République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Université Amar Thelidji- Laghouat

FACULTE: DE TECHNOLOGIE

DEPARTEMENT D'ELECTRONIQUE

MEMOIRE DE MASTER

Réalisé par :

- **Abdi Hadjer**
- **Ben ghechoua Asma**

Domaine : Sciences et Technologies

Filière : Télécommunication

Option : Réseaux et télécommunications

Thème

La carte DSP TMS320 C6713 DSK pour la conception des filtres numériques

Jury de soutenance :

Nom et Prénom	Grade	Qualité
Reguiegue Mourad	MCB	Encadrant
Chaker Mohcene Saleh	MCB	Président
Koriba Mustapha	MAA	Examinateur

Promotion : 2021/2022

ملخص

معالج الاشارات الرقمية (DSP) هو معالج متخصص في معالجة الإشارات الرقمية. تم تحسين بنيته لمعالجة كمية كبيرة من البيانات. يعتبر وضع التشغيل هذا فعالاً للغاية في معالجة الإشارات الرقمية (التصفية والضغط والاستخراج وما إلى ذلك) مثل الفيديو أو الموسيقى.

تستخدم معالجات الاشارات الرقمية في معظم تطبيقات معالجة الإشارات الرقمية في الوقت الفعلي. يهدف مشروع نهاية الدراسة هذا إلى دراسة تصميم المرشحات الرقمية بواسطة DSP TMS320C6713 وكذلك تم استغلال هاته البطاقة ، حتى نتمكن من دراسة هذا المرشح بطريقة عملية.

كلمات مفتاحية: معالجة الإشارات الرقمية - Matlab - DSP - Code composer

Résumé

Le DSP est un processeur spécialisé dans le traitement du signal numérique. Son architecture est optimisée pour traiter une grande quantité de données en parallèle à chaque cycle d'horloge. Ce mode de fonctionnement est très efficace pour traiter les signaux numériques (filtrer, compresser, extraire, etc.) comme la vidéo ou la musique.

Les DSP sont utilisés dans la plupart des applications de traitement numérique du signal en temps réel. Ce projet de fin d'étude vise à étudier la conception des filtres numériques par la DSP TMS320C6713. Cette carte d'évaluation a été exploitée, afin de pouvoir étudier ces filtres numériques de manière pratique.

Mots clés : Traitement numérique du signal, DSP, Code composer studio, Matlab.

Abstract

The DSP is a processor specialized in digital signal processing. Its architecture is optimized to process a large amount of data in parallel at each clock cycle. This operating mode is very effective in the processing of digital signals (filtering, compression, extraction, etc.) such as video or music.

DSPs are used in most real-time digital signal processing applications. This graduation project aims to study the design of digital filters by DSP TMS320C6713 as well as the evaluation board TMS320C6713DSK for digital filters, so that we can study this filter in a practical way.

Dédicaces

- *A mon chères mère :*

Vous avez su porter pour nous les soins et consentir les efforts pour notre éducation. Aucune dédicace ne saurait exprimer tout le respect et l'amour que nous vous portons, vous nous avez toujours fait confiance. Veuillez trouvez en ce travail la consolation et le témoin de la patience, Que Dieu ait pitié toi, mère, et j'espère que paradis sera a toi .

- *A mon cher père :*

Malgré les grandes responsabilités que vous assumez dans vos travaux ou autant que pères de familles, vous avez toujours été près de nous, pour nous écouter, nous soutenir, nous suivre et nous encourager. Puisse ce travail diminuer vos souffrances et vous porter bonheur.

- *A mon chers frères et sœur :*

j'avais réservons la plus grande partie de ce travail. Vous avez toujours été pour moi d'une aide précieuse. j'ai vous remercions pour tous les bienfaits que chacun a pu faire pour moi.

- *A nos familles.*
- *A tous nos enseignants de département D'electronique .*
- *A tous ceux qui ont eu et qui ont confiance en moi.*
- *A mon binôme **Asma** et tous mes amis.*

Hadjer

Dédicaces

Je dédie cet humble travail avec beaucoup d'amour, de sincérité et de fierté :

À mes chers parents, Ali et Zohra, pour leur sacrifice, leur amour, leur tendresse, leur soutien et leurs prières tout au long de ma période d'études.

A ma chère grand-mère Messaouda pour ses prières pour moi

A mes chères soeurs Saida et ses enfants, Iman.

A mes chers frères Hossam, Hassan, Houssein et Abderrahim.

Et à tous les membres de ma famille.

A mon binôme ***Hadjer*** *et tous mes amis.*

Asma

Remerciements

Nous remercions Allah, qui ma donné la force, le courage et la bonne volonté de faire.

Au terme de ce travail, Nous adressons mos remerciements les plus sincères à notre encadreur **Mr Reguiegue Mourad** Professeur à l'université de **Amar Telidji-Laghouat**, pour nous avoir permis de bénéficier de son grande savoir dans la matière et son aide précieux tout au long de ce travail.

On tiens à remercier tout personne ayant participé, de loin ou de prêt à la réalisation de ce travail.

Nous remercions particulièrement les membres de jury d'avoir l'amabilité de présider le jury et de nous avoir accordé l'honneur d'accepter l'évaluation de ce mémoire.

Pour finir, nous avons une pensée toute particulière pour nos parents et nos proches qui n'ont jamais cessé de nous fournir l'essentiel.

A jamais merci pour pousser à toujours Dépasser nos limites.

Table des matières

Liste des figures

Chapitre I : Filtrages Numériques

Chapitre II : La carte TMS320C6713 DSK

Chapitre III *: Résultats et Discussions*

Liste des tableaux

Chapitre I : Filtrages Numériques

Glossaire

Acronyme	Description
RIF	Réponse Impulsionnelle Finie
RII	Réponse Impulsionnelle Infinie
TFD	Transformée de Fourier Discrète
CAN	Conversion Analogique - Numérique
CNA	Conversion Numérique - Analogique
DSP	Digital Signal Processor
DSK	Digital Signal kit
CPU	Central Processing Unit
DRAM	Dynamic Random Access Memory
SDRAM	Synchronous Dynamic Random Access
CPLD	Complex Programmable Logic Devices
JTAG	Joint Test Action Group
DMA	Direct Memory Access
EDMA	Enhanced Direct Memory Access
EMIF	External Memory Interface
McBSP	Multichannel Buffered Serial Port
HPI	Host Port Interface
IRAM	Internal Random Access Memory
CCS	Code Composer Studio
IDE	Environnement de Développement Intégré

Introduction Générale

Introduction Générale

Le signal est support de l'information émise par une source est destinée à un récepteur ; c'est le véhicule de l'intelligence dans les systèmes. Il transporte les ordres dans les équipements de contrôle et de télécommande, il achemine sur les réseaux l'information, la parole ou l'image [1, 2].

Les filtres numériques peuvent être classifiés selon la réponse impulsionnelle en deux types : les filtres numériques à réponse impulsionnelle finie **R.I.F** (les filtres non récursifs) et les filtres numériques à réponse impulsionnelle infinie **R.I.I** (les filtres récursifs) [1].

Un processeur de signal numérique (DSP) est une puce de microprocesseur spécialisée, dont l'architecture est optimisée pour les besoins opérationnels du traitement du signal numérique, ils sont largement utilisés dans le traitement du signal audio, les télécommunications, le traitement numérique de l'image et les systèmes de reconnaissance vocale, ainsi que dans les appareils électroniques grand public courants tels que les téléphones mobiles, les lecteurs de disque et les produits de télévision haute définition (TVHD).

Le but d'un DSP est généralement de mesurer, filtrer ou compresser des signaux analogiques continus du monde réel [2]. La plupart des microprocesseurs à usage général peuvent également exécuter avec succès des algorithmes de traitement du signal numérique, mais peuvent ne pas être en mesure de suivre ce traitement en continu en temps réel. En outre, les DSP dédiés ont généralement une meilleure efficacité énergétique, ils conviennent donc mieux aux appareils portables tels que les téléphones portables en raison des contraintes de consommation d'énergie.

Les DSP utilisent souvent des architectures de mémoire spéciales qui sont capables d'extraire plusieurs données ou instructions en même temps. Les DSP mettent souvent également en œuvre une technologie de compression.

