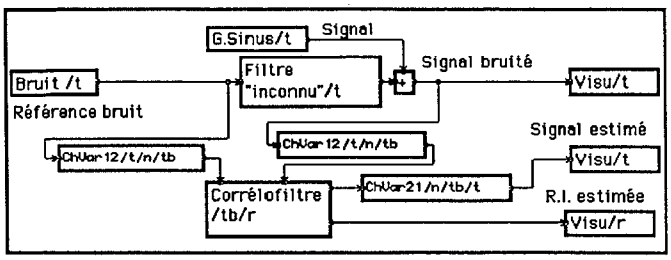
Il est toujours possible d'observer l'estimation de la réponse impulsionnelle, mais c'est maintenant une fonction des deux variables "t" et "r". En utilisant un module de visualisation par courbes décalées, on observe l'évolution de cette estimation au cours du temps. Dans l'exemple de simulation ci-dessous, la non stationnarité du filtre consiste en un retard introduit à partir de la deuxième moitié du temps.

c)calcul par blocs

Une autre façon de traiter le cas de signaux stationnaires est de décomposer les signaux en blocs juxtaposés et de les traiter un par un. La décomposition en blocs se fait, dans la version actuelle du logiciel, par un opérateur de changement de variables qui transpose un signal dépendant d'une variable "t" en un signal dépendant de deux variables "tb" et "n". "tb" devient l'indice à l'intérieur des blocs et "n" le numéro des blocs. Repartons du graphe du §a où le corrélofiltre est compacté en une macro :

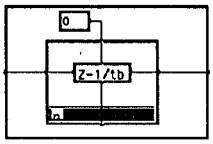


Il suffit d'appliquer au corrélofiltre des signaux transposés à deux variables et de faire la transposition inverse en sortie.



Nous utilisons les potentialités du langage pour les signaux multidimensionnels : le corrélofiltre travaille suivant la variable "tb". Son action est automatiquement répétée suivant la variable "n".

En fait ce schéma n'est pas totalement satisfaisant : on constate des discontinuités en sortie au passage d'un bloc à l'autre. Cela est dû au transitoire du filtre qui se trouve réinitialisé à chaque blocs. Pour l'éviter il faut, dans les opérateurs "z^{-1}" du filtre, utiliser comme conditions initiales d'un bloc la valeur finale du précédent. Cela se réalise par une modification de l'opérateur en créant un bouclage autour de l'opérateur "z^{-1}" généralisé.



On peut préférer utiliser l'estimation faite sur le bloc i-1 pour filtrer le bloc i. Cela se réalise simplement en insérant un retard selon "n" entre le corrélateur et le filtre.

4.Algorithme L.M.S

Un algorithme classique de réalisation de soustracteur de bruit adaptatif est l'algorithme L.M.S (ou algorithme du gradient). Il est formé par un filtre transversal

$$y'(t) = \mathbf{X}(t) \cdot \mathbf{W}$$

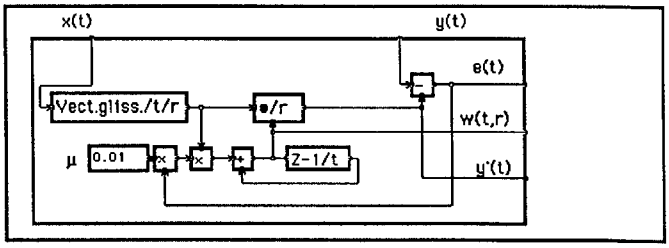
dont le vecteur des coefficients $\mathbf{W}$ est calculé itérativement

$$\mathbf{W}(t) = \mathbf{W}(t-1) + \mu . e(t) . \mathbf{X}(t)$$

en fonction de l'erreur de prédiction

$$e(t) = y'(t) - y(t)$$

Ces équations se transcrivent immédiatement en un graphe à l'aide des opérateurs définis précédemment.



Notons que l'opérateur "z^{-1}" est traversé par un signal fonction de "t" et de "r". La variable "t" est précisée après son nom pour indiquer que c'est selon "t" que le retard doit s'effectuer.

