

ECOLE NATIONALE POLYTECHNIQUE

DEPARTEMENT D'ELECTRONIQUE

المدرسة الوطنية المتعددة التقنيات
BIBLIOTHEQUE — المكتبة
Ecole Nationale Polytechnique

PROJET DE FIN D'ETUDES

SUJET

ETUDE DE LA FAISABILITE
DE L'ANALYSE CEPSTRALE SUR
LE MICROPROCESSEUR TMS 320.10

Proposé par :

B. BOUSSEKSOU

Etudié par :

K. ZEGAGH

M. BENAMIROUCHE

Dirigé par :

B. BOUSSEKSOU

PROMOTION Janvier 1987

PROMOTION :

S O M M A I R E

المدرسة الوطنية المتعددة التقنيات
 BIBLIOTHEQUE — المكتبة
 Ecole Nationale Polytechnique

I/ RECONNAISSANCE DE LA PAROLE	1
I.1. Les systèmes dits globaux	1
I.2. Analyse acoustique	2
I.3. Analyse cepstrale	3
II/ ELEMENTS DE TRAITEMENT DU SIGNAL	4
II.1. Echantillonnage et quantification	4
II.1.1. Echantillonnage	4
II.1.2. Théorème d'échantillonnage	4
II.1.3. Quantification	4
II.2. Convolution et corrélation	5
II.2.1. Convolution	5
II.2.2. Corrélation	6
II.3. Fenêtrage	7
II.3.1. Modèle de Hanning	7
II.3.2. Modèle de Hamming	8
II.3.3. Modèle de Kaiser	9
II.3.4. Conclusion	10
III/ TRANSFORMEE DE FOURIER	12
III.1. Transformée de Fourier discrète (T.F.D)	12
III.1.1. Discrétisation temporelle et fréquentielle d'un signal	12
III.1.2. Relation de définition de la T.F.D directe	17
III.1.3. Relation de la T.F.D inverse	20
III.2. Transformée de Fourier rapide (F.F.T)	21
III.2.1. Algorithme de Cooley	22
III.2.2. Algorithme de Sande	23
III.2.3. Généralisation de l'algorithme de Cooley	24
III.2.4. Généralisation de l'algorithme de Sande	25
III.2.5. Représentation graphique de l'algorithme de Cooley	26
III.2.6. Représentation graphique de l'algorithme de Sande	28
III.2.7. Autres algorithmes de calcul	30
- Entrelacement temporel	30
- Entrelacement fréquentiel	33

IV/	CALCUL D'UNE F.F.T COMPLEXE SUR LE MICROPROCESSEUR TMS320.10	-35
	IV.1. Choix de l'algorithme de calcul	35
	IV.2. Organisation des calculs	-35
	IV.3. Chargement des constantes	-36
	IV.4. Problème de recadrage	-37
	IV.5. Erreurs dues au calcul	38
V/	F.F.T POUR UN SIGNAL REEL EN ENTREE	-42
	V.1. Algorithme de calcul	42
	V.2. Mise en oeuvre sur le TMS320.10	43
VI/	ANALYSE CEPSTRALE SUR LE MICROPROCESSEUR TMS320.10	-53
	VI.1. Adaptation de l'algorithme d'analyse	53
	VI.2. Echelle Mel	53
	VI.3. Problème du lissage	-54
	VI.4. Mise en oeuvre sur le TMS320.10	-54
	VI.4.1. Chargement des constantes	54
	VI.4.2. Filtres Mel	54
	CONCLUSION	55

* DEDICACES *

المدرسة الوطنية المتعددة التقنيات
BIBLIOTHEQUE — المكتبة
Ecole Nationale Polytechnique

Je dédie ce modeste travail à :

- Mes parents
- Mes frères
- Ma soeur

Ma tante SALIHA

- Mes oncles HOCINE, ALLAOUA, et DJOUDI
- Mon ami SMATL
- Tous ceux qui me sont chers

KAMAL

Je dédie ce modeste travail à:

- Mes parents
- Mes frères et soeurs
- La mémoire de mes grands parents
- Mon ami NACER
- Tous ceux qui me sont chers

DJAMEL

* R E M E R C I M E N T S *

Nous tenons à exprimer nos plus vifs remerciements à notre promoteur Mr B.BOUSSEKSOU , pour son aide et ses conseils tout au long de l'élaboration de notre projet de fin d'études .

Nous remercions également Mr N.BENIDDIR pour ses conseils précieux .

Nous n'oublions pas Mr C.BENAMIROUCHE , et Mme S.KHOUFACHE , que nous tenons à remercier , pour nous avoir aidé dans l'exécution de ce document .

Que tous ceux qui ont contribué à notre formation , trouvent ici l'expression de notre profonde gratitude .



I N T R O D U C T I O N



Depuis fort longtemps , le domaine du traitement du signal de parole a suscité d'énormes recherches , notamment dans le domaine de la reconnaissance de la parole .

Le signal de parole , comme tous les autres signaux acoustiques peut être transformé en un signal électrique et se trouve ainsi transmis , traité et conservé en mémoire sous forme analogique ou numérique .

Parmi les méthodes d'analyse du signal de parole , l'analyse cepstrale a paru intéressante en reconnaissance de la parole , d'une part grâce à son économie de représentation (peu de paramètres : 8 coefficients au lieu de 12 pour la prédiction linéaire et 20 pour la méthode spectrale) , et d'autre part grâce à sa facilité de mise en oeuvre sur le micro-processeur .

Dans notre étude , on décrit deux applications potentielles du micro-processeur TMS 320.10 : la transformée de Fourier et l'analyse cepstrale . La première partie situe le rôle de l'analyse cepstrale dans les systèmes de reconnaissance "globale" de la parole . La deuxième partie concerne la mise en oeuvre de la transformée de Fourier sur le TMS 320.10 . Celle-ci est traitée séparément car elle peut être utilisée indépendamment de l'analyse cepstrale . Enfin la dernière partie est consacrée à l'analyse cepstrale , en particulier aux modifications qu'il faut apporter à l'algorithme pour pouvoir le mettre en oeuvre sur le micro-processeur .

Les systèmes de reconnaissance de la parole se répartissent en deux classes au niveau des algorithmes employés pour la reconnaissance :

Les systèmes dits "analytiques" où on cherche à identifier les sens élémentaires qui sont les phonèmes .

Les systèmes dits "globaux" où on considère le mot comme une entité que l'on cherche à reconnaître sans la scinder en ses constituants phonétiques .

I.1 LES SYSTEMES DITS GLOBAUX ..

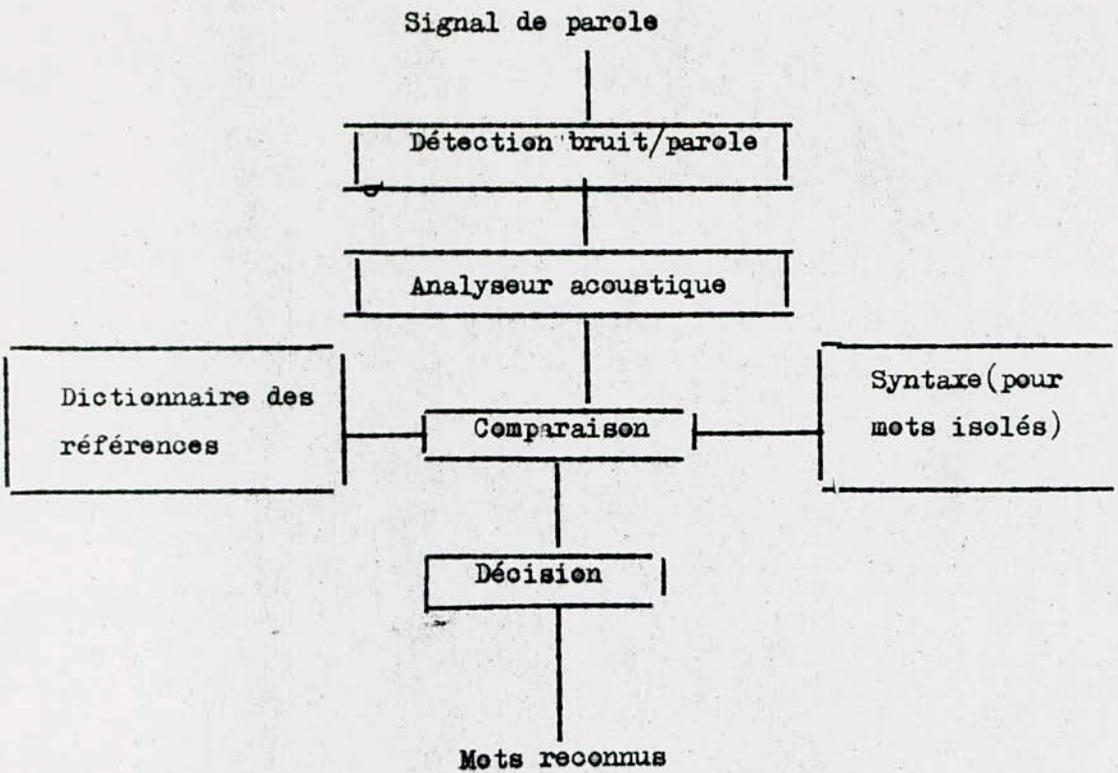
Dans un système global, l'unité minimale est le mot (ou groupe de mots) enregistré comme référence acoustique. Pour reconnaître une phrase avec un système de reconnaissance de mots isolés, il faudra avoir prononcé tous les mots de cette phrase au moins une fois au cours de l'apprentissage. Avec un système de reconnaissance de mots connectés, on pourra utiliser la décomposition des phrases en mots, ainsi on peut reconnaître le nombre 19 par exemple à partir des mots "dix" et "neuf", d'où une réduction de la taille du dictionnaire.

Dans un système de reconnaissance de mots isolés on compare le "mot" à reconnaître à tous les mots du dictionnaire avant de décider lequel a été prononcé.

Dans le cas de mots connectés, on cherche la suite de mots du dictionnaire qui correspond le mieux à la phrase prononcée; l'utilisation d'une syntaxe facilite la recherche en la limitant aux seules phrases grammaticalement correctes.

Puisqu'il faut stocker toutes les références en vue de la reconnaissance, on doit utiliser un codage qui réduise considérablement le débit, tout en gardant les paramètres pertinents pour la reconnaissance.

Le schéma général de la reconnaissance d'un système global de reconnaissance est représenté sur la "figure I.1".



"Fig 1.2" Synoptique d'un système de reconnaissance global

I.2 ANALYSE ACOUSTIQUE

Les méthodes d'analyse du signal de parole les plus employées sont:

_ L'analyse spectrale (par F.F.T) qui permet d'obtenir l'énergie du signal dans des bandes de fréquences.

-L'analyse par prédiction linéaire qui consiste à exprimer la valeur du n_i ème échantillon du signal de parole en fonction des échantillons précédents et de l'entrée du système.

Les principales méthodes utilisées dans cette analyse sont la méthode exacte et la méthode d'autocorrélation.

_L'analyse cepstrale: c'est cette dernière méthode d'analyse qui est actuellement employée dans les systèmes de reconnaissance de mots isolés et de mots connectés.

I.3 ANALYSE CEPSTRALE

Le cepstre complexe est défini comme la transformée de Fourier inverse du Logarithme complexe du spectre. Pour ne pas avoir à manipuler des Logarithmes complexes, on fait l'approximation que le signal de parole est à phase minimale et on utilise le cepstre réel.

Une méthode d'analyse sensiblement différente consiste à remplacer l'échelle linéaire des fréquences par une répartition "Mel" qui est basée sur des critères perceptifs.

Dans les systèmes de reconnaissance de la parole, la méthode d'analyse employée est la suivante:

_ Le signal de parole reçu par le micro est filtré, puis échantillonné à 12,8 KHz et codé sur 16 bits.

_ Le signal est ensuite analysé par blocs de 40 ms.

Pour l'analyse de chaque bloc ; on effectue:

-une préaccentuation numérique de 6dB/octave.

-un fenêtrage pour réduire la distorsion lors du calcul de la transformée de Fourier discrète.

-le calcul du spectre discret par F.F.T .

-le calcul des énergies dans les filtres "Mel" à partir du spectre discret .

-une transformée en cosinus pour obtenir les 8 coefficients cepstraux .

— 8 — o — o —

CHAPITRE II

ELEMENTS DE TRAITEMENT DU SIGNAL

II.1 ECHANTILLONNAGE ET QUANTIFICATION

II.1.1 ECHANTILLONNAGE

L'échantillonnage consiste à représenter un signal fonction du temps $s(t)$ par ses valeurs $s(nT)$ à des instants multiples entiers d'une durée T_e appelée période d'échantillonnage.

Quelle que soit la valeur de T_e , le signal obtenu après échantillonnage sera un signal discret. Ce n'est que dans le cas de la reconstitution du signal analogique qu'il faut imposer une contrainte dans le choix de T_e .

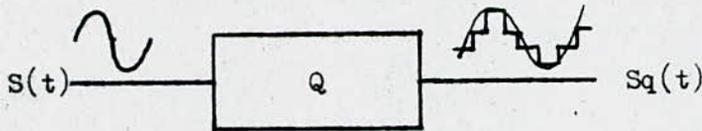
II.1.2 THEOREME D'ECHANTILLONNAGE

Pour que le signal échantillonné soit fidèlement reconstitué, il est nécessaire que la fréquence d'échantillonnage soit plus grande ou égale à deux fois la fréquence maximale contenue dans le spectre du signal (Shanon).

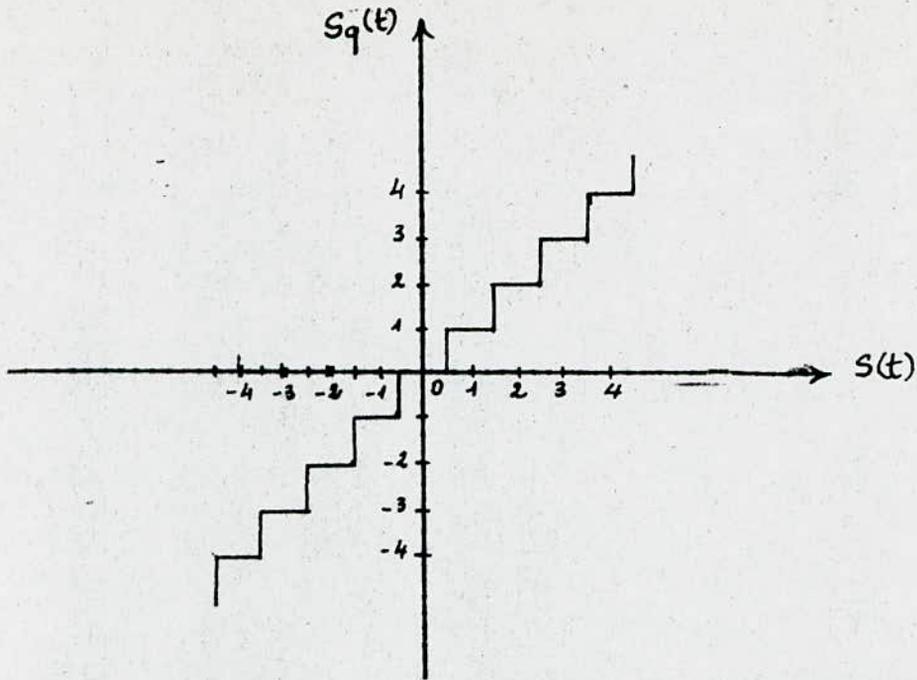
II.1.3 QUANTIFICATION

La quantification est l'approximation de chaque valeur du signal $s(t)$ par un multiple entier d'une quantité élémentaire q appelé échelon de quantification.