L'objectif principal de ce projet est l'implémentation des filtres numériques à l'aide d'une carte DSP TMS320 C6713 DSK ou nous devons réaliser la conception d'un filtrage numérique à réponse impulsionnelle finie **R.I.F** (filtre passe-bas, filtre passe-haut, filtre passe bande, filtre coupe bande) afin de filtré un signal audio.

Pour concrétiser l'objectif de notre mémoire, nous avons organisé notre travail en trois chapitres de la manière suivante :

Dans le premier chapitre, nous avons présenté quelques généralités sur le traitement numériques du signal et on a vu les différentes méthodes de conception des deux types de filtres R.I.F et R.I.I.

Dans le deuxième chapitre, on a mis en avant les notions principales des DSP, ont ce basant sur l'architecture interne, et les caractéristiques du DSP TMS320C6713 qui sera exploité dans notre travail. Puis nous avons présenté l'outil software afin d'exploité notre carte DSP, cet outil est le Code Composer Studio CCS. C'est un environnement de développement intégré (IDE) utile pour créer et déboguer des programmes. De plus, il permet une analyse en temps réel des applications.

Dans le troisième chapitre, nous allons illustrer les étapes à suivre on exploitant la carte DSP TMS320 C6713 DSK pour appliquer différents types de filtres à un signal audio.

On clôturera ce mémoire par une conclusion générale tout en proposant des perspectives pour la continuité de ce travail.

Chapitre I :

Filtrages numériques

Sommaire

1.1 Introduction

Le signal est support de l'information émise par une source est destinée à un récepteur ; c'est le véhicule de l'intelligence dans les systèmes. Il transporte les ordres dans les équipements de contrôle et de télécommande, il achemine sur les réseaux l'information, la parole ou l'image. Il est particulièrement fragile et doit être manipulé avec beaucoup de soins. Le traitement qu'il subit a pour but d'extraire des informations, de modifier le message qu'il transporte ou de l'adapter aux moyens de transmission; c'est là qu'interviennent les techniques numériques. En effet, si l'on imagine de substituer au signal un ensemble de nombres qui représentent sa grandeur ou amplitude à des instants convenablement choisis, le traitement, même dans sa forme la plus élaborée, se ramène à une séquence d'opération logiques et arithmétiques sur cet ensemble de nombres, associées à des mises en mémoire.

Un filtre numérique c'est un procédé de calcul permettant de transformer un signal numérique d'entrée x(k) en un signal numérique de sortie y(k) pour obtenir la modification voulue dans le spectre du signal. En général, un filtre numérique est un système numérique qui permet de faire passer le signal utile et d'éliminer la composante fréquentielle du signal bruit. Le filtrage est donc une opération (traitement) que le filtre effectue sur le signal d'entrée (en générale bruité), pour avoir en sortie que le signal utile.

Les filtres numériques peuvent être classifiés selon la réponse impulsionnelle en deux types: les filtres numériques à réponse impulsionnelle finie **R.I.F** (les filtres non récursifs) et les filtres numériques à réponse impulsionnelle infinie **R.I.I** (les filtres récursifs)[1].

1.2 Convolution discrète

La convolution discrète joue un rôle important dans le filtrage numérique. Un système linéaire invariant dans le temps peut être représenté en utilisant une somme de convolution discrète. Cette somme est déterminée par la réponse impulsionnelle $h\,(n)$ du système, qui relie son entrée et sa sortie. Pour trouver la séquence de sortie $y\,(n)$ pour toute séquence d'entrée $x\,(n)$, nous écrivons la convolution discrète comme indiqué dans l'équation comme suit:

$$y(n) = \sum_{i=-\infty}^{\infty} h(i)\, x(n-i) = \sum_{i=-\infty}^{\infty} h_i\, x(n-i) = h(n) * x(n) \qquad \text{(I.1)}$$

En utilisant une notation conventionnelle, nous exprimons la convolution numérique comme :

$$(n) = h(n) * x(n) \qquad \text{(I.2)}$$

Notez que pour un système causal, sa réponse impulsionnelle est:

$$h\,(n) = 0 \; pour \; n<0 \qquad \text{(I.3)}$$

La limite inférieure de la somme de convolution commence à 0 au lieu de moins l'infini, c'est-à-dire [2] :

$$\text{y}\,(n) = \text{y}\,(\text{n}) = \sum_{i=0}^{\infty} h(i)\, x(n-i) = \sum_{i=-\infty}^{\infty} x(i)\, h(n-i) \qquad \text{(I.4)}$$

1.3 Corrélation

Pour des signaux numérique (que l'on supposera réels) définis par {s(n)} et {r(n)} et tels que dim (s) =N et dim (s) = M, on définit la fonction de corrélation par:

$$C_{sr}\,(n) = \sum_{m}^{N-1} s(m)\, r(m-n) \;\; Avec\; n \in [0, \text{M} + \text{N} - 1] \qquad \text{(I.5)}$$

$$C_{sr}\,(n) = \sum_{m=0}^{N-1} s(m+n)\, r(m) \qquad \text{(I.6)}$$

Le problème du calcul de la fonction d'inter corrélation des deux signaux s et r est équivalent a celui rencontré dans l'opération de convolution [3].

1.4 Transformation en Z

La transformation en Z est un outil très important pour décrire et analyser les systèmes de données numériques. Il propose également les techniques de conception de filtre numérique et d'analyse de fréquence des signaux numériques.

1.4.1 Définition

Nous commençons par la définition de la transformation en Z. La transformation en Z d'une séquence causal $x\,(n)$, désignée par $X\,(z)$ ou $((n))$ est définie comme :

$$X(z) \;=\; Z\big(X(n)\big) \;=\; \sum_{n=0}^{+\infty} x(n)\, Z^{-n} \qquad \text{(I.7)}$$

Où z est la variable complexe. Le signal numérique $x\,(n)$ est une séquence causale, c'est-à-dire $x\,(n)$=0 pour n<0.

Dans ce qui suit, nous explorons la relation entre la transformée z et la transformée de Fourier. Utilisation de la relation :

$$\mathrm{Z} \;=\; re^{2j\pi f} \qquad \text{(I.8)}$$

Notez que lorsque r = 1, la transformée z devient la transformée de Fourier d'un signal échantillonné donné par :

$$X\,(z = e^{2j\pi f}\,) \;=\; Z\big(X(n)\big) \;=\; \sum_{n=0}^{+\infty} x(n)\, e^{-2j\pi f n} \qquad \text{(I.9)}$$

Par conséquent, la transformée en z est une généralisation de la transformée de Fourier d'un signal échantillonné [4].

1.5 L'analyse de Fourier

1.5.1 Fonction périodique

Une fonction f : R → R est dite périodique de période T si f(x+T)=f(x) pour tout x. On aussi f(x+nT)=f(x) pour tout entier n, donc tout multiple de T est aussi une période. En pratique on met en évidence la plus petite période. Exemple : $\cos(\omega x)$, $\sin(\omega x)$, la période est T = $2\pi\omega$[5].

1.5.2 Développement en série de Fourier d'une fonction périodique

Soit s(t), une fonction de la variable t périodique et de période T, c'est –à-dire satisfaisant la relation:

$$s(t+T) = s(t) \qquad (I.10)$$

Sous certaines conditions, on démontre que cette fonction est développable en série de Fourier, c'est-à-dire que l'égalité suivante est vérifiée:

$$s(t)=\sum_{n=-\infty}^{+\infty} C_n\, e^{j2\pi nt/T} \qquad (I.11)$$

L'indice *n* est un entier et les $\boldsymbol{C_n}$ sont appelés les coefficients de Fourier; ils sont définis par l'expression:

$$C_n = \frac{1}{T}\int_0^T s(t)\, e^{-j2\pi nt/T}\, dt \qquad (I.12)$$

En fait les coefficients de Fourier minimisent l'écart quadratique entre la fonction s(t) et le développement (I.11).En effet la valeur (I.12) est obtenue en dérivant par rapport au coefficient d'indice n l'expression:

$$\int_0^T (s(t) - \sum_{m=-\infty}^{\infty} C_m\, e^{j2\pi mt/T})^2\, dt \qquad (I.13)$$

Et en annulant cette dérivée.

