

DEFINITION D'UN OPERATEUR SYSTOLIQUE POUR LE TRAITEMENT D'IMAGES INFRAROUGES

Alain GIULIERI, Michel ARTAUD, Corinne BECHON

STS - CESDA DCAN de Toulon Les Oursinières 83220 Le pradet

RÉSUMÉ

Nous proposons un opérateur de calcul qui permet de déterminer en temps réel la présence de sources de faible surface apparente dans une image flot de données acquise suivant un balayage vertical. Les traitements nécessaires sont alors d'une part un filtrage bidimensionnel de type dérivateur et d'autre part une estimation du bruit local pour obtenir une normalisation de l'image.

En exploitant les équations récurrentes des calculs à effectuer, on définit alors une structure de calcul à base de réseaux systoliques. Cette structure a l'avantage de pouvoir également s'adapter à une famille de filtres dérivateurs séparables du deuxième ordre. Une simulation en langage SIGNAL (développé par l'IRISA) a permis de valider cette structure et de préciser la taille des bus de données.

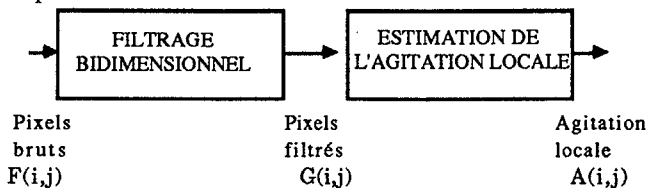
SUMMARY

In this paper we propose a processing operator that permits small target detection in real time in data flow pictures acquired from a vertical scan. This processing used a bidimensional derivative filter and a local noise estimation to obtain a normalised picture

Using recurrent equations in computing, we define a computer shape based on systolic network. This network has the advantage to accept a second order separated filters family. A simulation in SIGNAL langage (developped by IRISA) had can verify this static network and fix data bus sizes.

1 DEFINITION DES CALCULS

Le traitement de l'image infrarouge se fait en deux séquences successives:



En traitement d'image, un filtrage linéaire sur une image brute se fait par la convolution de l'image F avec un masque n de taille M, N et conduit à l'écriture suivante:

$$G(i,j) = \frac{1}{M \cdot N} \sum_{m=-\frac{M-1}{2}}^{\frac{M-1}{2}} \sum_{n=-\frac{N-1}{2}}^{\frac{N-1}{2}} (v(m,n) F(i+m,j+n))$$

avec: $F(i+m,j+n)$: image brute
 $v(m,n)$: noyau de convolution bidimensionnel
 $G(i+k,j+l)$: image filtrée
 M, N : tailles des masques

Lorsque le noyau de convolution est séparable, son écriture devient:

$$v(m,n) = \alpha(m) \cdot \beta(n) \quad \text{avec } m \in \left[-\frac{M-1}{2}, \frac{M-1}{2}\right] \text{ et } n \in \left[-\frac{N-1}{2}, \frac{N-1}{2}\right]$$

D'où l'écriture du filtrage:

$$G(i,j) = \frac{1}{M} \sum_{m=-\frac{M-1}{2}}^{\frac{M-1}{2}} \alpha(m) \left(\frac{1}{N} \sum_{n=-\frac{N-1}{2}}^{\frac{N-1}{2}} \beta(n) F(i+m,j+n) \right)$$

ou encore $G(i,j) = \frac{1}{N} \sum_{n=-\frac{N-1}{2}}^{\frac{N-1}{2}} \beta(n) \cdot \left(\frac{1}{M} \sum_{m=-\frac{M-1}{2}}^{\frac{M-1}{2}} \alpha(m) F(i+m,j+n) \right)$

ce qui montre que les filtrages par rapport aux lignes et aux colonnes sont commutatifs.

Pour la détection d'objets de petite dimension, le noyau de convolution à utiliser doit être un filtre de type passe haut. En effet, dans l'espace fréquentiel, les éléments de petite dimension ont une réponse spectrale large, tandis que le fond a sa réponse concentrée dans les basses fréquences.

Après filtrage, un point sera considéré comme source ponctuelle à extraire si son amplitude est supérieure à l'énergie moyenne de son environnement.