Si q est constante quel que soit le signal, la quantification est dite uniforme. Cette opération revient à faire passer le signal dans un organe qui possède une caractéristique en marche d'escalier appelé quantificateur "Q".



La manière dont est faite l'approximation définit le centrage de cette caractéristique (voir figure II.1).



"Fig"II.1 Opération de quantification

La figure ci-dessus représente le cas, appelé arrondi, où toute valeur du signal comprise entre $(n-1/2)q$ et $(n+1/2)q$ est arrondie à nq . C'est l'approximation qui minimise la puissance du signal d'erreur.

Une autre approximation peut être utilisée, qui consiste à approcher par nq toute valeur comprise entre nq et $(n+1)q$. Cette méthode est dite approximation par défaut

II.2. CONVOLUTION ET CORRELATION

Les notions de convolution et de corrélation occupent une place importante dans le traitement du signal, de ce fait il est indispensable de les connaître et de savoir utiliser leurs propriétés.

II.2.1 CONVOLUTION

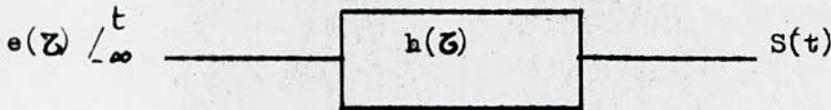
La convolution est définie à partir de la réponse impulsionnelle par une intégrale dans le cas des signaux analogiques. Ainsi, si " $e(t)$ " désigne le signal

à filtrer, "h" la réponse impulsionnelle du filtre, le signal filtré "s(t)" est donné par :

$$S(t) = \int e(\tau) \cdot h(t-\tau) d\tau$$

qui s'écrit symboliquement :

$$S(t) = e(t) * h(t) .$$



L'OPERATION DE CONVOLUTION

On peut passer d'un produit de convolution à un produit simple (Plancherel) par les relations suivantes :

$$e(t) * h(t) = E(f) \cdot H(f) .$$

$$e(t) \cdot h(t) = E(f) * H(f) .$$

C'est à dire qu'on peut faire le passage :

- Filtrage temporel = Convolution fréquentielle .
- Filtrage fréquentiel = Convolution temporelle .

II.2.2 CORRELATION

La définition même de la notion de corrélation montre son grand intérêt en traitement du signal .

La corrélation ou intercorrélation est une opération qui permet d'évaluer la ressemblance d'un signal avec un autre .

Pour cela on décale leurs échelles de temps d'un intervalle "τ" et on multiplie l'un par l'autre .

L'autocorrélation procède des memes opérations, mais exprime la ressemblance d'un signal avec lui même .

Considérons deux signaux x(t) et y(t) , leur fonction d'intercorrélation est définie par :

$$C_{xy}(\tau) = \lim_{T \rightarrow \infty} 1/T \int_0^T x(t) \cdot y(t-\tau) dt$$

En particulier pour un même signal $x(t)$ sa fonction d'autocorrélation est définie comme suit :

$$C_{XX}(\tau) = \lim_{T \rightarrow \infty} \frac{1}{T} \int_0^T x(t) \cdot x(t-\tau) dt$$

Les fonctions de corrélation jouissent de multiples propriétés . Nous citerons parmi elles la plus importante :

-La densité spectrale est la transformée de Fourier de la fonction de corrélation

$$S_{XX}(f) = \text{TF}(C_{XX}(\tau)) ; S_{XY}(f) = \text{TF}(C_{XY}(\tau))$$

II.3. FENETRAGE

Etant donné qu'il est nécessaire de manipuler un nombre fini de valeurs discrètes, il faut donc effectuer une troncature du domaine d'intégration . En général, la limitation de la durée s'obtient en multipliant le signal par une fenêtre rectangulaire d'intensité unité et de durée T_0 . Lors de la troncature, des ondulations parasites du spectre apparaissent, d'où nécessité du choix judicieux des fenêtres de troncature pour que le spectre possède des lobes latéraux aussi peu amples que possible. Les fenêtres rectangulaires sont trop abruptes et ne satisfont pas ces prescriptions. Les modèles les plus connues sont les fenêtres de Hamming, de Hanning et de Kaiser .

II.3.1 MODELE DE HANNING

La fonction fenêtre correspondante est définie dans le domaine temporel par la relation :

$$w_h(k) = 1/2(1 + \cos 2\pi k/N) \quad \text{pour } |k| \leq N/2$$

La transformée de Fourier de cette dernière est de la forme :

$$W_h(f) = 1/2 \frac{\sin \pi f N}{\sin \pi f} + 1/4 \frac{\sin \pi N(f-1/N)}{\sin \pi (f-1/N)} + 1/4 \frac{\sin \pi N(f+1/N)}{\sin \pi (f+1/N)}$$

Les variations des fonctions ci-dessus dans le domaine temporel et fréquentiel sont données par les figures "II.2" et "II.3" .

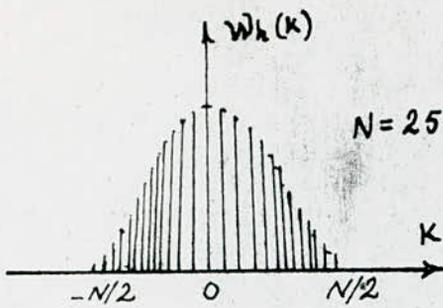
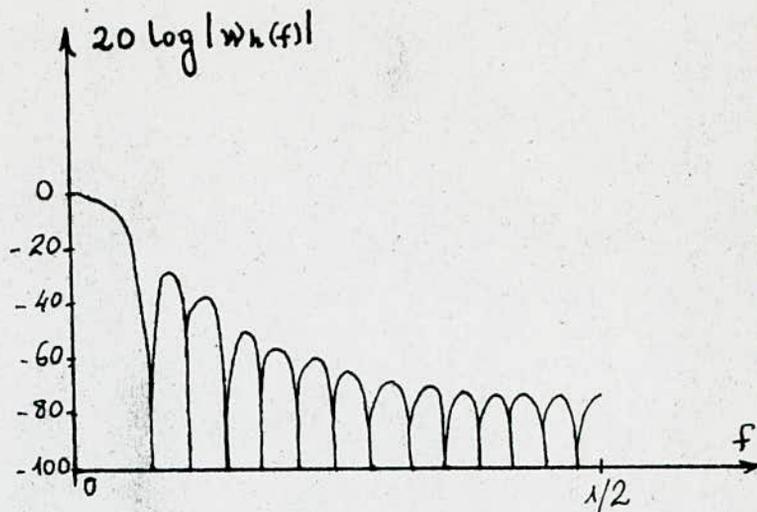


Fig II.2 "Variation de la fonction de Hanning dans le domaine temporel.



"Fig II.3" Variation de la fonction de Hanning dans le domaine fréquentiel

II.3.2 MODELE DE HAMMING

Sa fonction est donnée dans le domaine temporel sous une forme généralisée :

$$W_H(k) = \begin{cases} \alpha(1-\alpha) \cos(2\pi k/N) & \text{pour } |k| \leq N/2 \\ 0 & \text{partout ailleurs} \end{cases}$$

appelée fenêtre de Hamming généralisée .

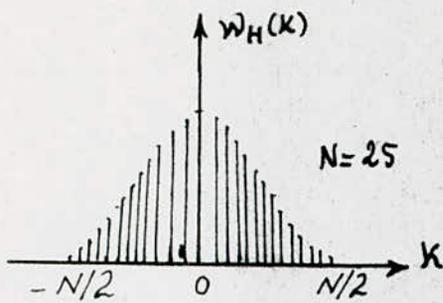
Pour $\alpha = 1/2$, on obtient la fenêtre de Hanning .

Et pour $\alpha = 0,54$, la fenêtre obtenue s'appelle "fenêtre de Hamming".

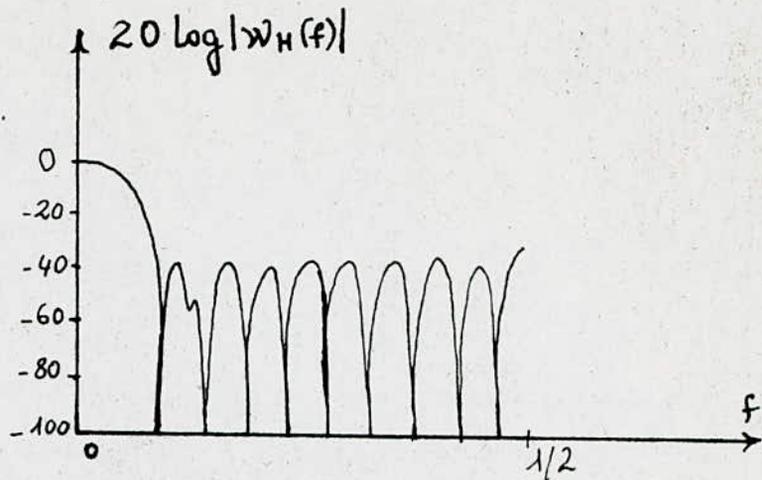
Sa transformée de Fourier est donnée par :

$$W_H(f) = 0.54 \frac{\sin \pi f N}{\sin \pi f} + 0.23 \frac{\sin \pi N(f-1/N)}{\sin \pi (f-1/N)} + 0.23 \frac{\sin \pi N(f+1/N)}{\sin \pi (f+1/N)}$$

Les variations des fonctions dans les domaines temporel et fréquentiel sont données par les figures "II.4" et "II.5" .



"Fig" II.4 Variation de la fonction de Hamming dans le domaine temporel



"Fig II.5" Variation de la fonction de Hamming dans le domaine fréquentiel.

II.3.3 FENETRE DE KAISER

Une autre famille de fonctions fenêtres a été proposée par Kaiser. Elle permet selon la valeur du paramètre β de spécifier dans le domaine des fréquences le compromis entre la largeur du pic central et la l'amplitude des lobes secondaires.

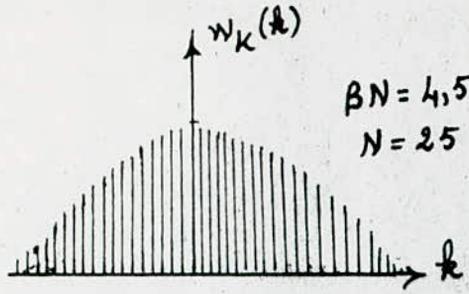
La forme générale de cette fenêtre est la suivante:

$$w_K = \begin{cases} \frac{I_0(\sqrt{N^2 - 4k^2})}{I_0(N)} & \text{pour } |k| \leq N/2 \\ 0 & \text{Ailleurs} \end{cases}$$

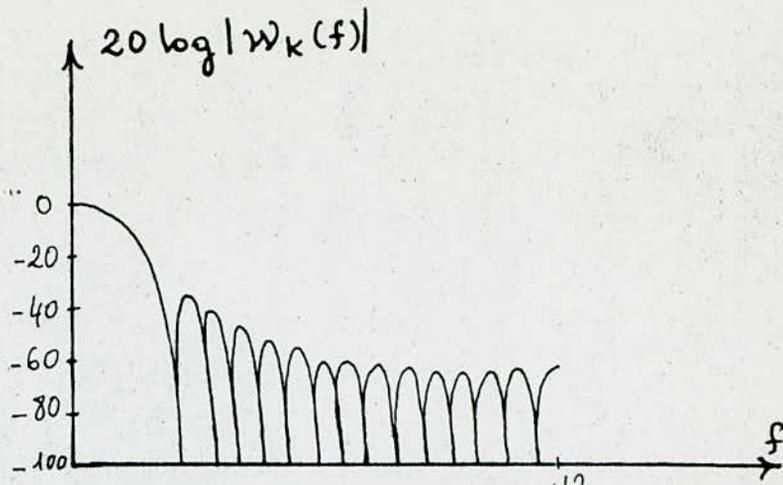
Où $I_0(\cdot)$ est la fonction de BESSEL du 1^{er} espèce d'ordre 0 .

β est un paramètre ajustable.

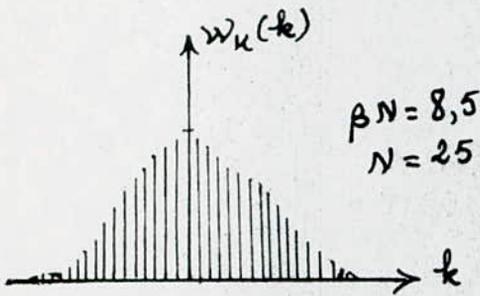
Les figures "II.6", "II.7", et "II.8", "II.9" ci-dessous représentent les variations des fonctions de KAISER dans les domaines temporel et fréquentiel pour $\beta N=8,5$ et $\beta N=4,5$.(N=25).



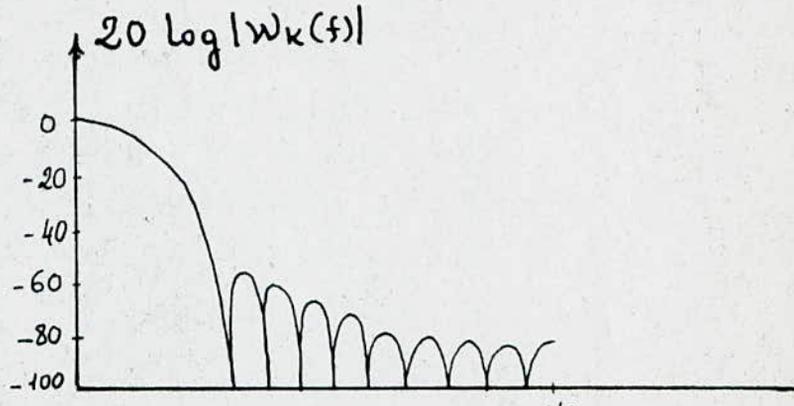
"Fig II.6" Variations de la fonction de Kaiser dans le domaine temporel



"Fig II.7" Variations de la fonction de Kaiser dans le domaine fréquentiel



"Fig II.8" Variations de la fonction de Kaiser dans le domaine temporel.



"Fig II.9" Variations de la fonction de Kaiser dans le domaine fréquentiel

II.3.4 CONCLUSION

Les fonctions fenêtres étudiées nous permettent d'atténuer les lobes secondaires, seulement pas de la même manière.

En mesurant le rapport λ du pic central au premier lobe secondaire, il apparaît que pour ($N=9$), la fenêtre de Hanning conduit à un rapport λ qui vaut 32dB .

Pour la même valeur de N , la fenêtre de Hamming conduit à un rapport $\lambda = -90$ dB .

Ceci montre l'énorme atténuation des lobes secondaires en passant de la fenêtre de Hanning ($\alpha=1/2$) à la fenêtre de Hamming ($\alpha=0,54$).

Cependant, il apparait une caractéristique importante de la famille de fenêtres de Kaiser qui est la possibilité d'obtenir de fortes atténuations des lobes secondaires tout en conservant une largeur minimale pour le pic central et ce en agissant sur le paramètre β .

— o — o — o —

CHAPITRE III

TRANSFORMEE DE FOURIER

C'est une transformation qui permet le passage d'une fonction de La Variabl temporelle t , à une fonction de la variable fréquentielle f par :

$$X(f) = \int_{-\infty}^{+\infty} x(t) e^{-2\pi jft} dt$$

à la condition que cette intégrale existe .