1.5.3 Transformation de Fourier d'une fonction

Soit s(t), une fonction de la variable t; sous certaines conditions on démontre l'égalité suivante:

$$s(t) = \int_{-\infty}^{\infty} S(f)\, e^{j2\pi ft}\, df \qquad (I.14)$$

Avec

$$S(f) = \int_{-\infty}^{\infty} s(f)\, e^{-j2\pi ft}\, dt \qquad (I.15)$$

La fonction S*(f)* est la transformée de Fourier de s*(t)*. Plus communément S*(f)* est appelé spectre du signal *s(t)*[6].

1.5.4 Transformée de Fourier discrète (T.F.D)

La transformée de Fourier d'un signal échantillonné ne peut être traitée directement sur un calculateur .Il faut pour pouvoir le faire, réalise une opération identique à celle effectuée sur la représentation temporelle, l'échantillonnage qui conduit à définir la transformée de Fourier discrète (T.F.D). C'est cette approche que l'on va utiliser pour l'introduire. On définira ensuite la T.F.D. et on précisera ses propriétés en remarquant que la T.F.D. d'un signal s'obtient par un produit matriciel. On indiquera enfin le principe des algorithmes rapides du calcul de la T.F.D.

1) Définition de la TFD

On appelle transformée de Fourier discrète d'une séquence x(n), où n=0,..., N-1, la séquence X(k), k=0,..., N-1 définie par:

$$X(k) = \sum_{n=0}^{N-1} x(n)\, W_N^{nk} \tag{I.16}$$

Ayant la séquence X(k), k=0,..., N-1, il peut être utile de retrouver la séquence x(n).On peut alors vérifier que:

$$\forall\, n \in [0, N-1], x(n) = X(k) = \frac{1}{N} \sum_{n=0}^{N-1} X(k)\, W_N^{-nk} \tag{I.17}$$

- **Forme matricielle**

La relation (I.16) peut s'exprimer sous forme matricielle [7].

$$\mathrm{X} = W_N\, x \tag{I.18}$$

2) Inversion de la TFD

$$x(n) = \frac{1}{N} \sum_{k=0}^{N-1} X(k)\, e^{-j2\pi\frac{nk}{N}} \tag{I.19}$$

En effet, calculons:[8]

$$\mathrm{A} = \frac{1}{N} \sum_{k=0}^{N-1} X(k)\, e^{j2\pi\frac{nk}{N}} = \frac{1}{N} \sum_{k=0}^{N-1} \left(\sum_{t=0}^{N-1} x(i)\, e^{-j2\pi\frac{ik}{N}} \right) e^{j2\pi\frac{nk}{N}} \tag{I.20}$$

$$A = \frac{1}{N} \sum_{k=0}^{N-1} x(i) \left(\sum_{k=0}^{N-1} e^{j2\pi\frac{(n-i)k}{N}} \right) \quad \text{(I.21)}$$

$$\text{Si} \quad i \neq n \left(\sum_{k=0}^{N-1} e^{j2\pi\frac{(n-i)k}{N}} \right) = \frac{1-e^{j2\pi(n-i)}}{1-e^{j2\pi\frac{(n-1)}{N}}} \quad \text{(I.22)}$$

$$\text{Si} \quad i = n \left(\sum_{k=0}^{N-1} e^{j2\pi\frac{(n-i)k}{N}} \right) = \sum_{k=0}^{N-1} 1 = N \quad \text{(I.23)}$$

$$A = \frac{1}{N} \sum_{k=0}^{N-1} x(i) \left(\sum_{k=0}^{N-1} e^{j2\pi\frac{(n-i)k}{N}} \right) = \frac{1}{N} x(n)N \quad \text{(I.24)}$$

$$A = x(n) \qquad \text{c.q.f.d} \quad \text{(I.25)}$$

1.5.5 Transformation de Fourier rapide

Le temps pris pour évaluer une TFD sur un ordinateur numérique dépend principalement du nombre de multiplications impliquées, car ce sont les opérations les plus lentes. Avec La TFD, ce nombre est directement lié à $N2$ (multiplication matricielle d'un vecteur), Où N est la longueur de la transformation de fourrier discrète. Pour la plus pars des problèmes, N est choisi pour être au moins égale à 256 pour obtenir une approximation raisonnable du spectre de la séquence considérée du signal. Des algorithmes informatiques hautement efficaces pour l'estimation des transformées de Fourier discrètes ont été développés depuis le milieu des années 60. On les appelle Les transformés de fourrier rapides. Ils s'appuient sur le fait que le DFT standard implique beaucoup de calculs redondants. L'équation $\boldsymbol{s(t)} = \sum_{\boldsymbol{n=0}}^{\infty} \boldsymbol{s(n)}\, \mathbf{sinc}\left(\frac{\boldsymbol{t-nT_e}}{\boldsymbol{T_e}}\right)$ peut s'écrire :

$$S(k) = \sum_{n=0}^{N-1} s(n)\, W_N^{nk} \quad \text{(I.26)}$$

$$\text{Avec} \quad W_N = e^{-\frac{j2\pi}{N}} \quad \text{(I.27)}$$

Il est facile de réaliser que les mêmes valeurs de W_N^{nk} sont calculées plusieurs fois à mesure que le calcul se déroule. Tout d'abord, le produit entier nk se répète pour différentes combinaisons de n et k; Deuxièmement, W_N^{nk} est une fonction périodique avec seulement N valeurs distinctes [9].

1.6 Analyse des filtres numériques

1. Forme générale des filtres numériques

Un filtre numérique peut être défini par une équation aux différences:

$$(\mathrm{n}) = \sum_{i=0}^{N} b_i\, x(n-i) - \sum_{j=1}^{M} b_j\, y(n-j) \qquad \text{(I.28)}$$

Comme l'ordre en *x* et *y* est 1, ce filtre est linéaire. Les termes $\Sigma(n-j)Mj=1$ représentent la partie récursive du système, la forme générale d'une fonction de transfert est :

$$\mathrm{H(z)} = \frac{Y(z)}{X(z)} = \frac{\sum_{i=0}^{N} b_i z^{-i}}{1+\sum_{j=1}^{M} a_j z^{-j}} = b_i\, \frac{\prod_{i=0}^{N}(1-Z_j Z^{-1})}{\prod_{j=0}^{M}(1-p_j Z^{-1})} \qquad \text{(I.29)}$$

Fonction de transfert $H\,(z)$ à M pôles pj et N zéros Z_i réels ou en paires complexes conjuguées [10].

2. Fonction de transfert

La réponse d'un système de réponse impulsionnelle $h(n)$, à une entrée arbitraire $x(n)$, est donnée par la relation de convolution suivante :

$$y\,(n) = h\,(n) * x\,(n) \qquad \text{(I.30)}$$

La transformée en z de (n) est donnée par:

$$Z(\mathrm{y(n)}) = Y(z) = X(z)H(z) \qquad \text{(I.31)}$$

Alors la fonction de transfert est exprimé par :

$$H(z) = \frac{Y(z)}{X(z)} \qquad \text{(I.32)}$$

La fonction de transfert est définie uniquement pour les systèmes linéaire invariant, comme le rapport de la sortie (z) et de l'entrée (z), Elle est définit aussi comme la transformée en z de la réponse impulsionnelle du système $h(n)$. Un système linéaire invariant est également décrit par l'équation de différence suivante :

$$y(n) = -\sum_{i=1}^{N} a_i\, y(n-i) + \sum_{j=0}^{M} b_j\, x(n-j) \qquad \text{(I.33)}$$

La transformée en z de (n) est donnée par :

$$Y(z) = y(n) = -\sum_{i=1}^{N} a_i z^{-i}\, Y(z) + \sum_{j=0}^{M} b_j\ z^{-j}\, X(Z) \qquad \text{(I.34)}$$

L'équation (I.33) peut être réarrangé et exprimé en termes de rapport de deux polynômes, comme suit :

$$\frac{Y(z)}{X(z)} = H(z) = \frac{\sum_{j=0}^{M} b_j\, z^{-j}}{1+\sum_{i=1}^{N} a_i z^{-i}} \qquad \text{(I.35)}$$

$H(z)$ est connu comme la fonction de transfert du système.[11]

1.7 Classification des filtres

On classe les filtres en deux grandes familles et suivant la durée de leur réponse impulsionnelle.