Il s'agit alors, de calculer la variance du signal à traiter, en l'occurrence le signal filtré $G(i,j)$. Cette opération délivre l'image "énergie". Le calcul s'écrit alors:

$$\sigma(i,j) = \frac{1}{K \cdot L} \sum_{k=-\frac{K-1}{2}}^{\frac{K-1}{2}} \sum_{l=-\frac{L-1}{2}}^{\frac{L-1}{2}} P^2(i+k,j+l) - P^{-2}(i,j)$$

avec: $P(i,j)$: moyenne des amplitudes filtrées sur une fenêtre de taille $K \cdot L$

$\sigma(i,j)$: variance de l'image filtrée

K, L : tailles des fenêtres du calcul de la puissance du bruit



Seuls les éléments ayant une amplitude après filtrage nettement supérieure à celle de la puissance moyenne de son voisinage seront conservés, c'est à dire $P(i,j) > \lambda s(i,j)$ où λ représente le seuil de détection

2 ARCHITECTURE DE CALCUL

2.1 Architecture pour le filtrage

Dans l'application considérée, le noyau de convolution utilisé, à un coefficient multiplicateur près, est le filtre dérivateur du second ordre:

1	1	1	1	-8	1	1	1	1
1	1	1	1	-8	1	1	1	1
-4	-4	-4	-4	32	-4	-4	-4	-4
1	1	1	1	-8	1	1	1	1
1	1	1	1	-8	1	1	1	1

qui se décompose en un produit de deux filtres, l'un suivant les lignes et l'autre suivant les colonnes, l'ordre des filtres étant commutatif:

-1	-1	-1	-1	8	-1	-1	-1	-1
----	----	----	----	---	----	----	----	----

-1	-1	-1	-1
----	----	----	----

Pour chaque ligne i, il faut alors calculer l'équation suivante:

$$(1.1) \quad G'(i,j) = F(i,j) - \frac{1}{L-1} \sum_{l=1}^{L-1} (F(i,j-l) + F(i,j+l))$$

et pour chaque colonne j:

$$(1.2) \quad G(i,j) = G'(i,j) - \frac{1}{K-1} \sum_{k=1}^{K-1} (G'(i-k,j) + G'(i+k,j))$$

Il y a deux manières d'effectuer ces calculs:

- soit en faisant chaque fois la somme totale (équation 1.2), méthode qui a été retenue pour le filtrage suivant les colonnes;
- soit en faisant une somme cumulée, mise à jour à chaque pas de calcul, méthode qui a été retenue pour le filtrage suivant les lignes:

$$(1.3) \quad G'(i,j) = F(i,j) - M(i,j-L_1)$$

avec
$$M(i,j-L_1) = \frac{1}{L-1} \left(\sum_{l=L}^0 F(i,j+l) - F(i,j-L_1) \right)$$

ou encore

$$(1.4) \quad M(i,j-L_1) = M(i,j-L_2) + \frac{1}{L-1} [F(i,j) - F(i,j-L) + F(i,j-L_2) - F(i,j-L_1)]$$

avec: $L =$ taille du masque,
 $L_1 = \frac{L-1}{2}$
 $L_2 = \frac{L+1}{2}$

Ces équations mettent en évidence l'ordre des calculs écrits de manière temps réel:

Etape n° 1 calcul de $E1.1 = F(i,j) - F(i,j-L)$ et $E1.2 = F(i,j-L_2) - F(i,j-L_1)$

Etape n° 2 calcul de $E2 = E1.1 + E1.2$

Etape n° 3 calcul de $E3 = M(i,j-L_2) + \frac{1}{L} * E2$

Etape n° 4 calcul de $E4 = F(i,j-L_1) - E3$

La division par L-1 se fait par un décalage de trois éléments binaires au niveau des liaisons des bus dans l'étape n° 3. Le résultat du filtrage de chaque ligne se trouve alors

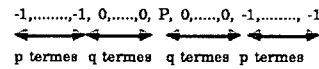
dans le registre R2 après deux tops d'horloge interne et celle que soit la taille du filtre.

Une fois les calculs pour chaque ligne exécutés, il est nécessaire de soustraire les valeurs correspondantes aux lignes supérieures et inférieures (équation 1.2). Dans ce but, l'échange des valeurs est réalisé à l'aide de deux registres qui fonctionnent dans une structure de pile afin de propager les données aux lignes adjacentes (figure 1). Par l'intermédiaire de ces registres, les valeurs passent d'un réseau de calcul au réseau voisin et en même temps sont additionnées à la valeur obtenue pour chaque ligne.