III.1 LA TRANSFORMEE DE FOURIER DISCRETE (T.F.D)

La transformation de Fourier telle que définie précédemment n'est pas sous une forme appropriée pour un traitement numérique pratique . Des modifications nous permettent d'obtenir une version discrète appelée transformée de Fourier discrète (T.F.D) qui se présente sous une forme utilisable. L'avantage de la T.F.D. est l'existence d'un algorithme qui permet d'élargir son domaine d'utilisation .

La transformation de Fourier discrète est la représentation fréquentielle des suites temporelles périodiques, elle possède trois caractères fondamentaux :

- Le signal est une suite périodique finie .
- Son spectre est également une suite périodique finie .
- Les périodes du signal et les périodes du spectre contiennent toutes le même nombre d'échantillons .

L'étude préalable de la T.F.D. est nécessaire pour comprendre la transformation de Fourier rapide (T.F.R); nous la présenterons en trois étapes :

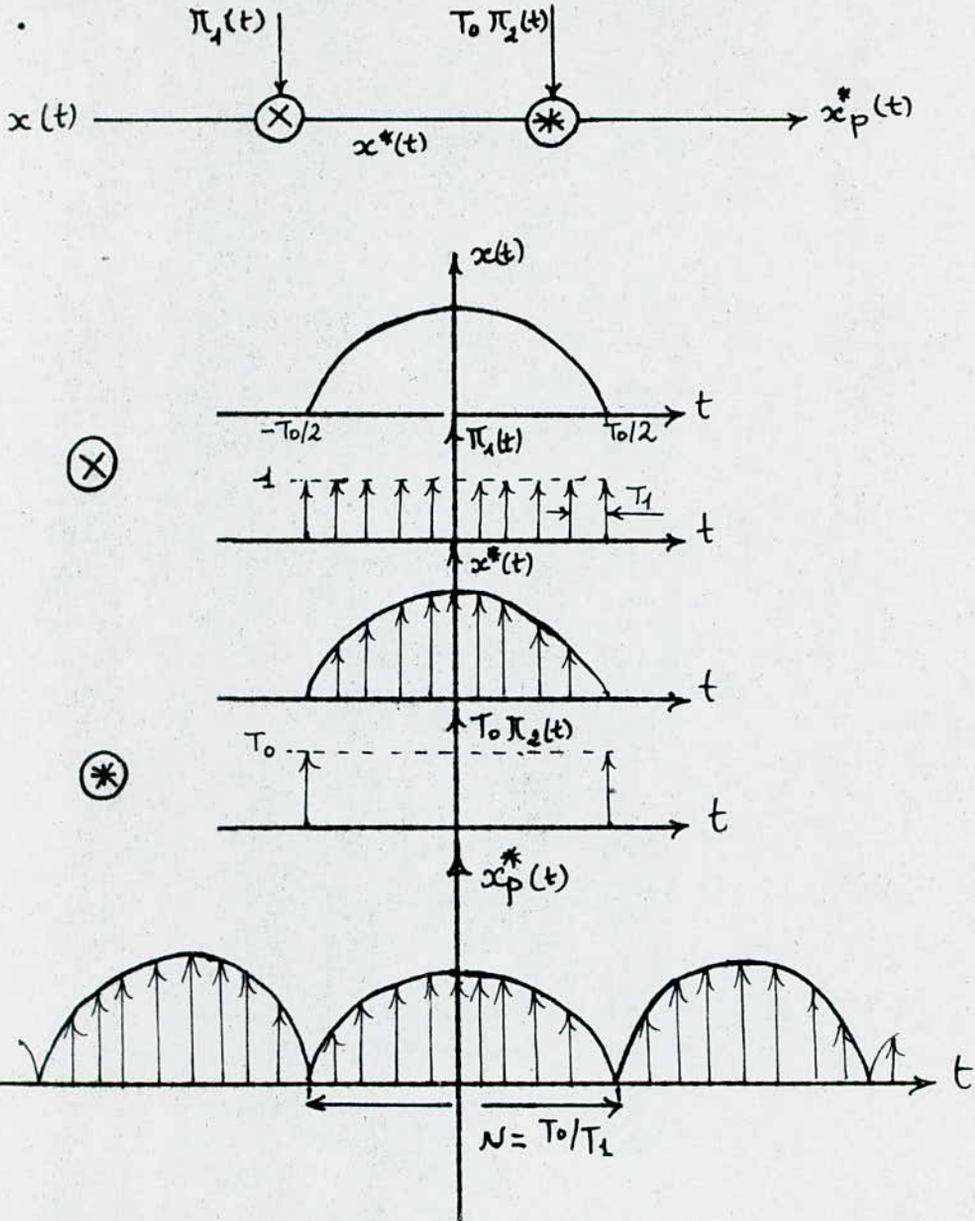
III.1.1 DISCRETISATION TEMPORELLE ET FREQUENTIELLE D'UN SIGNAL

a/ Signal de brève durée : dans ce premier cas, il suffit de subir au signal deux opérations : un échantillonnage impulsionnel et une périodisation .

L'échantillonnage du signal s'obtient en le multipliant par un peigne de Dirac unitaire et centré de période T_1 , d'où une suite de même durée que le signal initial $x^*(t)$.

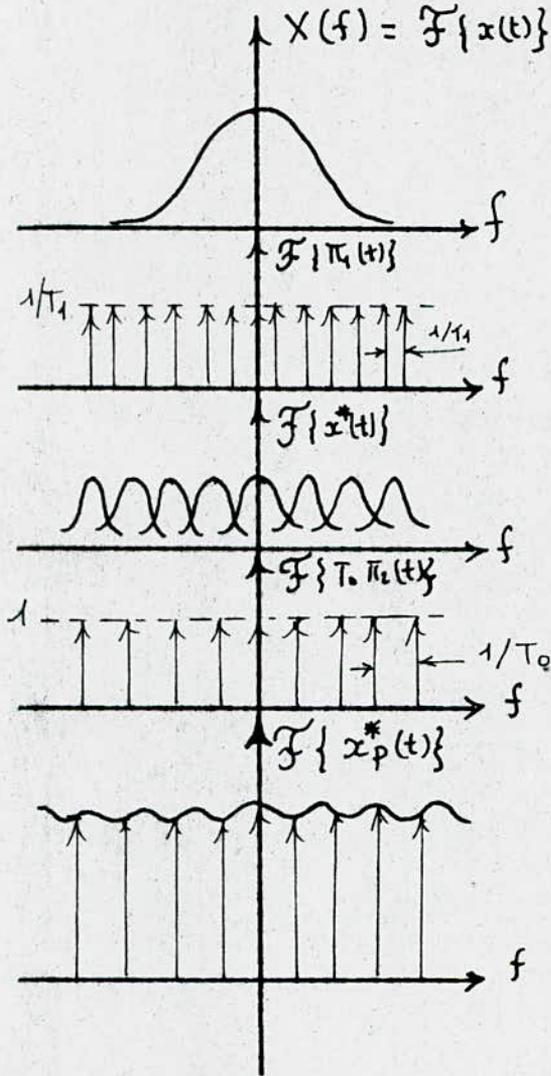
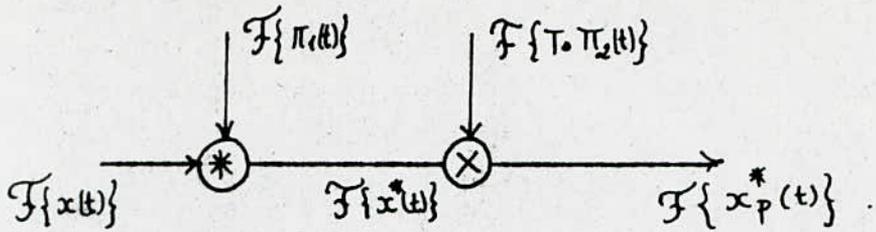
La périodisation de cette suite s'obtient en la convoluant avec un second peigne de Dirac centré et d'intensité T_0 , d'où une suite périodique $x_p^*(t)$.

Les deux opérations sont illustrées par les figures suivantes (dans le domaine temporel).



"Fig III.1" Opération d'échantillonnage impulsionnel et de périodisation dans le domaine temporel.

L'illustration des deux opérations précédentes dans le domaine fréquentiel est donnée par les figures suivantes :



"Fig III.2" Opérations d'échantillonnage impulsif et de périodisation dans le domaine fréquentiel.

CONSEQUENCE: Le nombre d'échantillons contenu dans chacune des périodes du spectre définitif est égal au nombre d'échantillons contenu dans chacune des périodes de la suite $x_p^*(t)$.

b/ Signal de longue durée : dans ce deuxième cas, trois opérations sont nécessaires pour convertir un signal et son spectre en deux suites périodiques

- Un échantillonnage .
- Une limitation de durée .
- Une périodisation .

L'échantillonnage s'obtient en multipliant le signal par un peigne de Dirac unitaire, centré et de période T_1 .

$$x^*(t) = \sum_{-\infty}^{+\infty} x(kT_1) \cdot \delta(t - kT_1) \quad \text{---(1)}$$

La limitation de la durée s'obtient en multipliant le signal échantillonné par une fenêtre $\phi(t)$ d'intensité unité et de durée T_0 . Cette seconde opération nous fournit une suite tronquée contenant N échantillons .

$$x_T^*(t) = \sum_{k=0}^{N-1} x(kT_1) \cdot \delta(t - kT_1) ; \quad N = \frac{T_0}{T_1} \quad \text{---(2)}$$

La périodisation de cette dernière suite s'obtient en la convoluant avec un peigne de Dirac d'intensité T_0 et de période T_0 , d'où la suite définitive

$$x_{T,p}^*(t) = \left[\sum_k x(kT_1) \cdot \delta(t - kT_1) \right] * \left[\sum_{l=-\infty}^{+\infty} T_0 \delta(t - lT_0) \right] \quad \text{---(3)}$$

CONSEQUENCE : Le nombre d'échantillons contenus dans chacune des périodes de la suite (3) est égal au nombre d'échantillons contenus dans chacune des périodes de son spectre .

D'autre part, la position de la fenêtre de troncature doit être judicieusement choisie, de préférence comme indiqué par la figure "III.4" :

L'illustration des trois opérations précédentes est donnée par la figure "III.3"

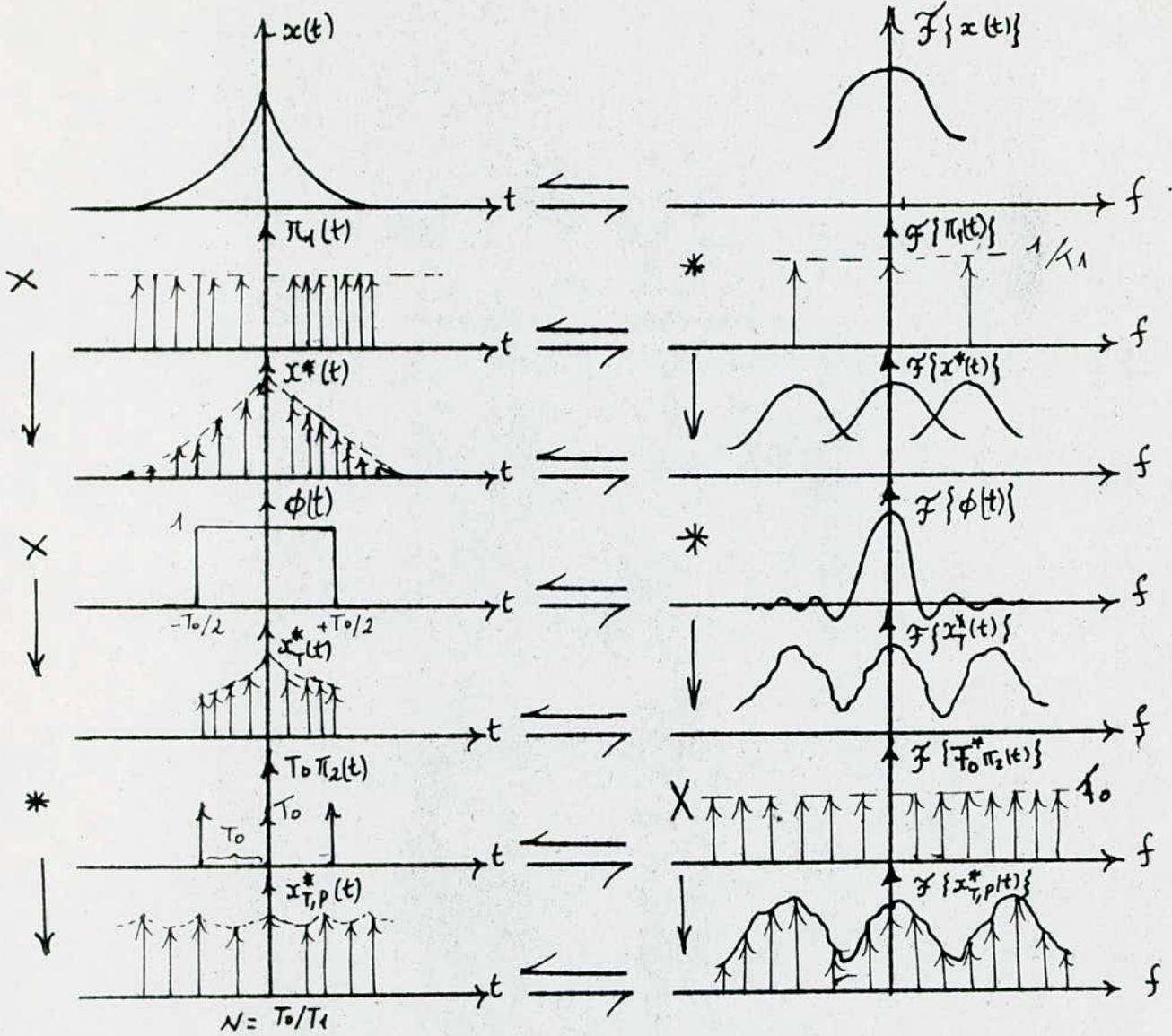
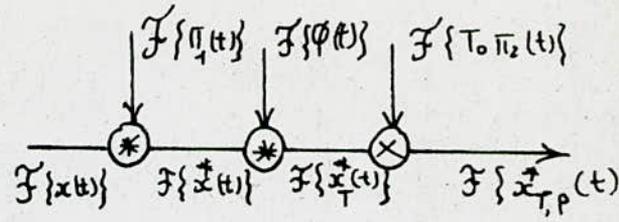
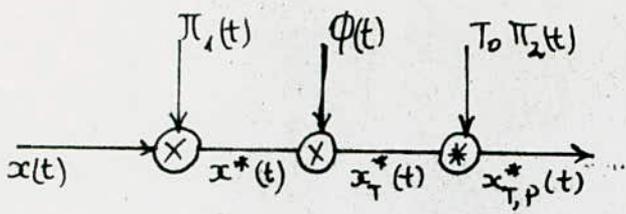


FIGURE "III.3"

ILLUSTRATION DES OPERATIONS : ECHANTILLONNAGE -
LIMITATION DE DUREE - PERIODISATION

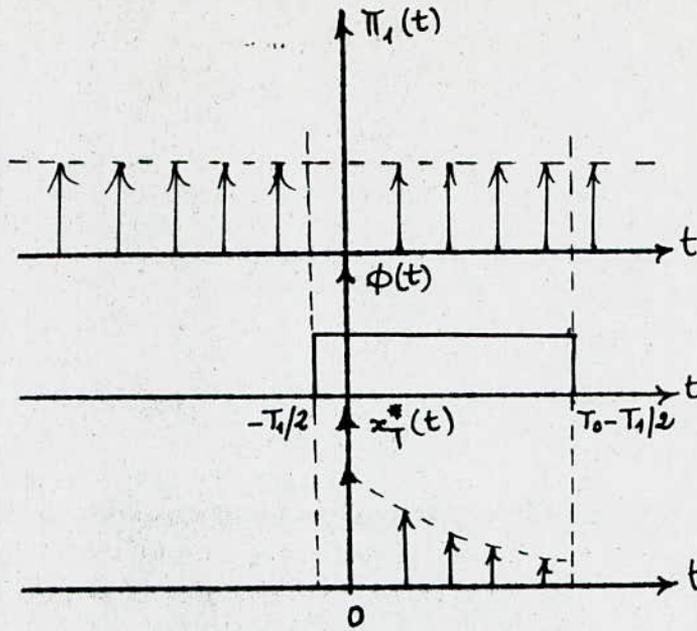


FIGURE "III.4" : ILLUSTRATION DU CHOIX JUDICIEUX DE LA POSITION DE LA FENETRE DE TRONCATURE .