1.7.1 Filtre à réponse impulsionnelle finie (R.I.F)

Un filtre à réponse impulsionnelle finie RIF est complètement spécifié par la relation entrée-sortie suivante:[12]

$$y(n) = \sum_{i=0}^{N} b_i\, x(n-i) \qquad \text{(I.36)}$$

$$y(n) = b_0\, x(n) + b_1\, x(n-1) + \cdots + b_N\, x(n-N) \qquad \text{(I.37)}$$

Où b_i représente les coefficients du filtre RIF et N indique la longueur du filtre FIR. L'application de la transformée en z des deux côtés de l'équation (I.36) conduit à :

$$Y(z) = \sum_{i=0}^{N} b_i\ z^{-i}\, X(Z) \qquad \text{(I.38)}$$

La fonction de transfert du filtre s'écrit :

$$\frac{Y(z)}{X(z)} = \sum_{i=0}^{N} b_i\ z^{-i} \qquad \text{(I.39)}$$

1. Principales propriétés des filtres RIF

➢ Stabilité

Un système sera stable si tous les pôles de sa fonction de transfert présentent un module inférieur à 1, c'est à dire tous les pôles sont à l'intérieur du cercle unité dans le plan complexe.

✓ Les filtres à réponse impulsionnelle finie RIF sont toujours stables car ils n'admettent pas de pôles.[13]

➢ Approximation

Toute fonction de filtrage numérique stable et causale peut être approchée par la fonction de transfert d'un filtre RIF.[14]

➢ Filtre à phase linéaire

La réponse fréquentielle d'un filtre RIF est la suivante :

$$H(e^{j\omega}) = \sum_{n=0}^{M} h(n)e^{-j\omega n} \tag{I.40}$$

Si la condition de symétrie est satisfaite : h(n)=h(M-n), la somme ci-dessus peut s'écrire sous la forme :

$$(He^{j\omega}) \begin{cases} e^{-\frac{j\omega M}{2}}\left[h\left(\frac{M}{2}\right) + \sum_{n=0}^{\frac{M}{2}-1} 2h(n)\cos\left(\omega\left(n-\frac{M}{2}\right)\right)\right] & (pour\ M\ pair) \\ e^{-\frac{j\omega M}{2}}\left[\sum_{n=0}^{\frac{M-1}{2}} 2h(n)\cos\left(\omega\left(n-\frac{M}{2}\right)\right)\right] & (pour\ M\ impaire) \end{cases} \tag{I.41}$$

Dans cette équation le terme entre les crochets est réel. La phase est donc $\frac{\omega M}{2}$, ce qui veut dire que la phase est linéaire et correspond à un délai de $\frac{M}{2}$ échantillons:

$$\tau = \frac{d}{d\omega}(\varphi\omega) = \frac{M}{2} \tag{I.42}$$

Dans beaucoup de cas pratiques, les filtres à phase linéaire sont exigés. De plus, cette propriété permet une implantation plus facile des filtres RIF. C'est la raison pour laquelle nous ne considérons que les filtres à phase linéaire [15]

- La durée du régime transitoire est limitée à la durée de la réponse impulsionnelle.
- Pas de propagation des erreurs de calcul (programmation non récursive).
- **Faiblesse :** pour améliorer les performances du filtre on peut être amené à augmenter le nombre d'échantillons.[16]

2. Conception des filtres R.I.F

✓ Trois principales méthodes de synthèse de filtres RIF :

2.1 Méthode des fenêtres

L'utilisation des fenêtres est la méthode la plus directe de la conception des filtres RIF. Imaginer que nous avons à notre disposition la réponse fréquentielle recherchée$H_d(e^{j\omega})$. En général, pour des filtres de fréquence, c'est une constante, 1, pour une plage fréquentielle et zéro pour le reste des fréquences. La réponse temporelle d'un tel filtre peut être calculée en faisant une transformé de Fourier inverse sur cette réponse fréquentielle :

$$h_d(n) = \frac{1}{2\pi}\int_{-\pi}^{\pi} H_d(e^{j\omega})e^{j\omega n}d\omega \qquad \text{(I.43)}$$

Si la durée de $h_d(n)$ est infinie, une méthode d'obtenir un filtre RIF est de tronquer cette réponse sur une fenêtre de largeur désirée :

$$h(n) = \begin{cases} h_d(n), & 0 \le n \le M \\ 0 \quad , & ailleur \end{cases} \qquad \text{(I.44)}$$

Cette troncation est équivalente à une multiplication d'une fonction porte par le $h_d(n)$: $h(n) = h_d(n)\, w(n)$ où:

$$H(e^{j\omega}) = \frac{1}{2\pi}\int_{-\pi}^{\pi} H_d\left(e^{j\theta}\right) W\left(e^{j(\omega-\theta)}\right)d\theta \qquad \text{(I.45)}$$

La multiplication dans le temps correspond à une convolution dans le domaine de fréquence :

$$\text{Où } W(e^{j\omega n}) = \sum_{n=0}^{M} e^{-j\omega n} = \frac{1-e^{-j\omega(M+1)}}{1-e^{-j\omega}} = e^{-j\omega M/2} \qquad \text{(I.46)}$$

- Le schéma ci-dessous présente graphiquement cette relation.[17]

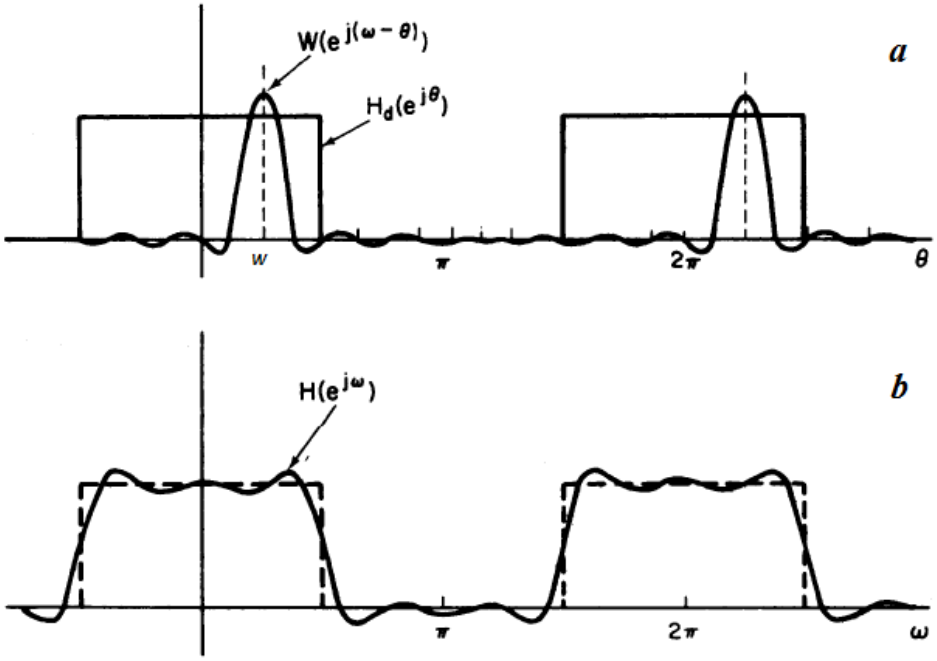

Figure 1.1: Schéma présenté la multiplication dans le temps à une convolution dans le domaine de fréquence.

Ceci signifie que la réponse fréquentielle résultant est la réponse "arrondie" du filtre désiré. C'est à dire que les discontinuités du filtre idéal sont remplacées par des ondulations autour de ces sauts. Pour ne pas trop dégrader la réponse idéale, la fenêtre doit avoir un certain nombre de propriétés. Dans le domaine fréquentiel, il faut qu'il se rapproche à un delta dirac.

Si on augmente la taille de la fenêtre, la largeur du lobe principal se réduit, d'où l'effet d'arrondissement va réduire. Plus la variation de $\boldsymbol{H_d(e^{j\omega})}$ est important, plus nous avons besoins d'un $\boldsymbol{w(e^{j\omega})}$ pointu ce qui se traduit par une fenêtre plus longue. De l'autre côté, il est préférable d'avoir un **h(n)** le plus court possible d'où une petite fenêtre (pour des raisons de complexité d'implantation). Il y a donc un compromis entre ces deux propriétés contradictoires.

La figure ci-dessous présente la réponse fréquentielle de w(n) pour différentes valeurs de M. Il se voit qu'en augmentant le M, la largeur (fréquentielle) du lobe principal diminue.