Le résultat final de chaque pixel filtré est encore dans le registre R2, le temps de calcul du filtrage par colonne étant alors de n tops d'horloge interne si la taille du filtre est de 2n+1

Le calcul total du filtrage de chaque pixel se fait donc en n+2 tops d'horloge interne.

Cette structure est très facilement modifiable lorsque le filtre dérivateur séparable du second ordre se décompose selon les vecteurs a et b sous la forme:



avec $P = 2s = 2p$

En effet pour le filtrage suivant les lignes, il suffit de décaler les connexions sur la mémoire et selon les colonnes de ne pas valider les q premières opérations sur le registre R2.

2.2 Architecture pour le calcul de l'agitation locale

En théorie, l'estimation de l'agitation locale se calcule à l'aide de l'écart type (valeur moyenne du filtrage nulle). Cette grandeur est comparée à la valeur du pixel filtré. Afin de diminuer le nombre et la complexité des calculs et par voie de conséquence le temps de réponse, l'estimation de l'agitation locale se fait à l'aide des valeurs absolues. La grandeur obtenue contient le même type d'information dans la mesure où on la compare avec la valeur filtrée.

On calcule la valeur absolue à l'aide d'un additionneur qui reçoit le résultat du filtrage bidimensionnel et d'un test ou un multiplexage.

Le noyau de convolution qui permet d'estimer l'agitation locale est un filtre séparable correspondant à l'équation :

$$(1.3) \quad A(i,j) = \frac{1}{M.N} \sum_{m=-\frac{M-1}{2}}^{\frac{M-1}{2}} \sum_{n=-\frac{N-1}{2}}^{\frac{N-1}{2}} |G(i+m,j+n)|$$

Toujours pour simplifier les calculs, le choix du nombre de lignes et celui du nombre de colonnes se fait de façon telle que leur produit soit proche d'une puissance de 2 afin que la division se traduise par un décalage (ici il a été choisi 7x9 = 63 qui est voisin de 64 = 2⁶).

Ce filtre se représente alors sous forme de deux filtres, dont tous les coefficients sont égaux à 1; appliqués sur les valeurs absolues des pixels précédemment calculés:

Le principe est le même que pour le filtrage mais tous les coefficients sont maintenant identiques. Les termes centraux n'interviennent plus de façon explicite dans le calcul. Cependant le calcul se faisant sur les valeurs absolues, il faut mettre un élément de test sur les $G(i,j)$ et supprimer les registres de synchronisation.

De plus, les calculs entre les lignes et entre les colonnes étant commutatifs, il est préférable de les intervertir afin d'utiliser le même système de passage de données entre les lignes que celui qui a été utilisé dans la partie correspondant au filtrage.

Le montage correspondant au calcul de l'agitation locale est présenté sur la figure 2. Le calcul selon les lignes et selon les colonnes étant commutatif, nous avons choisi, à l'inverse du filtrage, de commencer par le calcul selon les lignes. On constate que ce schéma est plus simple puisqu'il n'y a plus besoin de soustraire le terme central, tous les coefficients étant égaux à 1.

Pour calculer l'agitation selon les colonnes, il faut n tops d'horloge interne si le calcul se fait sur $2n+1$ lignes, et deux tops d'horloge interne pour le calcul selon les lignes quelque soit le nombre de colonnes considérées.