III.1.2 RELATION DE DEFINITION DE LA T.F.D DIRECTE

La transformée de Fourier de la suite aperiodique (1) est donnée par :

$$\begin{aligned}
 \mathcal{F}(x^*(t)) &= \mathcal{F}\left(\sum_{K=-\infty}^{+\infty} x(kT_1) \cdot \delta(t - kT_1)\right) \\
 &= \sum_{K=-\infty}^{+\infty} x(kT_1) \cdot \mathcal{F}(\delta(t - kT_1)) \\
 \mathcal{F}(x^*(t)) &= \sum_{K=-\infty}^{+\infty} x(kT_1) e^{-j2\pi kT_1 f}
 \end{aligned}$$

En limitant le signal à N échantillons, on aura :

$$\mathcal{F}(x_{T,P}^*(t)) = \sum_{K=0}^{N-1} x(kT_1) e^{-j2\pi kT_1 f}$$

Or, le spectre obtenu est discret, ses échantillons étant séparés par l'intervalle

$f_0 = \frac{1}{T}$, la fréquence ne prend donc que les valeurs $f = nf_0$ avec $n=1,2,\dots,N-1$. Ceci implique que $2\pi f_k T_1 = 2\pi \left(\frac{n}{T}\right) k T_1 = 2\pi nk \frac{T_1}{T} = 2\pi nk \frac{1}{N}$.

Chacun des échantillons des périodes spectrales est donc défini par:

$$X(nT_1) = \sum_{k=0}^{N-1} x(kT_1) e^{-j2\pi nk/N} \quad ; n, k=0, 1, 2, \dots, N-1 .$$

Cas usuel: On pose généralement $T_1=1$, d'où la forme usuelle :

$$X(n) = \sum_{k=0}^{N-1} x(k) e^{-j2\pi nk/N} \quad ; n = 0, 1, 2, \dots, N-1 .$$

INTERPRETATION : En posant $W = e^{-j2\pi/N}$, la relation de définition de la T.F.D se détaille comme suit :

$$X(0) = x(0) \cdot W^0 + x(1) \cdot W^0 + \dots + x(N-1) \cdot W^0$$

$$X(1) = x(0) \cdot W^0 + x(1) \cdot W^1 + \dots + x(N-1) \cdot W^{(N-1)}$$

·
·
·
·
·

$$X(N-1) = x(0) \cdot W^0 + x(1) \cdot W^{(N-1)} + \dots + x(N-1) \cdot W^{(N-1)^2}$$

Soit sous forme matricielle :

$$\begin{bmatrix} X(0) \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ X(N-1) \end{bmatrix} = \begin{bmatrix} W^0 & W^0 & \dots & \dots & \dots & W^0 \\ W^0 & W^1 & \dots & \dots & \dots & W^{(N-1)} \\ \cdot & \cdot & \dots & \dots & \dots & \cdot \\ \cdot & \cdot & \dots & \dots & \dots & \cdot \\ \cdot & \cdot & \dots & \dots & \dots & \cdot \\ W^0 & W^{(N-1)} & \dots & \dots & \dots & W^{(N-1)^2} \end{bmatrix} \begin{bmatrix} x(0) \\ x(1) \\ \cdot \\ \cdot \\ \cdot \\ x(N-1) \end{bmatrix}$$

Ou encore $\vec{X}(n) = (W) \cdot x(k)$

PROPRIETES DE LA MATRICE (W)

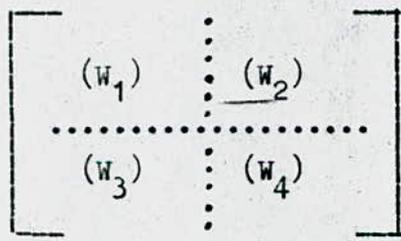
La matrice (W) est carrée symétrique du type N*N . Ses éléments sont de la forme $W^i = \exp(-j2\pi(i/N))$ qui satisfait la relation de conjugaison : $W^{(N-i)} = W^{i*}$ car : $W^{(N-i)} = \exp(-j2\pi(N-i)/N) = \exp(j2\pi(i/N)) = W^{i*}$.
 Par conséquent les lignes et les colonnes de la matrice (W) ayant pour numéros i et N-i sont conjuguées .

COSEQUENCES

La matrice (W) possède donc la structure caractéristique suivante : (N pair, ligne 0, colonne 0, exclues) .

$$(W) = \begin{bmatrix} W^1 & W^2 & \dots & W^{N/2} & \dots & W^{2*} & W^{1*} \\ W^2 & W^4 & \dots & W^N & \dots & W^{4*} & W^{2*} \\ \vdots & \vdots & & \vdots & & \vdots & \vdots \\ W^{N/2} & \dots & W^{N/4} & \dots & W^N & W^{N/2} \\ \vdots & \vdots & & \vdots & & \vdots & \vdots \\ W^{2*} & \dots & W^N & \dots & W^4 & W^2 \\ W^{1*} & W^{2*} & \dots & W^{N/2} & \dots & W^2 & W^1 \end{bmatrix}$$

Dans cette structure nous reconnaissons 4 matrices symétriques : (W₁), (W₂), (W₃) et (W₄), symétriquement disposées par rapport à la ligne N/2 et à la colonne N/2 .



En outre :

- La matrice (W₂) résulte de la conjugaison des éléments de (W₁), suivie d'une

d'une rotation de cette matrice autour de sa dernière colonne .

- La matrice (W_3) résulte de la conjugaison des éléments de (W_1), suivie d'une rotation de cette dernière autour de sa dernière ligne .

- La matrice (W_4) résulte de deux (02) rotations successives de (W_1), l'une autour de sa dernière colonne, l'autre autour de sa dernière ligne .

III.1.3 RELATION DE DEFINITION DE LA T.F.D INVERSE

Par analogie avec la transformation de Fourier continue, posons :

$$x(k) = C \sum_{n=0}^{N-1} X(n) \exp(+j2\pi nk/N) , \text{ et calculons } C .$$

On a :

$$X(n) = \sum_{k=0}^{N-1} x(k) \exp(-j2\pi nk/N) ; \text{ en remplaçant } x(k) \text{ dans cette formule, on obtient:}$$

$$X(n) = \sum_{k=0}^{N-1} \left[C \sum_{n'} X(n') \exp(+j2\pi n'k/N) \right] \exp(-j2\pi nk/N)$$

$$X(n) = C \left[\sum_{n'} X(n') \right] \left[\sum_k \exp(j2\pi(n'-n)k/N) \right]$$

(On a remplacé n par n' pour éviter toute confusion) .

On remarque que : $\sum_k \exp(+j2\pi(n'-n)k/N) = N$ si $n = n'$, par conséquent :

$$X(n) = C \sum_{n'=0}^{N-1} X(n') . N = C . N . X(n) , \text{ ce qui conduit à } C = 1/N .$$

D'où la relation de définition de la T.F.D inverse:

$$x(k) = 1/N \sum_{n=0}^{N-1} X(n) . e^{+j2\pi nk/N} ; k = 0, 1, 2, \dots, N-1 .$$

REMARQUE : En prenant l'expression conjuguée de la T.F.D inverse, on obtient :

$$x^*(k) = 1/N \sum_{n=0}^{N-1} X^*(n) e^{-j2\pi nk/N}$$

$$N . x^*(k) = \sum_{n=0}^{N-1} X^*(n) e^{-j2\pi nk/N}$$

Nous pouvons considérer cette relation comme la transformée directe de la suite $X(n)$; elle nous permet donc de revenir au signal par des calculs analogues à ceux effectués pour obtenir un spectre, soit:

$$x(k) = 1/N \sum_n \left[X^*(n) . W^{nk} \right]^*$$

III.2. TRANSFORMÉE DE FOURIER RAPIDE (F.F.T)

L'emploi de la T.F.D étudiée précédemment comporte une sévère limitation. En effet, le calcul des N échantillons d'une période spectrale exige N^2 opérations. Une organisation méthodique du calcul de la T.F.D ramène le nombre des opérations effectuées à $(N \log_2 N)$.

L'ensemble des méthodes permettant d'accélérer le calcul de la T.F.D constitue la F.F.T.

Dans ce présent chapitre, nous vous fournissons les principes de la F.F.T; comme le choix du nombre N joue un rôle prépondérant, nous commencerons par ses valeurs les plus simples: 4, 8, 16; nous les prendrons sous la forme plus générale 2^r ; r étant entier. Quant aux algorithmes constituant la F.F.T; ils sont à priori en nombre illimité, nous vous présentons au début les deux algorithmes usuels qui sont l'algorithme de COOLEY et l'algorithme de SANDERSON.

Avant d'étudier ces deux types d'algorithmes, nous jugeons nécessaire de disposer d'une notation méthodique pour le signal et sa T.F.D.

NOTATION DU SIGNAL

Puisque chaque période du signal contient $N=2^r$ échantillons, k variant par valeurs entières 0, 1, ..., $N-1$, nous pouvons donc le présenter sous forme binaire $k = k_{r-1}, k_{r-2}, \dots, k_0$.

Nous disposerons donc de deux modes de représentation:

- représentation binaire $k = (k_{r-1}, k_{r-2}, \dots, k_0)$
- représentation décimale $k = (2^{r-1}k_{r-1} + 2^{r-2}k_{r-2} + \dots + k_0)$

Avec la convention adoptée pour k , nous représenterons les échantillons du signal contenu dans une période par $x(k) = (k_{r-1}, k_{r-2}, \dots, k_0)$.

NOTATION DE LA T.F.D

La notation de la variable discrète n est analogue à celle de k . De même nous représenterons les échantillons de chaque période spectrale par:

$$X(n) = X(n_{r-1}, n_{r-2}, \dots, n_0)$$

LES ALGORITHMES FONDAMENTAUX

Le produit nk peut être développé soit au dépens de k , soit au dépens de n . Dans le premier cas, on effectue la décimation temporelle de nk . Cette méthode constitue l'algorithme de COOLEY.

Dans le second cas, on effectue la décimation fréquentielle de nk . Cette méthode constitue l'algorithme de SANDERSON.

III.2.1 ALGORITHME DE COOLEY ($N = 2^3$)

Nous développons nk au dépens de k .

-Le premier terme de nk sera : $n \cdot 4k_2 = (4n_2 + 2n_1 + n_0)4k_2 = 16n_2k_2 + 8n_1k_2 + n_04k_2$.
Par conséquent le premier terme de l'exponentielle se réduit à $W_8^{n_0 4k_2}$.

-Le second terme de nk sera : $n \cdot 2k_1 = 8n_2k_1 + (2n_1 + n_0)2k_1$.

Par conséquent le second terme de l'exponentielle se réduit à $W_8^{(2n_1 + n_0) 2k_1}$.

-Le troisième terme de nk est $(4n_2 + 2n_1 + n_0)k_0$.

Le terme de l'exponentielle correspondant sera : $W_8^{(4n_2 + 2n_1 + n_0)k_0}$.

En définitive les T.F.D à 8 points s'écriront :

$$X(n_2, n_1, n_0) = \sum_{k_0} \sum_{k_1} \sum_{k_2} x(k_2, k_1, k_0) W_8^{c_0} W_8^{c_1} W_8^{c_2},$$

$$\text{où } c_0 = n_0 4k_2$$

$$c_1 = (2n_1 + n_0) 2k_1$$

$$c_2 = (4n_2 + 2n_1 + n_0) k_0.$$

Pour simplifier l'expression de $X(n_2, n_1, n_0)$; on pose :

$$X_{1C}(n_0, k_1, k_0) = \sum_{k_2=0}^1 X_0(k_2, k_1, k_0) \cdot W_8^{c_0}$$

$$X_{2C}(n_0, n_1, k_0) = \sum_{k_1=0}^1 X_{1C}(n_0, k_1, k_0) \cdot W_8^{c_1}$$

$$X_{3C}(n_0, n_1, n_2) = \sum_{k_0=0}^1 X_{2C}(n_0, n_1, k_0) \cdot W_8^{c_2}$$

D'ou une expression des T.F.D à 8 points :

$$X(n_2, n_1, n_0) = X_{3C}(n_0, n_1, n_2)$$

INTERPRETATION

Les trois signaux intermédiaires X_{1C}, X_{2C}, X_{3C} peuvent être considérés comme résultant des substitutions sur les indices k_2, k_1, k_0 par les indices n_0, n_1, n_2 . L'ordre des substitutions est le suivant :

$$\begin{array}{l} k_2, k_1, k_0 \longrightarrow n_0, k_1, k_0 \\ n_0, k_1, k_0 \longrightarrow n_0, n_1, k_0 \\ n_0, n_1, k_0 \longrightarrow n_0, n_1, n_2 \end{array}$$

D'autre part la relation $X(n_2, n_1, n_0) = X_{3C}(n_0, n_1, n_2)$ montre que la T.F.D résulte du renversement des indices de X_{3C} .

$$X_{3C}(n_0, n_1, n_2) = X(n_2, n_1, n_0)$$

III.2.2 ALGORITHME DE SANDE ($N=2^3$)

Développons de nouveau le produit nk, mais cette fois ci au dépens de n. - Cette méthode nous permet d'écrire la T.F.D du signal $x(k_2, k_1, k_0)$ sous la forme :

$$X(n_2, n_1, n_0) = \sum_{k_0} \sum_{k_1} \sum_{k_2} X(k_2, k_1, k_0) W_8^{S_0} W_8^{S_1} W_8^{S_2}$$

Avec :

$$\begin{aligned} S_0 &= n_0 (4k_2 + 2k_1 + k_0) \\ S_1 &= 2n_1 (2k_1 + k_0) \\ S_2 &= 4nk_0 \end{aligned}$$

Posons :

$$X_{1S}(n_0, k_1, k_0) = \sum_{k_2=0}^1 X_0(k_2, k_1, k_0) W_8^{S_0}$$

D'où l'expression de la T.F.D d'un signal défini sur 2^r points et traité par la méthode de COOLEY :

$$X(n) = \sum_{k_0} \sum_{k_1} \dots \sum_{k_{r-1}} x_0(k_{r-1}, k_{r-2}, \dots, k_0) W_N^{C_0} W_N^{C_1} \dots W_N^{C_{r-1}} .$$

Pour simplifier cette expression nous définissons r signaux intermédiaires qui nous permettent de passer par récurrence du signal à son spectre .

$$X_{1C}(n_0, k_{r-2}, \dots, k_1, k_0) = \sum_{k_{r-1}=0}^1 x_0(k_{r-1}, \dots, k_0) W_N^{C_0}$$

$$X_{2C}(n_0, n_1, \dots, k_1, k_0) = \sum_{k_{r-1}=0}^1 X_{1C}(n_0, k_{r-2}, \dots, k_0) W_N^{C_1}$$

.