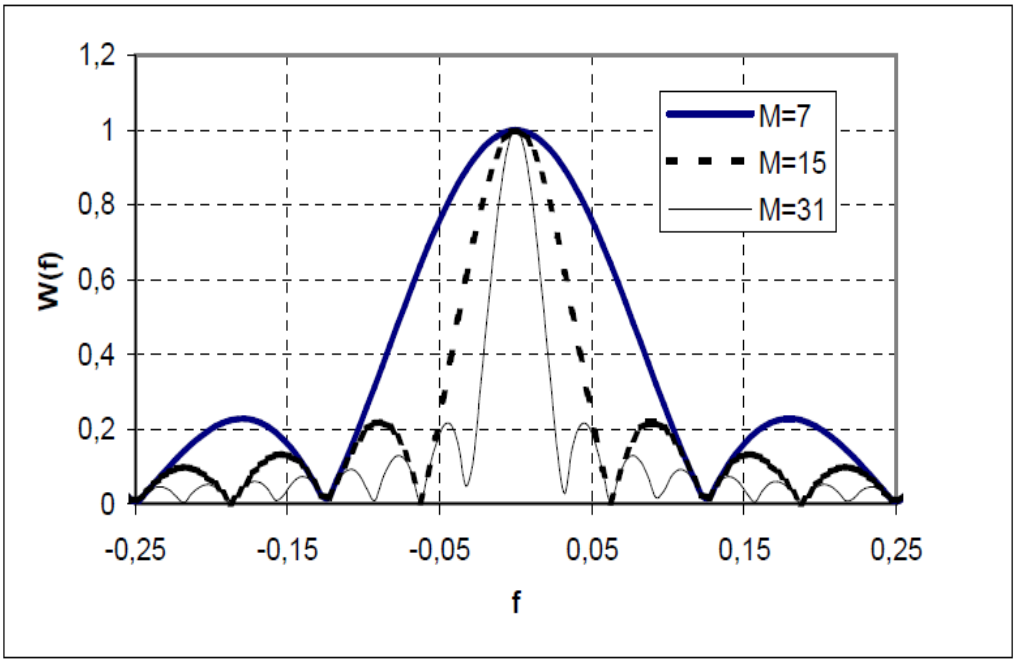

Figure 1.2: la réponse fréquentielle de w(n) pour différentes valeurs de M.

Il existe autres méthodes de fenêtrage qui, tout en gardant la largeur temporelle de la fenêtre constante, présente une réponse fréquentielle plus pointue (lobe principal plus mince ou évanouissement plus important des lobes secondaires). Pour M constant, un lobe principal plus étroit conduit à un évanouissement moins important.

Un fenêtrage rectangulaire est optimisé dans le sens de la minimisation de l'expression :

$$\mathrm{E} = \frac{1}{2\pi}\int_{-\pi}^{\pi}\left|H_d(e^{j\omega}) - H(e^{j\omega})\right|^2 d\omega \qquad \text{(I.47)}$$

La méthode Bartlett consiste à utiliser une fenêtre triangulaire à la place d'une fenêtre rectangulaire. C'est à dire qu'il pondère la réponse impulsionnelle du filtre idéal tout en insistant sur les points du milieu. Les méthodes Hanning, Hamming, Blackman et Kaiser proposent d'autres méthodes de fenêtrage.[17]

Bartelett :

$$w(n) = \begin{cases} \frac{2n}{M}, & 0 \le n \le \frac{M}{2} \\ 2 - \frac{2n}{M}, & \frac{M}{2} \le n \le M \end{cases} \qquad \text{(I.48)}$$

Hanning:

$$w(n) = \frac{1}{2}\left[1 - \cos\left(\frac{2\pi n}{M}\right)\right] 0 \leq n \leq M \qquad \text{(I.49)}$$

Hamming:

$$w(n) = 0.54 - 0.46\cos\left(\frac{2\pi n}{M}\right) \qquad 0 \leq n \leq M \qquad \text{(I.50)}$$

Blackman:

$$w(n) = 0.42 - 0.5\cos\left(\frac{2\pi n}{M}\right) + 0.08\cos\left(\frac{4\pi n}{M}\right) \qquad \text{(I.51)}$$

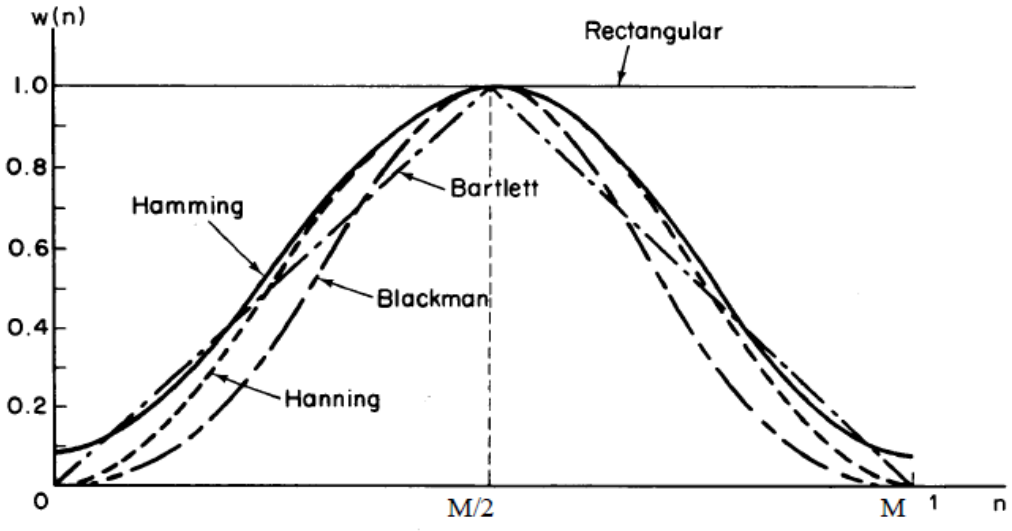

Figure 1.3: Schéma présenté quelque méthode de fenêtres.

Fenêtre	Amplitude du lobe secondaire	Largeur du lobe principal	Atténuation minimale de la bande coupée
Rectangulaire	-13	$4\pi(M+1)$	-25
Bartelett	-25	$8\,\pi/M$	-25
Hanning	-31	$8\,\pi/M$	-44
Hamming	-41	$8\,\pi/M$	-53
Blackman	-57	$12\,\pi/M$	-74

Tableau 1.1: Quelques propriétés des fenêtres [16].

2.2 Méthode de l'échantillonnage en fréquence

- La réponse fréquentielle est échantillonnée, la réponse impulsionnelle est obtenue par TF inverse.
- Possibilité de définir des gabarits « personnalisés ».

2.3 Méthode de synthèse de filtre optimal (REMEZ)

- La réponse impulsionnelle est synthétisée avec des méthodes. d'optimisation ayant comme critère la minimisation des oscillations et la raideur de la pente de coupure.[18]

1.7.2 Filtre à réponse impulsionnelle infinie (R.I.I)

Dans cette partie, nous étudierons plusieurs méthodes de conception de filtres à réponse impulsionnelle infinie (RII). Un filtre RII est décrit en utilisant l'équation de différence :

$$y(n) = a_0x(n) + a_1x(n-1) + \cdots + a_Mx(n-M) - b_1y(n-1) - \cdots - b_N(n-N) \quad \text{(I.52)}$$

Le nombre total de coefficients est N + M + 1, ce qui signifie que ces multiples multiplications sont nécessaires pour calculer chaque nouvelle sortie de l'équation de différence:

$$y(n) = \sum_{i=0}^{M} a_i\, x(n-i) - \sum_{j=1}^{N} b_j\, y(n-j) \quad \text{(I.53)}$$

La fonction de transfert de filtre RII est donnée par [19] :

$$H(z) = \frac{\sum_{i=0}^{M} a_i z^{-i}}{1+\sum_{j=1}^{N} b_j z^{-j}} \quad \text{(I.54)}$$

1. Spécificités des filtres RII

- Peuvent être obtenus par transposition d'un **filtre continu.**
- Peuvent être obtenus avec un **petit nombre** de coefficients.
- Mise en œuvre **récursive.**
- Peuvent être **instables.**
- La réponse fréquentielle peut présenter une phase **non linéaire.**
- Une bonne précision de calcul est nécessaire pour éviter la propagation des erreurs [20].