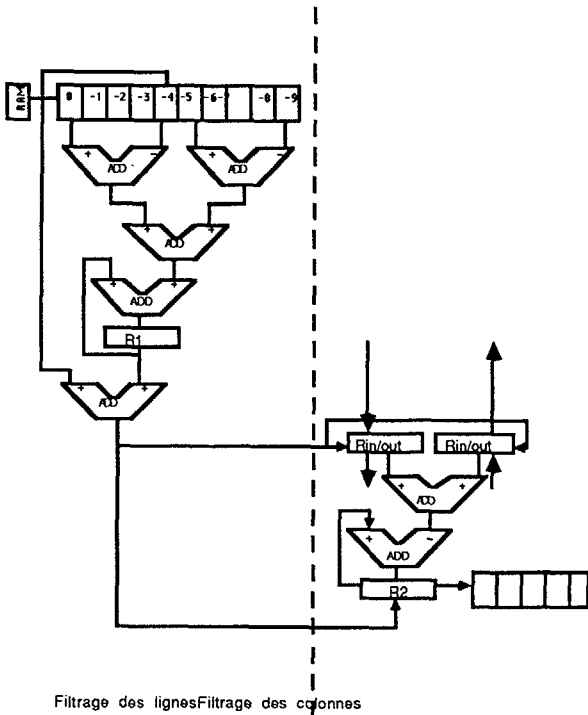


Figure 1 Réseau de calcul pour le filtrage

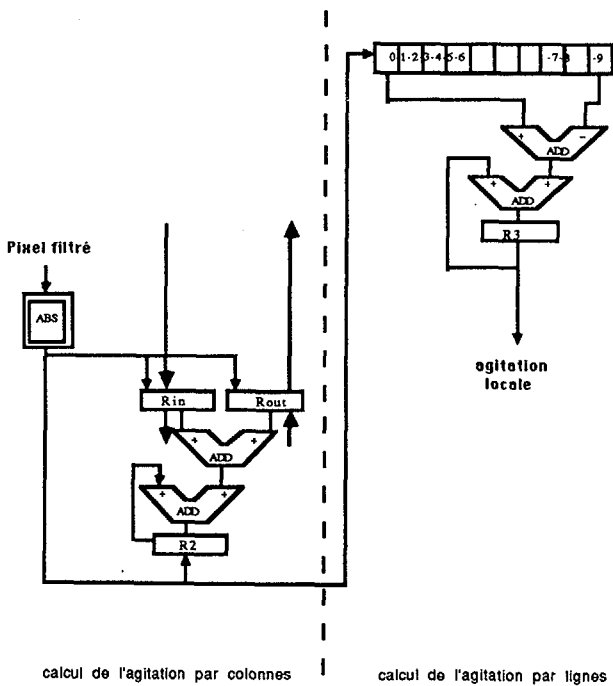


Figure 2 Réseau du filtrage d'un pixel de l'image et du calcul de l'agitation locale.

2.3 Architecture globale

Le schéma global d'une cellule de calcul est constitué des deux éléments qui ont été définis précédemment en utilisant pour le passage des paramètres du filtrage et celui de l'agitation le même système d'entrées-sorties (les registres Rin/out servent pour les deux calculs). Ceci permet de diviser par deux le nombre de ports de dialogue entre les différentes lignes (figure 3).

Il est alors nécessaire de gérer ces registres en fonction des étapes de calcul (figure 4):

- Lorsqu'une nouvelle valeur du pixel filtré a été calculée (elle est contenue dans le registre R2), elle est mise en mémoire et sa valeur absolue est chargée d'une part dans les deux registres Rin/out pour les transmettre aux lignes supérieures et inférieures, et d'autre part dans le registre R2.

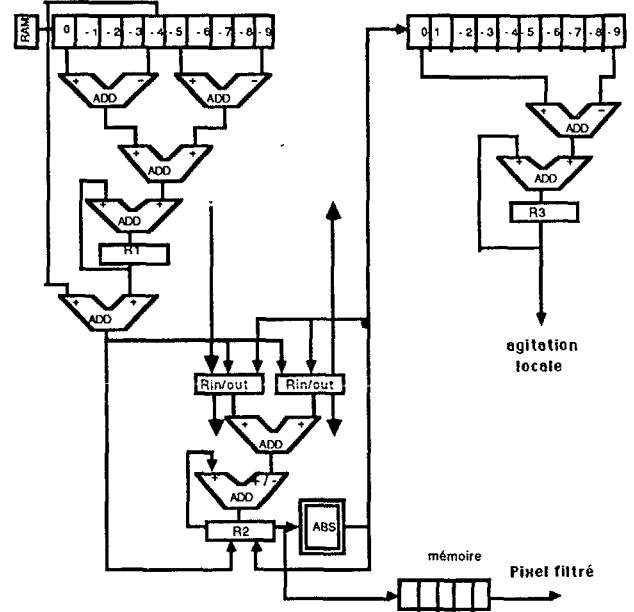


Figure 3 Réseau du filtrage d'un pixel de l'image et du calcul de l'agitation locale.

Le temps de calcul global est de $n_1 + n_2 + 4$ tops d'horloge interne si n_1 est le nombre de lignes du filtrage et n_2 le nombre de lignes du calcul de l'agitation locale. Il n'est pas comptabilisé le top de chargement des registres qui reçoivent la valeur du pixel brut et qui sert de synchronisation à tous les réseaux de calcul.