.

.

.

$$X_{rC}(n_0, n_1, \dots, n_{r-2}, n_{r-1}) = \sum_{k_0=0}^1 X_{r-1C}(n_0, n_1, \dots, k_0) W_N^{C_{r-1}}$$

Pour déduire la T.F.D, on inverse les indices du dernier signal intermédiaire;

$$X(n_{r-1}, n_{r-2}, \dots, n_0) = X_{rC}((n_0, n_1, \dots, n_{r-2}, n_{r-1})) .$$

Les (r+1) relations de récurrence constituent l'algorithme de COOLEY de la F.F.T pour le cas général $N = 2^r$.

III.2.4 GENERALISATION DE L'ALGORITHME DE SANDE ($N = 2^r$)

En développant le produit nk au depends de n:

- Le 1^{er} terme de W_N^{nk} se réduit à $W_N^{a_0}$ avec $a_0 = 2^{r-1} \cdot n_{r-1} \cdot k_0$.

- Le 2^{ème} terme de W_N^{nk} se réduit à $W_N^{a_1}$ avec $a_1 = 2^{r-1} \cdot n_{r-2} \cdot (2k_1 + k_0)$.

⋮

- Le dernier terme de W_N^{nk} se réduit à $W_N^{a_{r-1}}$ avec $a_{r-1} = (2^{r-1} + \dots + k_0) \cdot n_0$;

Par conséquent, la T.F.D d'un signal défini sur 2^r points, traitée par la méthode de Sande se formule comme suit :

$$X(n) = \sum_{k_0} \sum_{k_1} \dots \sum_{k_{r-1}} X(k) \cdot W_N^{a_0} \dots W_N^{a_{r-1}} .$$

Après le renversement de l'ordre des exposants de W_N , on obtient:

$$X(n) = \sum_{k_0} \sum_{k_1} \dots \sum_{k_{r-1}} X(k) \cdot W_N^{a_{r-1}} \dots W_N^{a_0} .$$

Définissons (r) signaux intermédiaires pour pouvoir passer par récurrence du signal à son spectre.

$$X_{1S}(n_0, k_{r-2}, \dots, k_0) = \sum_{k_{r-1}} x_0(k_{r-1}, k_{r-2}, \dots, k_0) \cdot W_N^{S_0}$$

$$X_{2S}(n_0, n_1, \dots, k_0) = \sum_{k_{r-2}} X_{1S}(n_0, k_{r-2}, \dots, k_0) \cdot W_N^{S_1}$$

·
·
·
·

$$X_{rS}(n_0, n_1, \dots, k_0) = \sum_{k_0} X_{r-1S}(n_0, n_1, \dots, k_0) \cdot W_N^{S_{r-1}}$$

Avec $S_0 = a_{r-1}$; $S_1 = a_{r-2}$; ; $S_{r-1} = a_0$.

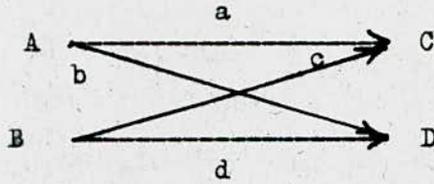
A partir du dernier état intermédiaire X_{rS} , on obtient la T.F.D, et ce par le renversement des indices de X_{rS} .

III.2.5 REPRESENTATION GRAPHIQUE DE L'ALGORITHME DE COOLEY (méthode pratique)

Pour le calcul des T.F.D, on met à profit les propriétés des graphes de fluences. Ainsi les algorithmes de COOLEY et SANDE peuvent être représentés d'une manière plus facile à exploiter que les expressions mathématiques qui les définissent.

Etudions les propriétés des graphes représentant l'algorithme de COOLEY dans l'intention de pouvoir les retrouver ultérieurement sans l'aide des relations de départ.

1°/ Les motifs des graphes de COOLEY qui relient le signal initial à ses états intermédiaires sont des treillis qui sont soit simples, doubles ou multiples. La représentation d'un treillis simple est la suivante:



A et B sont les nœuds sources.

C et D sont les nœuds puits.

a, b, c et d sont les gains des branches.

Les graphes de fluxes de COOLEY satisfont les prescriptions suivantes:

$$a = b = 1$$

$$c = W_N^j$$

$$d = W_N^i$$

2°/ Le nombre de colonnes de treillis contenues dans un graphe de flux de COOLEY en 2^r points est égal à r .

3°/ Le nombre de treillis contenus dans une colonne de rang P d'un graphe de COOLEY en 2^r points est égal à 2^{P-1} ; $P=1,2,\dots,r$.

4°/ La multiplicité des treillis contenus dans une colonne de rang P d'un graphe de COOLEY à 2^r points est égal à 2^{r-P} .

5°/ Les gains des branches (c) et (d), sont de la forme :

$$c = W_N^i, \quad d = W_N^j, \quad \text{avec } j = i + N/2.$$

DEFINITION SUPPLEMENTAIRE

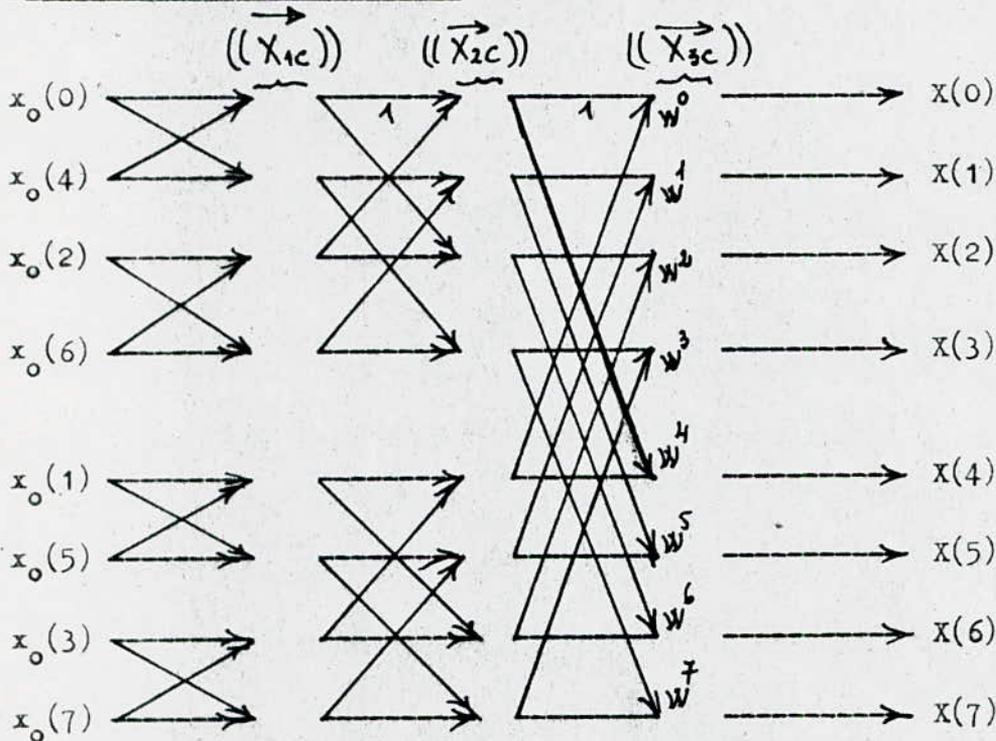
Une colonne de rang P d'un graphe de COOLEY à 2^r points contient 2^{P-1} treillis.

Soient $i_1, \dots, i_{2^{P-1}}$ les valeurs correspondantes de P . Par définition, la liste de ces valeurs constitue l'adresse de i pour la colonne considérée. On appelle adresses pilotes, les adresses des treillis de multiplicité 1.

Les termes des adresses pilotes des colonnes des graphes obéissent aux lois suivantes:

- * Le nombre de termes de l'adresse pilote de la colonne de rang $P \leq r$ est 2^{P-1} .
- * Ces termes sont constitués par les nombres entiers compris entre 0 et $2^{P-1}-1$.
- * La 1^{ère} moitié d'une adresse pilote est constituée de nombres pairs.
- * La 2^{ème} moitié est constituée de nombre impairs.
- * Le contenu d'une adresse pilote s'obtient par renversement binaire.

GRAPHE DE FLUENCE DE COOLEY. (pour $N=2^3$ points)



III.2.6 REPRESENTATION GRAPHIQUE DE L'ALGORITHME DE SANDE (méthode pratique)

La représentation graphique de l'algorithme de SANDE utilise les mêmes méthodes que la représentation de COOLEY.

STRUCTURE GENERALE DES GRAPHES DE SANDE :

Nous désignerons par T_1, T_2, \dots, T_r les treillis (identiques), contenus dans les colonnes de rang $1, 2, \dots, r$.

- La formule générale d'un graphe de SANDE représentant un programme en 2^r points est:

$$G = \sum_{p=1}^r 2^{p-1} \cdot T_p$$

- L'ordre d'un treillis composé est par définition le nombre de treillis simples qui le constituent, par conséquent les treillis notés T_p sont d'ordre 2^{r-p} .

p variant de 1 jusqu'à r .

- Les gains des branches des treillis de SANDE se déduisent des relations suivantes

$a=1$; $b=W^i$; $c=1$; $d=-W^i$ avec :

$i=0, 2^{p-1}, 2 \cdot 2^{p-1}; \dots; 2^{r-1} - 2^{p-1}$

$=0, 2^{p-1}, 2 \cdot 2^{p-1}; \dots; (2^{r-p} - 1) \cdot 2^{p-1}$

- Le désembrouillage du spectre s'effectue en inversant l'adresse binaire du dernier état du signal.

III.2.7 AUTRE ALGORITHMES DE CALCUL

Les algorithmes de COOLY et SANDE sont basés sur un calcul direct qui nécessite $N \cdot N$ multiplications complexes, ce qui exige un temps de traitement considérable.

Dans le but de réduire ce temps de traitement, des simplifications peuvent être envisagées conduisant à des algorithmes plus rapides ne nécessitant que $N/2(\log_2 N)$ multiplications complexes ou moins. Nous citons les algorithmes avec entrelacement temporel et fréquentiel.

1°/ ENTRELACEMENT TEMPOREL

La suite des éléments $x(k)$ peut être décomposée en deux suites entrelacées, celle des éléments d'indice pair et celle des éléments d'indice impair. Le calcul des $N/2$ premiers éléments $X(n)$ apparait comme suit:

$$\begin{bmatrix} x_0 \\ x_1 \\ \cdot \\ \cdot \\ x_{N/2-1} \end{bmatrix} = \begin{bmatrix} 1 & 1 & \cdot & \cdot & 1 \\ & & 2 & & 2^{(N/2-1)} \\ 1 & W & \cdot & \cdot & W \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ 1 & W & \cdot & \cdot & W \end{bmatrix} \begin{bmatrix} x_0 \\ x_2 \\ \cdot \\ \cdot \\ x \end{bmatrix} + \begin{bmatrix} 1 & 1 & \cdot & \cdot & 1 \\ & & 3 & & (N-1) \\ W & W & \cdot & \cdot & W \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ W & W & \cdot & \cdot & W \end{bmatrix} \begin{bmatrix} x_1 \\ x_3 \\ \cdot \\ \cdot \\ x_N \end{bmatrix}$$

$$= \begin{bmatrix} 1 & 1 & \cdot & \cdot & \cdot & 1 \\ & & 2 & & & 2^{(N/2-1)} \\ 1 & W & \cdot & \cdot & \cdot & W \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ 1 & W & \cdot & \cdot & \cdot & W \end{bmatrix} \begin{bmatrix} x_0 \\ x_2 \\ \cdot \\ \cdot \\ x \end{bmatrix} + \begin{bmatrix} 1 & 1 & 1 & \cdot & \cdot & \cdot & \cdot & 1 \\ & & 2 & & & & & 2^{(N/2-1)} \\ W & 1 & W & \cdot & \cdot & \cdot & \cdot & W \\ \cdot & \cdot \\ \cdot & \cdot \\ W & 1 & W & \cdot & \cdot & \cdot & \cdot & W \end{bmatrix} \begin{bmatrix} x_1 \\ x_3 \\ \cdot \\ \cdot \\ x_N \end{bmatrix}$$

Soit sous forme condensée:

$$X(n) = \sum_{k=0}^{N/2-1} x_{2k} \cdot W_{N/2}^{nk} + W_N^n \left\{ \sum_{k=0}^{N/2-1} x_{2k+1} \cdot W_{N/2}^{nk} \right\}$$

Et pour les $N/2$ derniers éléments de l'ensemble des $X(n)$ on a :

$$\begin{bmatrix} X_{N/2} \\ X_{N/2+1} \\ X_{N/2+2} \\ \vdots \\ X_{N-1} \end{bmatrix} \begin{bmatrix} 1 & 1 & \dots & \dots & 1 \\ & W & & & W^{2(N/2-1)} \\ & & W^4 & & W^{4(N/2-1)} \\ & & & \dots & \\ & & & & W^{2(N/2-1)} \\ 1 & W & \dots & \dots & W^{2(N/2-1)} \end{bmatrix} \begin{bmatrix} X_0 \\ X_2 \\ X_4 \\ \vdots \\ X_{2(N/2-1)} \end{bmatrix} = \begin{bmatrix} 1 & 1 & \dots & \dots & 1 \\ & W & & & W^{N-1} \\ & & W^4 & & W^{2(N-1)} \\ & & & \dots & \\ & & & & W^{(N/2-1)(N-1)} \\ W & W & \dots & \dots & W \end{bmatrix} \begin{bmatrix} X_1 \\ X_3 \\ X_5 \\ \vdots \\ X_{N-1} \end{bmatrix}$$

Soit sous forme condensée:

$$X_{N/2+k} = \sum_{k=0}^{N/2-1} x_{2k} \cdot W_{N/2}^{nk} - W_N^n \left\{ \sum_{k=0}^{N/2-1} x_{2k+1} \cdot W_{N/2}^{nk} \right\}$$

En posant $G_n = \sum_{k=0}^{N/2-1} x_{2k} \cdot W_{N/2}^{nk}$ et $H_n = \sum_{k=0}^{N/2-1} x_{2k+1} \cdot W_{N/2}^{nk}$, on aura:

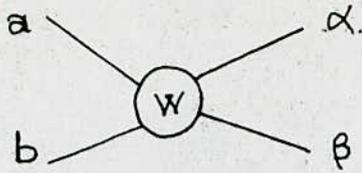
$$X_n = G_n + W_N^n \cdot H_n$$

pour $n = 0, 1, 2, \dots, (N/2-1)$.

$$X_{N/2+n} = G_n - W_N^n \cdot H_n$$

Donc, le calcul d'une F.F.T sur N points se réduit à un calcul de deux F.F.T sur $N/2$ points en plus d'une étape de papillon.