2. Conception des filtres R.I.I

✓ Principales méthodes de synthèse de filtre RII**:**

2.1 Transformation bilinéaire

Pour une période d'échantillonnage Te, nous effectuons maintenant la transformation bilinéaire avec :

$$S = \frac{1}{T_e}\frac{1-z^{-1}}{1+z^{-1}} \quad \text{(I.55)}$$

Cette transformation peut être motivée par la formule trapézoïdale pour l'intégration numérique. Définir la fonction du système à temps discret :

$$H(z) = H_a(s)\big|\ \ S = \frac{1}{T_e}\frac{1-z^{-1}}{1+z^{-1}} \quad \text{(I.56)}$$

Cette transformation produit une fonction de système rationnelle, c'est-à-dire un rapport de polynômes en z. Ce (z) est une fonction système dont la réponse en fréquence est liée à la réponse en fréquence du filtre analogique RII [21].

2.2 Transposition du filtre analogique en filtre numérique

2.3 Méthode de l'invariance impulsionnelle

2.4 Equivalence à la dérivation **[**22]

1.8 Conversion analogique – numérique(CAN) (Analog to Digital Converter (ADC))

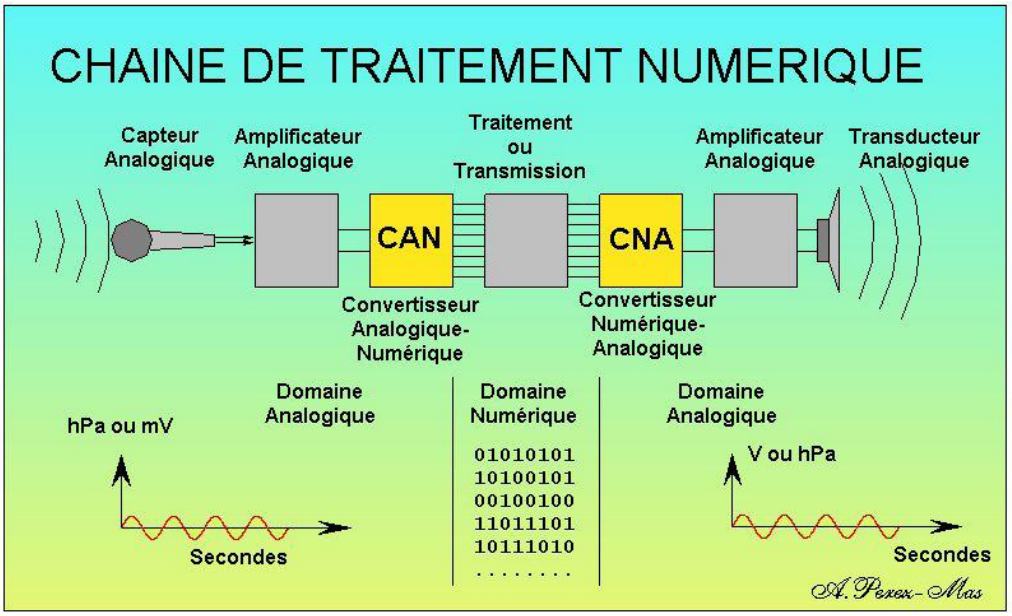

Figure1.4: Chaine de traitement numérique [23].

1.8.1 Principe de la conversion analogique - numérique

✓ **Définition :** Un convertisseur analogique – numérique (CAN) est un dispositif électronique permettant la conversion d'un signal analogique en un signal numérique.

Cette première définition pour être complète en appelle deux autres, celles des signaux analogiques et numériques :

Signal analogique : signal continu en temps et en amplitude.

Signal numérique : signal échantillonné et quantifié, discret en temps et en amplitude.

Conceptuellement, la conversion analogique – numérique peut être divisée en trois étapes :

L'échantillonnage temporel, la **quantification** et le **codage**.

La figure I.5 présente successivement ces trois étapes pour un CAN dont la sortie du signal numérique est sur 3 bits :

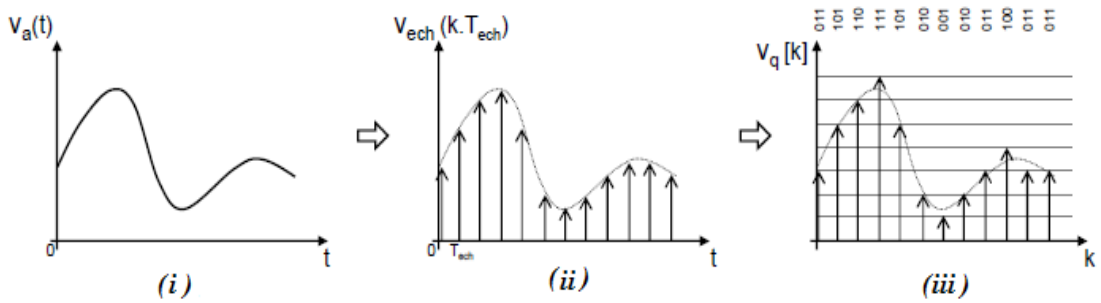

Figure1.5: – (i) signal analogique (ii) signal échantillonné (iii) puis quantifié.

Un signal analogique, $V_a(t)$ continu en temps et en amplitude (i) est échantillonné à une **période d'échantillonnage** constante T_{ech}. On obtient alors un signal échantillonné

V_{ech}(k. T_{ech}) discret en temps et continu en amplitude (ii). Ce dernier est ensuite quantifié, on obtient alors un signal numérique V_q[k] discret en temps et en amplitude (iii).

La quantification est liée à la **résolution** du CAN (son nombre de bits) ; dans l'exemple précédent V_q [k] peut prendre huit amplitudes différentes (soit 23, 3 étant le nombre de bits du CAN). La figure 1.3.iii présente également le code numérique sur trois bits (en code binaire naturel) associé à V_q[k] en fonction du temps.

Les notions précédentes seront approfondies dans les parties suivantes.

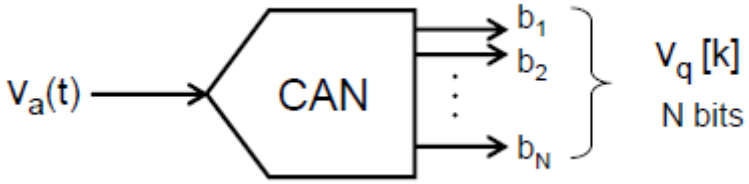

Figure 1.6: Schéma présente le symbole d'un CAN à N bits [24].

1.8.2 Fonction de transfert

$N = \frac{U_E - U_{Emin}}{q}$ avec le quantum (précision $q = \frac{\Delta U_{Pleine\ échell}}{2^n - 1} = \frac{U_{Emax} - U_{Emin}}{2^n - 1}$

La résolution numérique d'un convertisseur correspond a son de bits n.

Remarque : $\frac{1}{q}$ représente le pente.

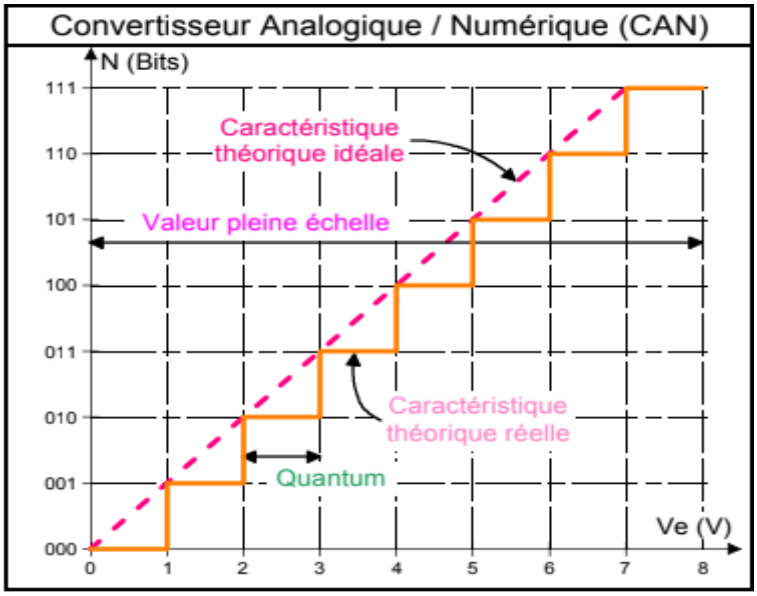

Figure 1.7: courbe représentant la grandeur de sortie en fonction de grandeur d'entrée d'un convertisseur Analogique / Numérique [25].