L'opérateur final est constitué de N cellules montées selon le schéma de la figure 5 (N étant le nombre de lignes de l'image infrarouge).

Lorsque le filtre comporte une zone de zéros centrés, on agit de la même manière que pour le filtrage.

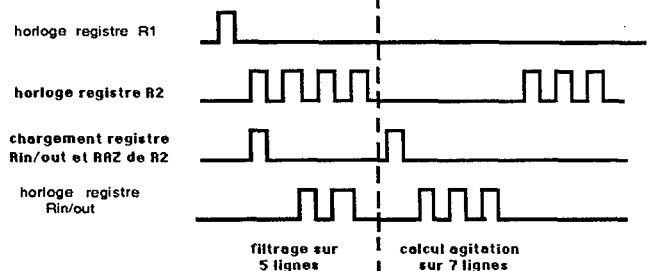


Figure 4 Synchronisation des différents registres d'une cellule de calcul.

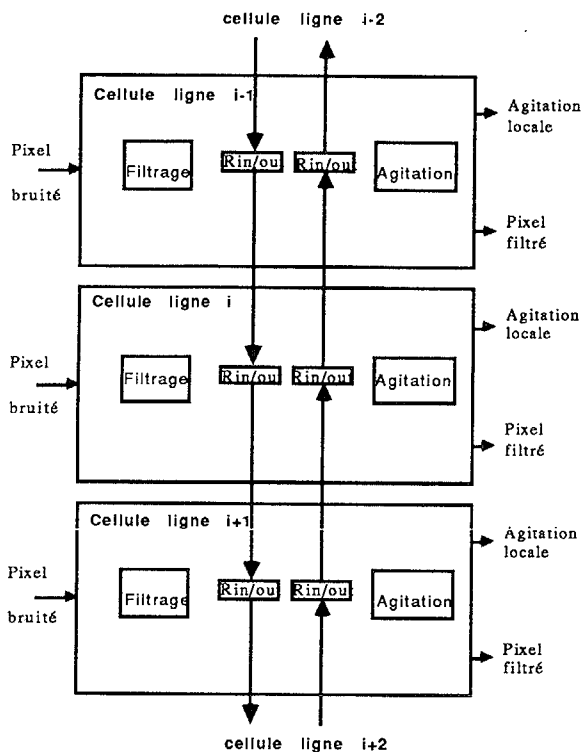


Figure 6 Association des cellules pour former l'opérateur de calcul

3 Programmation en SIGNAL

SIGNAL est un langage parallèle de programmation de haut niveau, développé par l'IRISA. Ce langage, synchrone temps réel orienté flot de données a été conçu pour le traitement du SIGNAL temps réel. Il décrit aussi facilement les algorithmes que les architectures.

3.1 Description

SIGNAL permet de décrire l'architecture établie de manière rigoureuse en respectant le parallélisme, le mécanisme de synchronisation des signaux, le séquençement des traitements, et ce indépendamment d'un composant spécifique.

3.2 Simulation

Le compilateur SIGNAL associe à chaque signal une horloge. Grâce à son calcul d'horloges, il vérifie statiquement avant l'exécution le bon déroulement temporel du programme.

Le code exécutable généré est du FORTRAN.

Le signal d'entrée, associé à un fichier représente l'image acquise suivant le balayage colonne par colonne.

L'architecture intégrale est décrite en 166 lignes seulement. La programmation modulaire a permis la mise au point séparée de chacun des traitements.

Les temps de simulation, obtenus sur une VAX STATION GPX, sont de 8 secondes pour une image 20x20 et de 22 minutes pour une image de 144x512.