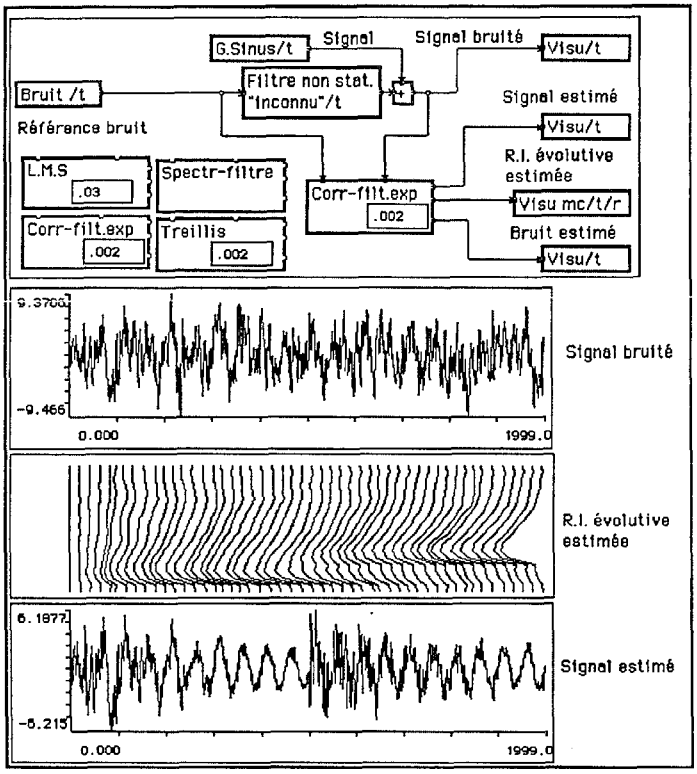
Nous avons utilisé la même disposition des bornes d'entrée sortie que pour le corrélo-filtre. Nous pouvons donc échanger un opérateur par l'autre sans rien changer à l'environnement. (L'éditeur graphique du logiciel le permet en une seule commande.)

5.Comparaisons des Soustracteurs de bruits

Si nous mettons les divers soustracteurs de bruits réalisés sous formes de macros de même apparence externe, nous pouvons utiliser le même diagramme de mise en oeuvre pour chacun d'entre eux. Les résultats peuvent être présentés à l'aide

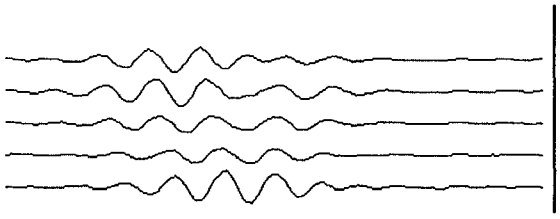
du mode "image" du logiciel. Ce mode permet de rassembler sur une même fenêtre de résultats des parties de diagramme, des courbes et des commentaires. Cette présentation est dynamique : le diagramme et les paramètres peuvent être modifiés directement dans cette fenêtre. En phase d'exploitation, elle suffit au pilotage du traitement.

Nous voyons ci-dessous un exemple dans lequel apparait seulement le diagramme global du traitement et une collection de modules soustracteurs de bruits, avec leurs paramètres, qui peuvent prendre la place du module soustracteur courant. (Cela se fait à l'aide de de la souris, en une seule manoeuvre)

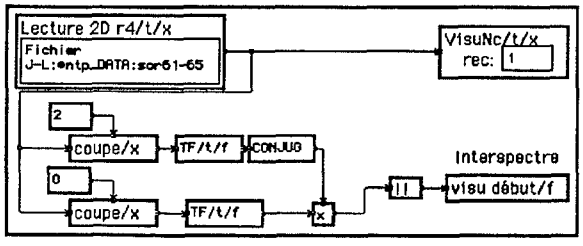


6.Analyse de signaux multi-dimensionnels

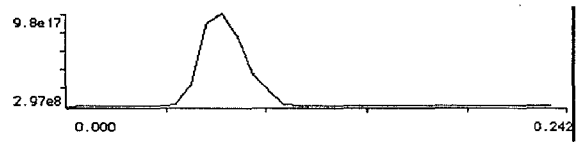
Nous nous proposons ici de traiter un ensemble de signaux sismiques enregistrés sur plusieurs capteurs. Nous avons un ensemble de signaux fonctions du temps "t", soit un signal fonction de "t" et de la position "x". Nous pouvons tout d'abord seulement observer les formes d'ondes.