1.9. Conclusion

Dans ce chapitre nous avons présenté quelques généralités sur le traitement numériques du signal et on a vu les différentes méthodes de conception des deux types de filtres R.I.F et R.I.I.

Pour le prochain chapitre, nous allons présenter les différentes caractéristiques de la carte DSP TMS320 C6713 DSK.

Chapitre II :

La carte DSP TMS320 C6713 DSK

Sommaire

2.1. Introduction

Le filtrage est l'une des opérations de traitement des signaux les plus utilisées. Les DSP sont maintenant disponible pour implémenter des filtres numériques en temps réel. Le jeu d'instructions TMS320C6x et l'architecture le rend bien adapté pour de telles opérations de filtrage.

Un filtre analogique fonctionne sur des signaux continus et généralement réalisé avec des composants discrets comme les amplificateurs opérationnels, les résistances et les condensateurs. Toutefois, un filtre numérique, comme en tant que filtre FIR, fonctionne sur des signaux - temps discrets et peut être mis en œuvre avec un DSP comme le TMS320C6x.

Cela implique l'utilisation d'un ADC pour capturer le signal d'entrée, le traitement des échantillons d'entrée et l'envoi de la sortie d'un CAD. Au cours des dernières années, le coût des DSP a été considérablement réduit, qui ajoute aux nombreux avantages que les filtres numériques ont sur leur analogique .

Le traitement numérique du signal est au «cœur» de la plupart des technologies que nous utilisons aujourd'hui. Le cœur réel d'une structure de traitement numérique du signal est le processeur DSP (Digital Signal Processor) [26].

2.2. Définition DSP

- DSP (digital signal processor) est un processeur spécialement conçu pour le traitement numérique du signal en temps réel.
- Effectuer en grandes vitesse les opérations divers de traitement de signal tel que le filtrage, la transformation et l'analyse spectrale en temps réel.
- L'architecture en DSP est optimale pour effectuer des calculs complexes en un cycle d'horloge, mais aussi pour accéder très facilement à un grand nombre d'entrées-sorties (numériques ou analogiques).
- Les DSP sont utilisés dans la plupart des applications en temps réel. On les trouve dans les modems, les téléphones mobiles, les appareils multimédia, les récepteurs GPS, etc... .
- Le DSP représente le cœur de système de traitement numérique du signal [27].

2.3. Classification de DSP

On peut classer les DSP soulent deux critère :

2.3.1. DSP à virgule flottante

Les données sont représentées en utilisant une mantisse et un exposant. La représentation de ces nombres s'effectue selon la formule suivante:
n= mantisse x 2 exposant. Généralement, la mantisse est un nombre fractionnaire (-1 à +1), et l'exposant est un entier indiquant la place de la virgule en base 2 (c'est le même mécanise qu'en base 10). La très grande dynamique proposée par les DSP à virgule flottante (comme TMS320C6713) permet virtuellement de ne pas se soucier des limites des résultats calculés lors de la conception d'un programme. Cet avantage a cependant un prix, à savoir qu'un système basé sur un DSP à virgule flottante a un coût de fabrication supérieur par rapport à un système basé sur DSP à virgule fixe. La puce d'un DSP à virgule flottant nécessite à la fois une surface de silicium plus important (cœur plus complexe), et un nombre de broches supérieur, car la mémoire externe est elle aussi au format 32 bits. Le système revient donc plus cher (exemple : 2 x 32 broches juste pour les bus de données externes avec une architecture Harvard de base).[28]

2.3.1.1. Codage en virgule flottante

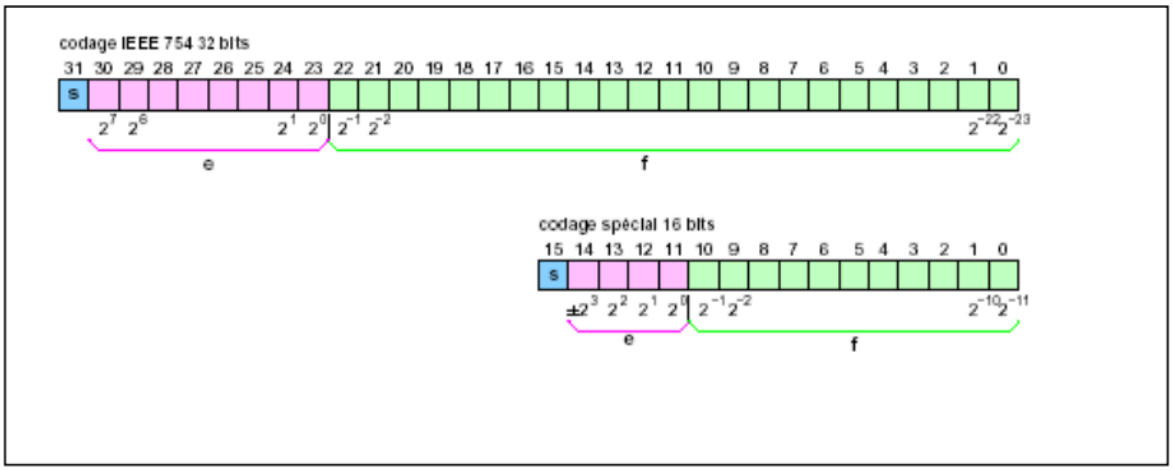

Figure 2.1 : codage à virgule flottante

La mantisse est composée du 1 implicite et de la fraction f. Suivant les DSP, il y a des codages différents de l'exposant (avec ou sans biais) et des valeurs particulières.

2.3.2 Les DSP à virgule fixe

Un DSP à virgules fixes est un peu plus compliqué à programmer qu'un DSP à virgules flottantes. Le programmeur doit être très attentif lors de la programmation car c'est à lui de connaître à tout instant l'état du système. Dans un DSP à virgules fixes, les nombres sont codés sur 16 bits.

Toutefois, sur ce DSP, les calculs sont effectués avec des accumulateurs de 32 bits. Lorsque les résultats doivent être stockés en mémoire, les 16 bits les moins significatifs sont perdus. Ceci permet de limiter les erreurs d'arrondis cumulatives.

La précision des calculs est un point critique des DSP à virgules fixes, car le concepteur de programmes doit rester vigilant à chaque étape d'un calcul. Il doit rechercher la plus grande dynamique possible (c'est à dire exploiter au mieux la gamme des nombres disponibles), pour conserver une bonne précision des calculs Les programmeurs contournent les limites des DSP à virgu le fixe en déterminant à l'avance la précision et la dynamique nécessaire pour réaliser leurs projets.

Il est également possible d'effectuer des opérations en virgule flottante dans un DSP à virgules fixes par le biais de routines logicielles adéquates. Cette approche est néanmoins pénalisante en temps d'exécution, même sur un DSP à virgules fixes très rapide. En termes de rapidité, les DSP à virgules fixes se placent d'ordinaire devant leurs homologues à virgules flottantes, ce qui constitue un critère de choix important. Les DSP à virgules fixes sont les plus utilisés, car ils sont moins chers que les DSP à virgules flottantes. On les trouve dans tous les produits de grande diffusion où le coût est un facteur important.