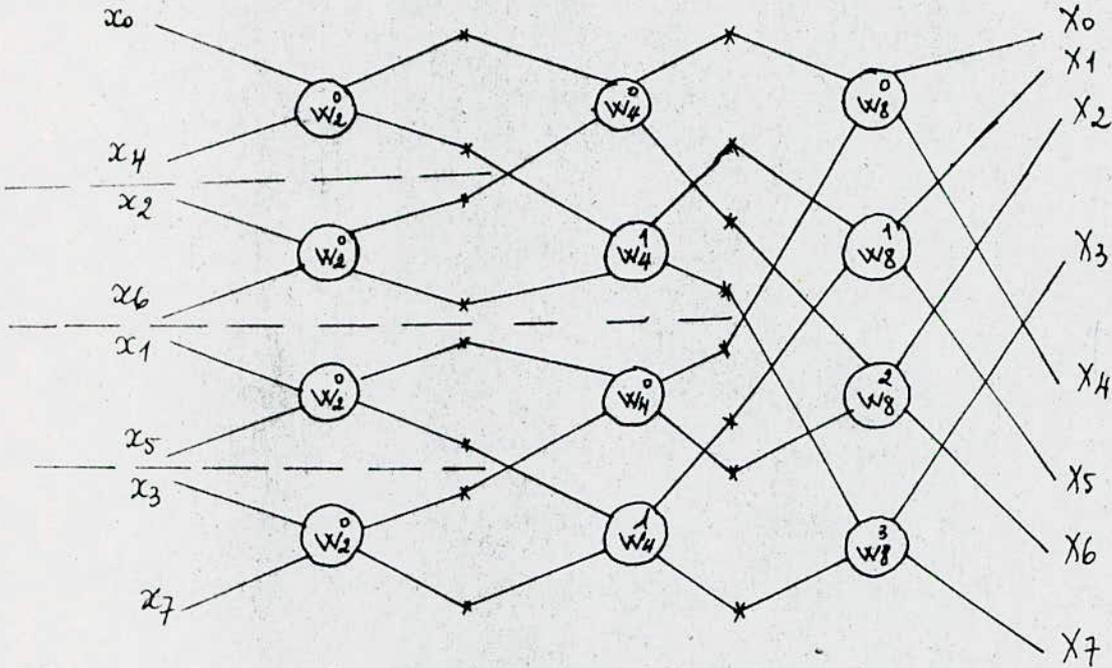
Le graphe de fluence est basé sur le schéma de papillon suivant :



Avec

$$\begin{cases} \alpha = a + Wb \\ \beta = a - Wb \end{cases}$$

Ainsi pour $N = 8$ points on obtient le schéma suivant :

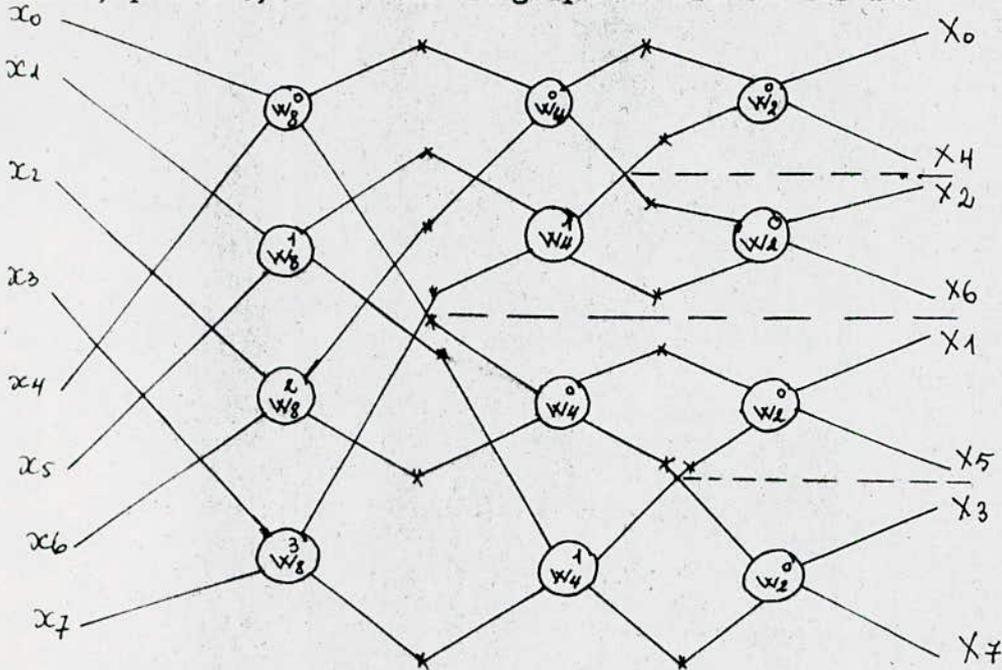


Il est à noter que ces graphes sont donnés dans le cas où le signal a subi un renversement binaire , dans le but d'obtenir un spectre ordonné .

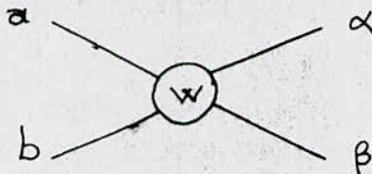
2°/ ENTRELACEMENT FREQUENTIEL

Cependant, il est possible d'utiliser une autre décomposition des calculs où on rentre les données dans l'ordre normal et où on obtient les résultats dans l'ordre du "bit reverse". Dans ce cas on fait décomposer la suite des éléments $X(n)$ en deux suites entrelacées, celle des éléments d'indice pair et celle des éléments d'indice impair.

Ainsi, pour $N=8$, on obtient le graphe de fluence suivant:



Le schéma du papillon est défini comme suit:



$$\begin{cases} \alpha = a + b \\ \beta = W \cdot (a - b) \end{cases}$$

Ces décompositions montrent que le calcul d'une transformée de Fourier d'ordre N revient au calcul de deux transformées d'ordre $N/2$, auquel il faut ajouter $N/2$ multiplications complexes. L'ensemble de la transformation demande un nombre de multiplications complexes M_c qui s'écrit :

$$M_c = N/2 * \text{Log}_2 (N/2)$$

et un nombre d'additions A_c tel que :

$$A_c = N * \text{Log}_2 N$$

REMARQUE

Ces algorithmes sont basés sur une décomposition de la transformée d'ordre N en transformées élémentaires d'ordre 2, c'est à dire le nombre de points est une puissance de 2. Cependant, il est possible d'obtenir encore des algorithmes plus performants en faisant des regroupements par 4 ou 8 au lieu de 2. Les algorithmes sont alors plus compliqués.

CHAPITRE IV

CALCUL D'UNE F.F.T COMPLEXE SUR LE MICROPROCESSEUR

TMS 320.10

IV.1 CHOIX DE L'ALGORITHME DE CALCUL

La place mémoire disponible sur le TMS 320 (1 RAM de 144 mots de 16 bits) ne permet pas de réaliser une F.F.T sur plus de 64 points complexes sans utiliser de mémoires extérieures. Dans la suite on s'intéressera seulement au cas de la F.F.T sur 64 points complexes. Une partie de cette RAM est utilisée pour stocker les données $x_k, k=0,1,\dots,63$; en fin de calcul, on récupérera les résultats dans cette même place mémoire. Une autre partie est utilisée pour stocker les constantes $W_{64}^n, n=0,1,\dots,31$ qui sont nécessaires pour le calcul.

Le calcul du papillon ($\alpha=a+Wb, \beta=a-Wb$) nécessite (45) cycles de calcul. Comme le calcul de la F.F.T sur 64 points complexes nécessite (32x6) papillons, cela fait un temps de calcul égal à (1,5ms) .

Pour éviter les débordements au cours du calcul de la F.F.T, il est nécessaire de procéder à des recadrages . Pour ce faire, on peut utiliser les divisions par 2 ou par 4 ou pas de division (suivant ce qui est programmé) au cours du transfert de l'accumulateur vers la mémoire RAM, à condition que l'on n'écrive pas en mémoire des résultats intermédiaires au cours du calcul d'un papillon.

IV.2 ORGANISATION DES CALCULS

Pour chaque niveau le nombre de paquets de papillons est reporté ci-dessous (avec les constantes utilisées à chaque niveau).

$$\text{niveau } 0 : 32 * \left\{ W_2^0 \right\}$$

$$\text{niveau } 1 : 16 * \left\{ W_4^0, W_4^1 \right\}$$

$$\text{niveau } 2 : 8 * \left\{ W_8^0, W_8^1, W_8^2, W_8^3 \right\}$$

$$\text{niveau } 3 : 4 * \left\{ W_{16}^0, W_{16}^1, \dots, W_{16}^7 \right\}$$

$$\text{niveau } 4 : 2 * \left\{ W_{32}^0, W_{32}^1, \dots, W_{32}^{15} \right\}$$

$$\text{niveau } 5 : 1 * \left\{ W_{64}^0, W_{64}^1, \dots, W_{64}^{31} \right\}$$

A chaque niveau, il est nécessaire d'écrire un papillon, à cause de l'incrément du pointeur mémoire nécessaire pour passer de a à b qui diffère d'un niveau à un autre.

Au niveau j l'incrément est égal à 2^{j+1} , en rangeant les données dans l'ordre du "bit reverse" au départ. (Chaque donnée occupe 2 cases mémoires consécutives; la première pour la partie réelle, la suivante pour la partie imaginaire).

Pour les niveaux 0 et 1, il vaut mieux remplacer les constantes W et W par leur valeur respective 1 et -j, pour gagner en temps de calcul.

Il serait possible de gagner un peu en place mémoire RAM nécessaire pour le stockage des constantes W en prenant compte de la symétrie:

$$W_{64}^{32-n} = \overline{W_{64}^n}$$

Mais dans ce cas, on augmenterait considérablement la longueur du programme.

IV.3 CHARGEMENT DES CONSTANTES

Les constantes W devront être disponibles dans la RAM pour le calcul de la F.F.T, il faut donc aller les lire quelque part au cours d'une phase d'initialisation. Deux solutions sont possibles, soit les lire à partir d'une ROM externe, soit faire des chargements littéraux à partir de la ROM programme. Cette 2^{ème} solution est préférable à condition qu'il n'y ait pas de problème de place mémoire.

Dans le cas de calcul d'une F.F.T sur 64 points, pour stocker les 32 valeurs complexes dans la mémoire programmée, il faudrait utiliser 64 lignes; cependant, on peut tenir compte des différentes symétries existant entre les W pour ne stocker que le minimum de valeurs dans la RAM.

Le temps nécessaire pour l'initialisation sera alors un peu plus long, mais le programme de la F.F.T ne modifiant pas ces valeurs, il suffira de les initialiser une seule fois lors de la mise sous-tension; le temps d'exécution du sous programme d'initialisation n'est donc pas une contrainte.

En utilisant toutes les symétries possibles:

$$R_e(W_{64}^{16-n}) = -\text{Im}(W_{64}^n) ; R_e(W_{64}^{16+n}) = \text{Im}(W_{64}^n) ; \text{Im}(W_{64}^{16+n}) = \text{Im}(W_{64}^{16-n})$$

il suffit de stocker les 17 valeurs $\text{Im}(W_{64}^k)$; $k= 0, 1, 2, \dots, 16$.

IV.4 PROBLEMES DE RECADRAGE

D'après le théorème de Parseval :

$$\sum_0^{N-1} |x_n|^2 = 1/N \sum_0^{N-1} |X_k|^2$$

On voit bien qu'en moyenne les valeurs de la transformée de Fourier discrètes (X_k , $k=0, 1, \dots, N-1$) sont plus grandes que les valeurs initiales (x_n ; $n=0, \dots, N-1$). Pour garder la meilleure précision tout au long du calcul, il est nécessaire de cadrer les valeurs les plus à gauche possible en mémoire.

La structure du papillon utilisé est :

$$\begin{aligned} X_{m+1}(i) &= X_m(i) + W \cdot X_m(j) \\ X_{m+1}(j) &= X_m(i) - W \cdot X_m(j) \end{aligned} \quad (i \text{ et } j \text{ variant de } 0 \text{ à } 63)$$

X_{m+1} est le résultat après le calcul des papillons au niveau m . Cette structure permet d'écrire:

$$\max_i (|X_m(i)|) \leq \max_j (|X_{m+1}(j)|) \leq 2 \cdot \max_i (|X_m(i)|)$$

A chaque étape de calcul, le module des valeurs complexes augmente d'au plus d'un facteur 2.

Plusieurs méthodes de recadrage sont possibles:

PREMIERE METHODE

On calcul le module de toutes les données, et éventuellement un recadrage avant le calcul si une des valeurs dépasse 1, soit des divisions systématiques par 2 après chaque étape de papillon, mais qui peuvent entraîner une perte inutile de précision.

DEUXIEME METHODE

Avant chaque étape de papillon on calcul le maximum des $X_m(i)$, suivant la valeur obtenue on sait si un recadrage est ou non nécessaire pour cette étape. Cette méthode remédie à l'inconvénient cité dans la première méthode, seulement un autre inconvénient surgit, qui réside dans la nécessité de calculer les module à chaque niveau du calcul de la F.F.T (64*6 soit 384 modules à calculer pour une F.F.T sur 64 points.

IV.5 ERREURS DUES AU CALCUL

Les coefficients réellement utilisés par la machine représentent une approximation des coefficients théoriques dont la valeur de la partie réelle et imaginaire est comprise dans l'intervalle $[-1, +1]$. Une représentation des nombres en mémoire sur 16 bits entraîne pour les W (et aussi pour les données) une erreur d'arrondi. Si les bits ont une signification $(S, 2^{-1}, \dots, 2^{-15})$, la variance de l'erreur d'arrondi vaut alors:

$$\Delta_{ar}^2 = 1/12 * (2^{-15})^2 = 1/3 * (2^{-16})^2$$

L'écriture des résultats des calculs en mémoire en gardant le plus de bits significatifs nécessite l'utilisation d'un facteur multiplicatif (2^A), telle que la signification des bits soit: $(S, 2^{A-1}, 2^{A-2}, \dots, 2^{A-15})$.

En faisant une division par 2^m pour le recadrage au cours du transfert de

l'accumulateur vers la mémoire) on introduit une erreur de troncature dont la variance est $\Delta_{tr}^2 = 1/2(2^{A-16} \cdot 2^m)^2$ où A représente le nombre de décalages (divisions par 2) déjà effectués avant celui-ci .

$$\frac{\Delta_{tr}^2}{\Delta_{ar}^2} = 3/2 (2^{A+m})^2$$

L'accumulateur travaillant sur 32 bits et utilisant directement les 26 bits du résultat du multiplieur, l'erreur d'arrondi est donc négligeable devant l'erreur de troncature au cours du transfert de l'accumulateur vers la mémoire .

Soit V_{n+1} la moyenne de la variance des éléments au niveau n :

$$V_{n+1} = 1/2N \sum_0^{N-1} \left[\text{Var}(\text{Re}(X_{n+1}(i))) + \text{Var}(\text{Im}(X_{n+1}(i))) \right]$$

et $V_0 = \delta^2$ la variance moyenne sur les données .