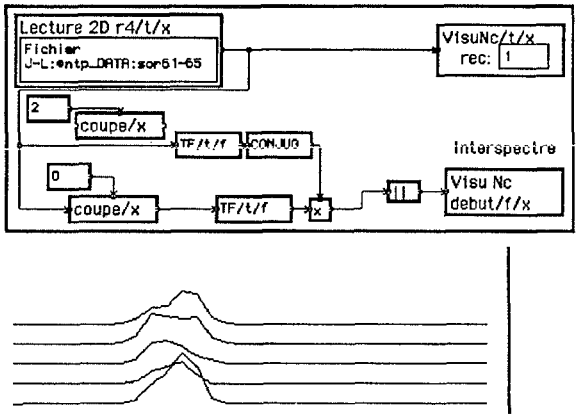
Intéressons nous aux interspectres entre capteurs. Pour calculer l'interspectre entre deux capteurs, nous pouvons utiliser le diagramme suivant. Le module "coupe /x" sélectionne une valeur de "x" : il en ressort un signal fonction seulement de "t", donc la sortie de l'un des capteurs.



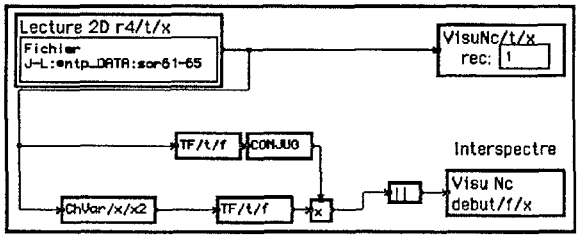
Nous observons l'interpspectre de la voie x=0 avec la voie x=2.

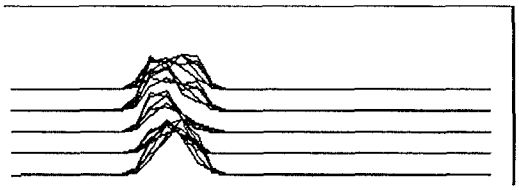


Il est intéressant de comparer les différents interspectres entre l'une des voies et chacune des autres. Il suffit pour cela de supprimer l'un des modules "coupe". L'un des modules "tf" voit alors un signal fonction de "t" et de "x" : il effectue la transformée de Fourier pour chaque valeur de "x". Il en ressort un signal fonction de la fréquence "f" et de la position "x". Celui-ci est multiplié par un signal fonction seulement de "f", ce qui donne encore un signal de "t" et "x", que l'on visualise avec une visualisation en courbes décalées.



Si maintenant nous voulons calculer la matrice interspectrale complète, il faut remplacer le dernière module "coupe" par un changement de variable de"x" en "x2". Ainsi, le multiplieur opère sur un signal fonction de "f" et "x" d'une part, et de "f" et "x2" d'autre part, ce qui donne un signal fonction de "t", "x" et "x2". Nous n'avons pas de visualisation prévue pour trois variables. Si nous gardons la visualisation précédente, nous obtenons des courbes superposées.





Conclusion

Nous avons montré l'aptitude du langage à effectuer quelques traitements classiques. Nous avons montré comment on peut tirer parti de la structure modulaire pour construire progressivement ces traitements et passer d'une variante à l'autre. Nous avons vu la puissance du langage pour les traitement multi-dimensionnels. Ce qui n'apparait malheureusement pas, c'est la simplicité et la rapidité de cette construction, apportée par l'éditeur graphique du logiciel. Indiquons seulement que cet article a été rédigé en utilisant un transfert direct des graphiques (par "copier/coller") entre MUSTIG et un logiciel de traitement de texte.