La simulation sur un nombre important d'images, dont les pixels étaient codés sur 8 bits, a permis de fixer la taille maximale des bus. Ces dimensions sont de: 8 bits pour le bus des pixels bruts, 9 bits en sortie du filtrage lignes, 9 bits en sortie du filtrage colonnes, 10 bits en sortie du calcul de l'agitation suivant les colonnes, 4 bits en sortie du calcul de l'agitation suivant les lignes

```
SYSTOLYQUZ( integer NLIG, NCOL,      % dimension de l'image brute
%
% M, N, K, L:      % taille des masques %
real SEUIL        % valeur du seuil %
? integer ECHAN   % signal à traiter %
```

```
! [NLIG]integer Y, PF,      % résultat des filtrages
%
% SIG1, SIG:        % résultat des calculs de la
puissance du bruit
integer POSX, POSY, AMPL ) % composantes des détections
%
= ( ! COMP_MOD(NLIG)
| ZCOL:=COL $1
| synchro H, COL
| COL:=((ZCOL+1) when(R=1))default ZCOL
| INDICE:=R when(R>0)
| H:=event ECHAN % horloge de ECHAN %
| H1:=H3 when(((R=NLIG) or (R<((M-1)/2))) and (COL>1))
| H2:=H1 when(R=NLIG) % horloge validant
l'acquisition d'une colonne
%
| FIL_LIG(NLIG, N) % filtrage suivant les lignes %
| FIL_COL(NLIG, M) % filtrage suivant les colonnes %
| VAL_ABS(NLIG)
| H3:=(((R=NLIG) or (R<((K-1)/2))) and (COL>1)) when(((R=NLIG) or (R<((K-1)/2))) and (COL>1))
| AGI_COL(NLIG, K) % calcul de la puissance de bruit en
colonnes
| AGI_LIG(NLIG, L) % calcul de la puissance de bruit en lignes
%
| SEUILLAGE(SEUIL, (NLIG*(L-1))/2, M, N, K, L, NLIG, NCOL)
) !! Y, PF, SIG1, SIG, POSX, POSY, AMPL
where
logical H, H1, H2, H3;
integer R, COL, ZCOL init 0, CELLCOL, INDICE, PFILTRE, SIGMA;
[NLIG]integer P1
process
COMP_MOD { integer N ? logical H ! integer R }
.....end;
FIL_LIG { integer NLIG, N ? integer ECHAN; logical H2 !
[NLIG]integer Y }
.....end;
FIL_COL { integer NLIG, M ? [NLIG]integer Y; integer INDICE;
logical H1
! [NLIG]integer PF, Y1, Y2; integer PFILTRE }
.....end;
VAL_ABS { integer NLIG ? [NLIG]integer PF ! [NLIG]integer P1 }
.....end;
AGI_COL { integer NLIG, K ? [NLIG]integer P1; logical H3 !
[NLIG]integer SIG1 }
.....end;
AGI_LIG { integer NLIG, K ? [NLIG]integer SIG1; integer INDICE;
logical H2
! [NLIG]integer SIG; integer SIGMA }
.....end;
SEUILLAGE { real SEUIL; integer RETARD, M, N, K, L, NLIG, NCOL
? integer PFILTRE, SIGMA, R, COL; logical H
! integer POSX, POSY, AMPL, A1, A2 }
.....end
end
```

CONCLUSION

Cette étude a permis de définir un réseau systolique constitué de cellules qui calculent, à partir des pixels bruts délivrés par un senseur à balayage, la valeur du pixel filtré et l'amplitude de l'agitation locale.

La configuration obtenue est suffisamment générale pour pouvoir être adaptée à toute une famille de filtres dérivateurs séparables du second ordre, en jouant sur les branchements des registres de réception des pixels bruts. De plus en utilisant des valeurs entières écrites dans des mots de 8 éléments binaires, une cellule peut aisément être intégrée dans des composants de logique programmable tels que des Logic Cells Array (LCA) ou des Erasable Programmable Logic Device (EPLD) qui sont des composants disponibles actuellement sur le marché.

BIBLIOGRAPHIE

- [1] W.K.PRATT: "Digital Image Processing". Jhon Wiley Sons 1978.
- [2] A.BENVENISTE, P. LE GUERNIC: Data-flow model of synchronous programming, application to the langage SIGNAL. IRISA publication interne n°385 décembre 87.
- [3] P.LE GUERNIC, A.BENVENISTE, P.BOURNAY, T.GAUTIER: SIGNAL: A data flow oriented langage for signal processing. IEEE Transaction on acoustic speech and signal processing, Vol ASSP34 N°2 April 1986, pages 362-374.
- [4] F.DECHELLE, Y.SOREL: Utilisation du langage SIGNAL pour la spécification et la simulation d'algorithmes de traitement du signal, colloque du GRETSI juin 1987, pages 657-660.
- [5] Y.SOREL, P WOLF: Outils de mise au point de machines pour le traitement du signal, étude d'un cas: réalisation d'un modem 4800bps. Dixième colloque sur le traitement du signal et ses applications. Nice mai 1985, pages 1029-1034.