Soit d_1 le nombre de décalages effectués au niveau i pour le recadrage, en posant D_1 le nombre de décalages total effectués jusqu'à ce niveau, on a alors :

$$D_1 = d_0 + d_1 + \dots + d_i .$$

Comme les niveaux 0 et 1 ne font intervenir que des sommes et des différences on a donc :

$$V_1 = V_0 + V_0 + 1/2 2^{2d_0} \cdot (2^{-16})^2$$

$$V_2 = V_1 + V_1 + 1/2 2^{2(d_0 + d_1)} \cdot (2^{-16})^2 .$$

Pour les niveaux 2, 3,, 5, on effectue les calculs suivants :

$$\text{Re}(X_{n+1}(i)) = \text{Re}(X_n(i)) + \text{Re}(W) \cdot \text{Re}(X_n(j)) - \text{Im}(W) \cdot \text{Im}(X_n(j))$$

$$\text{Im}(X_{n+1}(i)) = \text{Im}(X_n(j)) + \text{Im}(W) \cdot \text{Re}(X_n(j)) + \text{Re}(W) \cdot \text{Im}(X_n(j))$$

D'où :

$$V_{n+1} = V_n + V_n + (\text{moyenne}/W)^2 + \text{Var}(W) \cdot (\text{moyenne}/X_n(j))^2 + \text{erreur de troncature}$$

Notons par K la valeur moyenne du carré du module du signal en entrée de la F.F.T

$$K = 1/N \sum_0^{N-1} |X_0(i)|^2$$

Ce qui nous permet d'écrire :

$$1/N \sum_0^{N-1} |X_{n+1}(i)|^2 = 2 \cdot 1/N \sum_0^{N-1} |X_n(i)|^2 = 2^{n+1} \cdot K$$

L'expression de V_{n+1} devient alors :

$$\begin{aligned} V_{n+1} &= V_n + V_n + 1/3(2^{-16})^2 \cdot 2^n K + 1/2 \cdot 2^{2D_n}(2^{-16})^2 \\ &= 2V_n + 1/3 \cdot 2^n \cdot K (2^{-16})^2 + 1/2 \cdot 2^{2(d_0+d_1+\dots+d_n)} \cdot (2^{-16})^2 \end{aligned}$$

La variance V sur le résultat (pour une F.F.T sur 64 points complexes) est égale à V_6 soit :

$$V_6 = 2 V_5 + 1/3 \cdot 2^5 K (2^{-16})^2 + 1/2 \cdot 4^{(d_0+d_1+d_2+\dots+d_5)} \cdot (2^{-16})^2$$

$$V_5 = 2 V_4 + 1/3 \cdot 2^4 \cdot K (2^{-16})^2 + 1/2 \cdot 4^{(d_0+d_1+d_2+d_3+d_4)} \cdot (2^{-16})^2$$

$$V_4 = 2 V_3 + 1/3 \cdot 2^3 \cdot K (2^{-16})^2 + 1/2 \cdot 4^{(d_0+d_1+d_2+d_3)} \cdot (2^{-16})^2$$

$$V_3 = 2 V_2 + 1/3 \cdot 2^2 \cdot K (2^{-16})^2 + 1/2 \cdot 4^{(d_0+d_1+d_2)} \cdot (2^{-16})^2$$

$$V_2 = 2 V_1 + 1/3 \cdot 2^1 \cdot K (2^{-16})^2 + 1/2 \cdot 4^{(d_0+d_1)} \cdot (2^{-16})^2$$

$$V_1 = 2 V_0 + 1/3 \cdot 2^0 \cdot K (2^{-16})^2 + 1/2 \cdot 4^{(d_0)} \cdot (2^{-16})^2$$

Soit :

$$V = 64 \delta^2 + 128/3 \cdot K (2^{-16})^2 + (2^{-16})^2 / 2 \cdot 4^{d_0} (32 + 4^{d_1} (16 + 4^{d_2} (8 + 4^{d_3} (4 + 4^{d_4} (2 + 4^{d_5}))))))$$

L'erreur notée \mathcal{E} sur le résultat de la F.F.T peut être présentée sous forme d'un rapport bruit/signal :

$$\mathcal{E}^2 = V/32K \simeq 2 \delta^2 / K + (2^{-16})^2 / 64K \cdot V'(d_0, d_1, \dots, d_5) \quad \text{où}$$

$$V'(d_0, d_1, \dots, d_5) = 4^{d_0} (32 + 4^{d_1} (16 + 4^{d_2} (8 + 4^{d_3} (4 + 4^{d_4} (2 + 4^{d_5})))))) .$$

Le premier terme provient de l'erreur de quantification sur les données, par contre le second terme correspond aux erreurs dues aux calculs .

Si $d_0, \dots, d_5 = 1$ on trouve :

$$\mathcal{E}'^2 \simeq \frac{64 (2^{-16})^2}{K/2}, \text{ d'où une erreur } \mathcal{E}' \simeq \frac{8(2^{-16})^2}{K/2} \text{ si l'erreur de quantification}$$

est négligeable celle-ci .

Si un des $d_i = 0$, l'erreur varie alors de :

$$\mathcal{E}'_0 = \frac{4(2^{-16})^2}{K/2} \text{ pour } d_0 = 0 \text{ à } \mathcal{E}'_5 = \frac{12}{k/2} (2^{-16})^2 \text{ pour } d_5 = 5$$

Il est donc souhaitable d'éviter les divisions par 2 si celles-ci sont unitiles .

V. F.F.T POUR UN SIGNAL REEL EN ENTREE

Lors du calcul de la transformée d'un signal réel, il est possible d'utiliser la symétrie hermitienne du spectre pour réduire le temps de calcul et la place mémoire nécessaire .

V.1. ALGORITHME DE CALCUL :

La suite réelle x_k ; $k=0, \dots, 2N-1$ a pour transformée de Fourier :

$$X_n = \sum_0^{2N-1} x_k \cdot W_{2N}^{nk} ; n=0, \dots, N .$$

La suite x_k étant réelle , on a la symétrie $X_{2N-n} = \overline{X_n}$; il suffit donc de calculer X_n ; $n=0, \dots, N$ pour connaître tout le spectre discret .

On peut décomposer X_n sous la forme :

$$X_n = \sum_0^{N-1} x_{2k} \cdot W_N^{nk} + W_{2N}^n \left(\sum_0^{N-1} x_{2k+1} \cdot W_N^{nk} \right)$$

Soit :

$$X_n = G_n + W_{2N}^n \cdot H_n$$

Or en appliquant la transformée de Fourier à la suite complexe $(x_{2k} + j x_{2k+1})$, on obtient :

$$Y_n = \sum_0^{N-1} (x_{2k} + j x_{2k+1}) W_N^{nk} = G_n + j H_n ; n=0, \dots, N-1 .$$

En utilisant les relations :

$$Y_n = G_n + j H_n \quad \text{et} \quad \overline{Y_{N-n}} = G_n - j H_n$$

on peut déduire facilement les G_n et les H_n pour $n=0, \dots, N/2 - 1$, d'où les X_n ; $n=0, \dots, N/2-1, N/2+1, \dots, N-1$ en effectuant une dernière étape de "pseudo-papillons" :

$$X_n = G_n + W_{2N}^n H_n$$

$$X_{N-n} = G_n - W_{2N}^n H_n$$

Pour X_0 , $X_{N/2}$ et X_N on tient compte du fait que G_0 , H_0 , $G_{N/2}$ et $H_{N/2}$ sont réels

$$\begin{aligned} \text{d'où : } X_0 &= G_0 + H_0 \\ X_N &= G_0 - H_0 \\ X_{N/2} &= Y_{N/2} \end{aligned}$$

V.2 MISE EN OEUVRE SUR LE TMS 320.10

La mémoire RAM disponible sur le TMS 320.10 permet de réaliser une F.F.T sur 64 points réels, en sortie, on obtient le résultat sur 31 complexes (X_1, \dots, X_{31}) et 2 réels (X_0, X_{32}).

Pour le calcul, on effectue au départ une F.F.T sur les 32 points complexes ($x_{2k} + j x_{2k+1}$) ce qui permet d'obtenir les Y_n ; $n=0, \dots, 31$.

La fin du calcul est décomposée en 2 étapes, la première consiste à calculer les G_n et les H_n , la seconde à faire un calcul de papillons pour obtenir les X_n .

Pour la sortie des résultats, il faut faire attention à l'ordre et à la place des X_n à la fin du calcul.

PROGRAMME

TABLE 2- TMS32010 INSTRUCTION SUMMARY

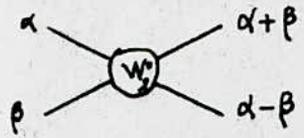
ACCUMULATOR INSTRUCTIONS			
MNEMONIC	DESCRIPTION	NO. OF CYCLES	NO. OF WORDS
ADD	Add to accumulator with shift	1	1
SUB	Subtract from accumulator with shift	1	1
LAC	Load accumulator with shift	1	1
SACL	Store low order accumulator bits	1	1
SACH	Store high order accumulator bits with shift	1	1
ADDH	Add to high-order accumulator bits	1	1
ADDS	Add to accumulator with no sign extension	1	1
SUBH	Subtract from high-order accumulator bits	1	1
SUBS	Subtract from accumulator with no sign extension	1	1
SUBC	Conditional subtract (for divide)	1	1
ZATH	Zero accumulator and load high order bits	1	1
ZALS	Zero accumulator and load low order bits	1	1
LACI	Load accumulator immediate	1	1
ABS	Absolute value of accumulator	1	1
ZAC	Zero accumulator	1	1
XOR	Exclusive OR with accumulator	1	1
AND	AND with accumulator	1	1
OR	OR with accumulator	1	1
AUXILIARY REGISTERS AND DATA PAGE POINTER INSTRUCTIONS			
MNEMONIC	DESCRIPTION	NO. OF CYCLES	NO. OF WORDS
SAR	Store auxiliary register	1	1
LAR	Load auxiliary register	1	1
MAR	Modify auxiliary register and pointer	1	1
LDPK	Load data memory page pointer immediate	1	1
LDP	Load data memory page pointer	1	1
LARK	Load auxiliary register immediate	1	1
LARP	Load auxiliary register pointer immediate	1	1
T REGISTER, P REGISTER, AND MULTIPLY INSTRUCTIONS			
MNEMONIC	DESCRIPTION	NO. OF CYCLES	NO. OF WORDS
LT	Load T Register	1	1
LTA	Load T Register and accumulate product	1	1
LTD	Load T Register, accumulate product, and move data in memory forward one address	1	1
MPY	Multiply with T Register, store product in P Register	1	1
PAC	Load accumulator from P Register	1	1
APAC	Add P Register to accumulator	1	1
SPAC	Subtract P Register from accumulator	1	1
MPYK	Multiply T Register with immediate operand, store product in P Register	1	1

TABLE 2 - TMS32010 INSTRUCTION SUMMARY (CONTINUED)

BRANCH INSTRUCTIONS			
MNEMONIC	DESCRIPTION	NO. OF CYCLES	NO. OF WORDS
BANZ	Branch on auxiliary register not zero	2	2
BV	Branch on overflow	2	2
BIOZ	Branch on $\overline{BIO} = 0$	2	2
B	Branch unconditionally	2	2
BLZ	Branch if accumulator < 0	2	2
BLEZ	Branch if accumulator ≤ 0	2	2
BGZ	Branch if accumulator > 0	2	2
BGEZ	Branch if accumulator ≥ 0	2	2
BNZ	Branch if accumulator $\neq 0$	2	2
BZ	Branch if accumulator $= 0$	2	2
CALL	Call subroutine immediate	2	2
CALA	Call subroutine from accumulator	2	1
RET	Return from subroutine	2	1
CONTROL INSTRUCTIONS			
MNEMONIC	DESCRIPTION	NO. OF CYCLES	NO. OF WORDS
LST	Load status register	1	1
SST	Store status register	1	1
NOP	No operation	1	1
DI	Disable interrupt	1	1
EI	Enable interrupt	1	1
ROVM	Reset overflow mode	1	1
SOVM	Set overflow mode	1	1
POP	Pop stack to accumulator	2	1
PUSH	Push stack from accumulator	2	1
I/O AND DATA MEMORY OPERATIONS			
MNEMONIC	DESCRIPTION	NO. OF CYCLES	NO. OF WORDS
IN	Input data from port	2	2
OUT	Output data to port	2	2
TBLR	Table read from program memory to data RAM	3	1
TBLW	Table write from data RAM to program memory	3	1
DMOV	Shift contents of data memory forward one address	1	1

NIVEAU: 0

A ce niveau on a 16 paquets de papillons utilisant la même constante W_2^0 .
($16 * W_2^0$).



Le programme de calcul correspondant est le suivant:

```

ZALS DAT R_e(alpha)
ADDS DAT R_e(beta)
SACL DAT 130
ZALS DAT R_e(alpha)
SUBS DAT R_e(beta)
SACL DAT R_e(beta)
ZALS DAT 130
SACL DAT R_e(alpha)

```

Ce programme sera utilisé pour tous les papillons de ce niveau, en tenant compte des entrées α et β qui changent d'un papillon à un autre, et qui sont rangées dans la RAM dans l'ordre du "bit reverse".

Les résultats de chaque papillon sont rangés à la place des données correspondantes qui seront ensuite utilisés comme les entrées des papillons du niveau suivant.

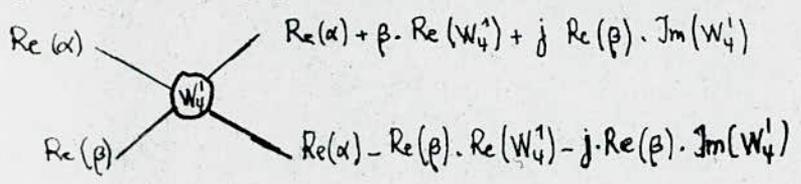
NIVEAU: 1

A ce niveau on a:

$$8 * \{ W_4^0, W_4^1 \}$$

Le programme de calcul des papillons utilisant les constantes W_4^0 est identique à celui du niveau 0.

Tandis que celui des papillons utilisant les constantes W_4^1 est différent:



LT DAT $R_e(\beta)$
 MPY DAT $R_e(W_4^1)$
 PAC
 ADDS DAT $R_e(\alpha)$
 SACL DAT 130
 LT DAT $R_e(\beta)$
 MPY DAT $R_e(W_4^1)$
 ZAC
 SPAC
 ADDS DAT $R_e(\alpha)$
 SACL DAT 131
 LT DAT $R_e(\beta)$
 MPY DAT $I_m(W_4^1)$
 PAC
 SACL DAT $I_m(\alpha)$
 ZAC
 SPAC
 SACL DAT $I_m(\beta)$
 ZALS DAT 130
 SACL DAT $R_e(\alpha)$
 ZALS DAT 131
 SACL DAT $R_e(\beta)$

NIVEAU: 2

A ce niveau on a:

$$4 * \{ W_8^0, W_8^1, W_8^2, W_8^3 \}$$

Les programmes de calcul des papillons utilisant les constantes W_8^0, W_8^2 sont identi-
 -ques respectivement à ceux des papillons utilisant les constantes W_4^0, W_4^1 .
 Le programme de calcul du papillon utilisant la constante W_8^1 ou W_8^3 est le suivant

$$\begin{array}{l}
 R_e(\alpha) + jI_m(\beta) \\
 R_e(\alpha) + jI_m(\beta)
 \end{array}
 \begin{array}{c}
 \diagup \\
 \textcircled{W_8'} \\
 \diagdown
 \end{array}
 \begin{array}{l}
 [Re(\alpha) + Re(\beta) \cdot Re(W_8') - Im(\beta) \cdot Im(W_8')] + j [Im(\alpha) + Im(\beta) \cdot Re(W_8') + Re(\beta) \cdot Im(W_8')] \\
 [Re(\alpha) - Re(\beta) \cdot Re(W_8') + Im(\beta) \cdot Im(W_8')] + j [Im(\alpha) - Re(W_8') \cdot Im(\beta) - Re(\beta) \cdot Im(W_8')]
 \end{array}$$

LT DAT I_m(W₈¹)

MPY DAT I_m(β)

ZAC

SPAC

LT DAT R_e(W₈¹)

MPY DAT R_e(β)

APAC

ADDS DAT R_e(α)

SACL DAT 130

LT DAT I_m(W₈¹)

MPY DAT I_m(β)

PAC

SACL DAT 134

LT DAT R_e(W₈¹)

MPY DAT R_e(β)

ZAC

SPAC

ADDS DAT 134

ADDS DAT R_e(α)

SACL DAT 131

LT DAT I_m(W₈¹)

MPY DAT R_e(β)

PAC

LT DAT R_e(W₈¹)

MPY DAT I_m(β)

APAC

ADDS DAT $I_m(\alpha)$
 SACL DAT 132
 LT DAT $I_m(w_8^1)$
 MPY DAT $R_e(\beta)$
 PAC
 LT DAT $I_m(\beta)$
 MPY DAT $R_e(w_8^1)$
 APAC
 SACL DAT 133
 ZAC
 SUBS DAT 133
 ADDS DAT $I_m(\alpha)$
 SACL DAT $I_m(\beta)$
 ZAIS DAT 132
 SACL DAT $I_m(\alpha)$
 ZAIS DAT 131
 SACL DAT $R_e(\beta)$
 ZAIS DAT 130
 SACL DAT $R_e(\alpha)$

NIVEAU: 3

A ce niveau on a;

$$2 * \{ w_{15}^0, w_{15}^1, w_{15}^2, \dots, w_{16}^7 \}$$

Les programmes de calcul des papillons ayant les constantes $w_{16}^1, w_{16}^2, w_{16}^3, w_{16}^5, w_{16}^6$ et w_{16}^7 sont identiques à celui du papillon utilisant la constante w_8^1 .

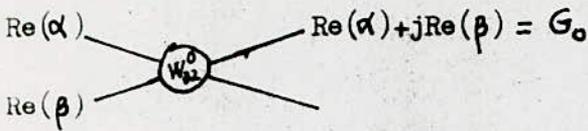
Par contre le programme de calcul des papillons ayant les constantes w_{16}^0 et w_{16}^4 sont identiques respectivement à ceux des papillons utilisant les constantes w_2^0 et w_4^1 .

NIVEAU: 4

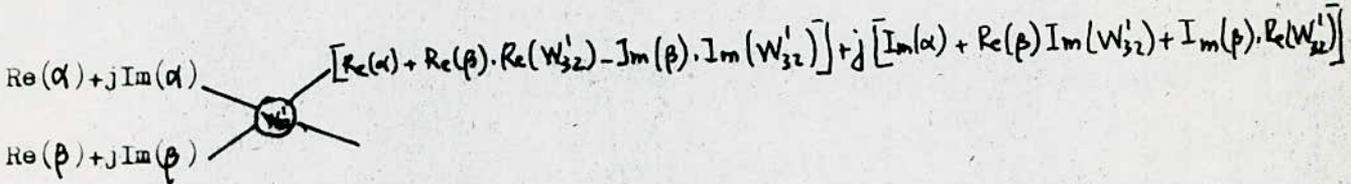
A ce niveau on a :

$$1 * \{ W_{32}^0, W_{32}^1, \dots, W_{32}^{15} \}$$

Les programmes ci-dessous nous permettent de calculer les sorties $G_n, n=0,1,\dots,15$



```
ZALS DAT Re(alpha)
ADDS DAT Re(beta)
SACL DAT Re(alpha)
```



```
LT DAT Im(W_{32}^1)
MPY DAT Im(beta)
ZAC
SPAC
LT DAT Re(W_{32}^1)
MPY DAT Re(beta)
APAC
ADDS DAT Re(alpha)
SACL DAT Re(alpha)
LT DAT Re(W_{32}^1)
MPY DAT Im(beta)
PAC
LT DAT Im(W_{32}^1)
MPY DAT Re(beta)
APAC
ADDS DAT Im(alpha)
SACL DAT Im(alpha)
```

Le calcul des sorties G_n avec $n=2, \dots, 15$, se fait en utilisant ce même programme.

Le calcul des H_n avec $n=0,1,\dots,15$, se fait exactement de la même façon que celui des G_n .

Calcul des X_n : $n=1,\dots,N/2-1$

Calcul des X_{N-n} : $n=N/2+1,\dots,N-1$.

$$X_n = G_n + W_{64}^n \cdot H_n$$

$$\text{Re}(X_n) = \text{Re}(G_n) + \text{Re}(H_n) \cdot \text{Re}(W_{64}^n) - \text{Im}(H_n) \cdot \text{Im}(W_{64}^n).$$

$$\text{Im}(X_n) = \text{Im}(G_n) + \text{Re}(W_{64}^n) \cdot \text{Im}(H_n) + \text{Im}(W_{64}^n) \cdot \text{Re}(H_n).$$

$$\text{Re}(X_{N-n}) = \text{Re}(G_n) - \text{Re}(W_{2N}^n) \cdot \text{Re}(H_n) + \text{Im}(W_{2N}^n) \cdot \text{Im}(H_n).$$

$$\text{Im}(X_{N-n}) = \text{Im}(W_{2N}^n) \cdot \text{Re}(H_n) + \text{Re}(W_{2N}^n) \cdot \text{Im}(H_n) - \text{Im}(G_n).$$

Programme de calcul de X_n et X_{N-n} :

Calcul du réel de X_n : $\text{Re}(X_n)$

```

LT   DAT Re(H_n)
MPY  DAT Re(W_64^n)
PAC
LT   DAT Im(H_n)
MPY  DAT Im(W_64^n)
SPAC
ADDS DAT Re(G_n)
SACL DAT 130

```

Calcul de l'imaginaire de X_n : $\text{Im}(X_n)$

```

LT   DAT Re(H_n)
MPY  DAT Im(W_64^n)
PAC
LT   DAT Im(H_n)
MPY  DAT Re(W_64^n)
APAC
ADDS DAT Im(G_n)
SACL DAT 131

```

Programme de calcul de X_0 :

$$X_0 = \text{Re}(G_0) + \text{Re}(H_0) = G_0 + H_0$$

```

ZALS DAT Re(G_0)
ADDS DAT Re(H_0)
SACL DAT Re(G_0)

```

Calcul du réel de X_{N-n}

```

LT   DAT Im(Hn)
MPY  DAT Im(W64n)
PAC
LT   DAT Re(Hn)
MPY  DAT Re(W64n)
SPAC
ADDS DAT Re(Gn)
SACL DAT 132
    
```

Calcul de l'imaginaire de X_{N-n}

```

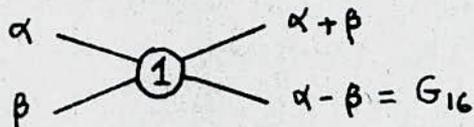
LT   DAT Re(Hn)
MPY  DAT Im(W64n)
PAC
LT   DAT Re(W64n)
MPY  DAT Im(Hn)
APAC
SUBS DAT Im(Gn)
SACL DAT Im(Hn)
ZALS DAT 132
SACL DAT Re(Hn)
ZALS DAT 131
SACL DAT Im(Gn)
ZALS DAT 130
SACL DAT Re(Gn)
    
```

Calcul de X_{16} :

Pour calculer X_{16} , il est nécessaire de calculer G_{16} et H_{16} puisque X_{16} est liée à Y_{16} par la relation :

$$X_{16} = \overline{Y_{16}} \text{ et d'autre part:}$$

$\overline{Y_{16}} = G_{16} - jH_{16}$; sachant que H_{16} et G_{16} sont réels, le papillon utilisé pour le calcul de G_{16} est le suivant :



Le programme correspondant est :

```

ZALS DAT Re(α)
SUBS DAT Re(β)
SACL DAT Re(Ho)
    
```

Le même programme sera utilisé pour le calcul de H_{16} qui sera stocké dans l'adresse mémoire de la partie imaginaire de H_0 .

Programme pour le calcul de X_{16} .

$$X_{16} = G_{16} - H_{16}$$

ZALS DAT Re(H_0)

SUBS DAT Im(H_0)

SACL DAT Re(H_0)

ZAC

SACL DAT Im(H_0)



Conclusion:

Le temps de calcul d'une F.F.T sur 64 points est résumé comme suit :

Niveau 0 :

16* W_2^0 : 128 instructions

Niveau 1 :

8* W_4^0, W_4^1 : 240 instructions

Niveau 2 :

4* $W_8^0, W_8^1, W_8^2, W_8^3$: 480 instructions

Niveau 3 :

2* $W_{16}^0, W_{16}^1, \dots, W_{16}^7$: 600 instructions

Niveau 4 :

1* $W_{32}^0, \dots, W_{32}^{15}$: 1100 instructions

Le nombre total d'instructions nécessaire pour le calcul de la F.F.T sur 64 points est donc 2548 instructions . Comme chaque cycle d'instruction exige 200 ns , le temps de calcul de cette F.F.T sera d'environ 510 us auquel il faut ajouter le temps de chargement des constantes .

CHAPITRE VI

V/. ANALYSE CEPSTRALE SUR LE TMS 320 10

Dans ce chapitre, on propose une méthode de calcul du cepstre pour sa mise en oeuvre sur le u- processeur . Une fréquence d'échantillonnage de 12,8 KHZ et une analyse par blocs de 40 ms correspond à 512 échantillons , ce qui se traduirait par une taille mémoire nécessaire très importante . Cependant on peut se ramener au calcul d'une F.F.T sur 128 points réels au lieu de 512 , de façon qu'une seule RAM extérieure de 144 mots .

VI.1 ADAPTATION DE L'ALGORITHME D'ANALYSE

La solution consiste à faire une partie de l'analyse sur des blocs de 128 points (soit 10ms) toutes les 5ms, puis faire un lissage tutes les 20ms sur les valeurs obtenues pour 8 blocs successifs (40 ms) .

Pour l'analyse, on fait donc toutes les 5 ms une préaccentuation numérique , un fenêtrage , une F.F.T sur 128 points réels , un calcul des énergies dans les filtres MEL et toutes les 20 ms , un lissage , un calcul du logarithme , une transformée en cosinus .

VI.2 ECHELLE MEL

L'échelle spectrale dans le calcul du cepstre est linéaire en fréquence, cependant l'oreille est sensible à une échelle logarithmique de la fréquence, les informations pertinentes se trouvent réparties de façon exponentielle dans les zones éloignées .

L'échelle logarithmique tente d'éviter de donner un même poids à des zones de fréquences qui n'ont pas la même densité d'informations .

C'est pour cette raison qu'on utilise une échelle quasi-logarithmique appelée "échelle MEL" qui est linéaire sur la bande (0-1KHZ) et logarithmique au delà . Dans l'échelle MEL les coefficients cepstraux sont donnés par l'équation :

$$MF_{cc}(i) = 1/NF \sum \text{Log}(E(K) \cdot \cosn(K-i) \cdot \pi/NF) ; \quad 1 \leq i \leq M$$

n : rang du coefficient cepstral

K : numéro du filtre

NF : nombre de filtres triangulaires (en géeéral NF = 25)

M : nombre de coefficients cepstraux

E(K) : réponse en énergie des filtres triangulaires .

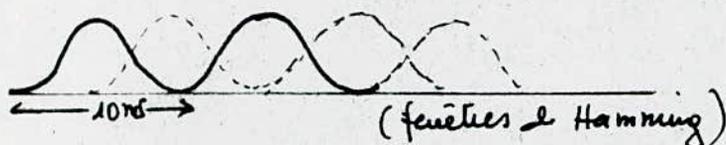
VI.3 PROBLEME DU LISSAGE

Pour le calcul on conserve une fréquence d'échantillonnage de 12,8 KHZ et un recouvrement de moitié des blocs d'analyse de 128 points .

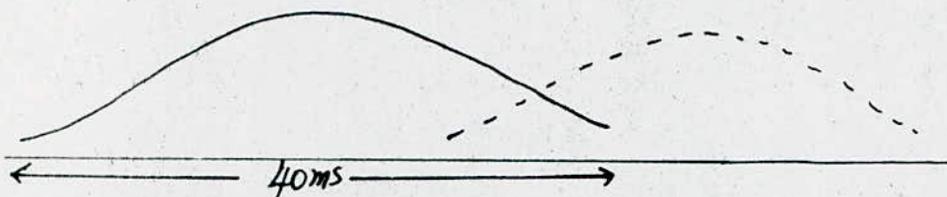
Pour que les paramètres calculés correspondent à une durée de parole de l'ordre de 40 ms , on fait un lissage sur 8 blocs successifs .

Cette méthode de lissage consiste en une moyenne pondérée des résultats obtenus sur 8 blocs successifs de 128 points, avec pour coefficients les valeurs $1/4$, $1/3$, $3/4$, 1 , 1 , $3/4$, $1/2$, $1/4$.

Avec les blocs de 128 points , on a le recouvrement suivant :



auquel il faut ajouter le lissage :



VI.4 MISE EN OEUVRE SUR LE U-PROCESSEUR

VI.4.1 CHARGEMENT DES CONSTANTES

De façon à limiter la place nécessaire pour le stockage des constantes dans la ROM programme, on essaye d'utiliser au mieux toutes les symétries possibles. EN fait il suffit de stocker $\text{Re}(W_{128}^k)$, $k=0, \dots, 32$ pour obtenir tous les W nécessaires pour la F.F.T à partir des symétries existantes entre les cosinus et les sinus .

VI.4.2 FILTRES MEL

Le calcul des énergies dans les filtres MEL nécessite presque d'écrire un sous programme par type de filtre (en raison du nombre assez faible de points où on connaît le spectre).

C O N C L U S I O N

Ce travail nous a permis d'élargir nos connaissances dans le domaine du traitement de la parole , qui est actuellement un nouveau foyer de recherches.

L'idée générale qui a guidé ce travail a été d'envisager l'utilisation d'un microprocesseur dans le domaine du traitement du signal.

L'étude faite montre qu'il est possible d'envisager l'utilisation du microprocesseur TMS 320.10 pour réaliser l'analyse cepstrale en vue de la reconnaissance de la parole. La place disponible en mémoire n'est suffisante que pour le calcul d'une F.F.T sur 54 points réels. Comme on peut utiliser une extension de mémoire de 8 K-bytes , il est donc possible d'envisager le calcul des logarithmes des énergies dans les filtres MEL, et également le calcul de la transformée en cosinus et le lissage.

Le recouvrement de moitié des blocs de 128 points autorise 5 ms pour faire les calculs, or la plus grande part de ceux-ci (F.F.T) peut être faite en (1 ms), ce qui laisse suffisamment de temps pour faire le reste.

BIBLIOGRAPHIE

- 1/ M. BELLAGER
"Traitement numérique du signal" Masson 1981
- 2/ J. MAX
"Méthodes et techniques de traitement du
signal et applications aux mesures physiques" Masson 1981
Volumes I et II
- 3/ J. LIFERMANN
"Les méthodes rapides de transformation du signal" Masson 1980
- 4/ M. KUNT
"Traitement numérique des signaux" DUNOD 1981
- 5/ J. GUIBERT
"La parole, compréhension et synthèse par les
ordinateurs." 1979
- 6/ M. FERRETTI-F. CINAREL
"Synthèse, reconnaissance de la parole" TESTS 1983
- 7/ M. AUMIAUX
"L'emploi des microprocesseurs" Masson 1982
- 8/ T M S 320.10
"Assembly langage programmers guide" TEXAS INSTRUMENTS
1983