2.3.2.1 Codage en virgule fixe

Le principe de ce codage est représenté par la figure ci-dessous :

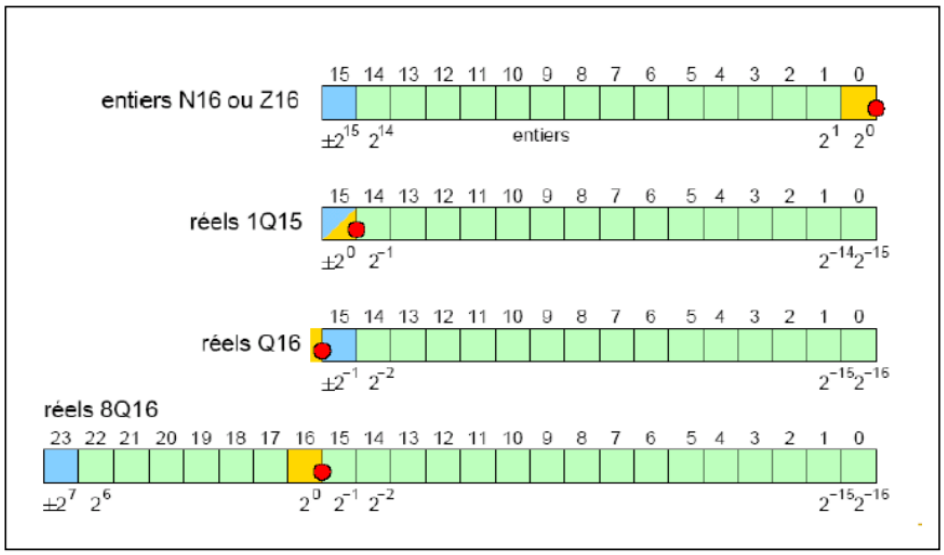

Figure 2.2 : Codage en virgule fixe [29]

2.4. Type d'architecture interne des processeurs

Elle définit la manière dont les éléments d'un système à microprocesseur sont interconnectés et échangent leurs informations. On distingue :

2.4.1. Architecture de Von Neumann

Dans l'architecture de la machine de Von Neumann, les programmes et les données sont enregistrés sur la même mémoire. Chaque instruction contient la commande de l'opération a effectuer et l'adresse de la donnée a utilisé, il faut donc souvent plusieurs cycles d'horloge pour exécuter une instruction. La caractéristique de cette architecture est qu'elle ne possède qu'un système de bus et une mémoire globale. Un microprocesseur basé sur cette architecture stocke les programmes et les données dans la même zone.

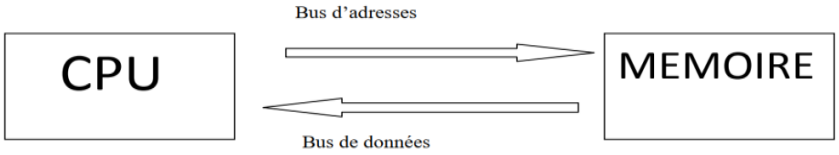

Figure 2.3 : L'architecture Von Neumann

2.4.2. Architecture Harvard

Avec cette architecture, les mémoires sont séparées l'exécution des instructions est plus rapide. Il n'y a aucun risque de conflit d'adresse.

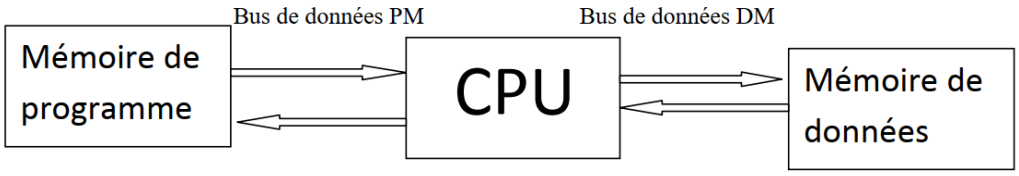

Figure 2.4: L'architecture Harvard.[30]

2.5. Avantages des systèmes à base de DSP

Tous les systèmes à base de DSP bénéficient des avantages suivants :

1. Souplesse de la programmation

Un DSP est avant tout un processeur exécutant un programme de traitement du signal. Ceci signifie que le système bénéficie donc d'une grande souplesse de développement. De plus, les fonctions de traitement numérique peuvent évoluer en fonction des mises à jour des programmes, et cela pendant toute la durée de vie du produit incluant le système. La modification d'un filtre numérique ne nécessite pas un changement matériel.

2. Implémentation d'algorithmes adaptatifs

Une autre qualité issue de la souplesse des programmes. Il est possible d'adapter une fonction de traitement numérique en temps réel suivant certains critères d'évolutions du signal (exemple : les filtres adaptatifs).

Des possibilités propres au système de traitement numérique du signal:

Certaines fonctions de traitement du signal sont difficiles à implanter en analogique, voire irréalisables (exemple : un filtre réponse en phase linéaire) [31].

3. Stabilité

En analogique, les composants sont toujours plus ou moins soumis à des variations de leurs caractéristiques en fonction de la température, de la tension d'alimentation, du vieillissement, etc. Une étude sérieuse doit tenir compte de ces phénomènes, ce qui complique et augmente le temps de développement. Ces inconvénients n'existent pas en numérique.

4. Répétitivité, reproductibilité

Les valeurs des composants analogiques sont définies avec une marge de précision plus ou moins grande. Dans ces conditions, aucun montage analogique n'est strictement reproductible à l'identique, il existe toujours des différences qu'il convient de maintenir dans des limites acceptables. Un programme réalisant un traitement numérique est par contre parfaitement reproductible, « à l'infini » [32].

2.6. La carte DSK TMS320C6713

2.6.1. Définition

La plateforme TMS320C6000 des processeurs de signaux numériques fait partie de la famille TMS320 de TEXAS INSTRUMENTS. Elle comporte les processeurs TMS320C62x à arithmétique fixe et TMS320C67x à arithmétique flottante. Le TMS320C6713 est considéré comme le membre le plus performant de la catégorie C67x, son architecture VLIW lui permet le traitement par paquets de huit instructions en parallèle par huit unités fonctionnelles. La discussion dans ce chapitre se focalisera sur le processeur TMS320C6713. L'architecture et les périphériques qui lui sont associés seront également discutés.

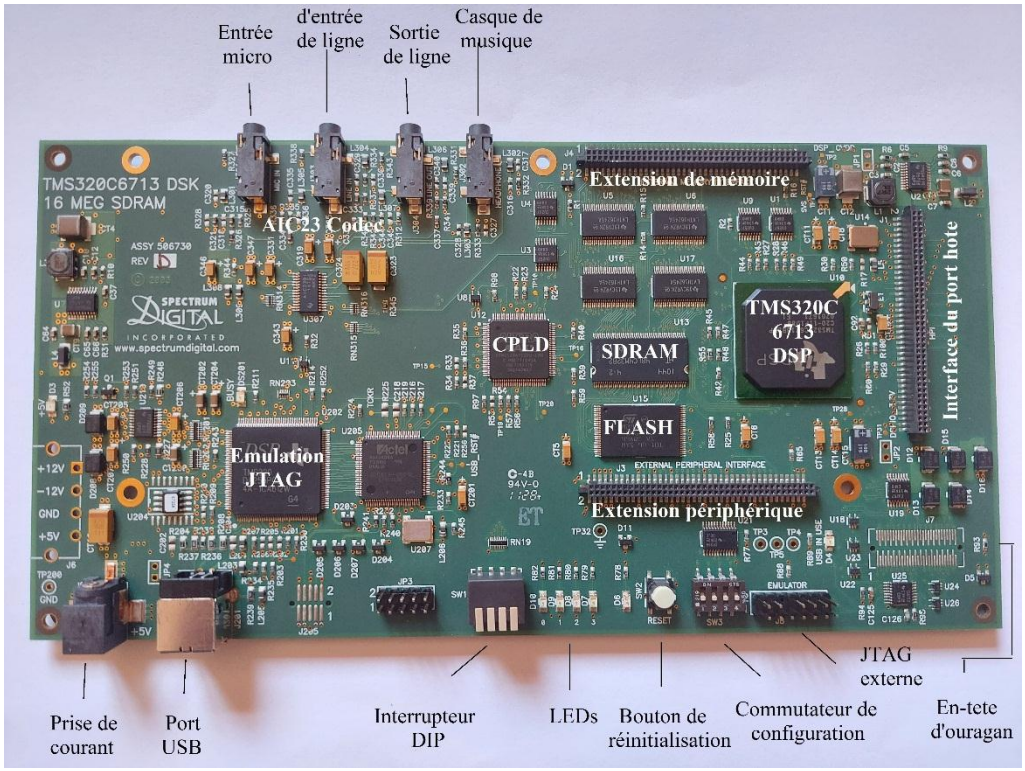

Figure 2.5 : La carte du kit TMS320C6713 DSK

2.6.2. Les principales caractéristiques de la carte DSK

Les principales caractéristiques et fonctionnalités matérielles de la carte DSK sont :

- ✓ Un processeur DSP TMS320C6713 fonctionnant à une fréquence d'horloge de 225 MHz.
- ✓ Un codec stéréo AIC23 avec une ligne d'entrée, une ligne de sortie, un MIC et une line pour un casque stéréo ; avec une fréquence d'échantillonnage de 8 kHz à 96 kHz.
- ✓ Une mémoire dynamique DRAM synchrone (SDRAM) de 16 Mo.
- ✓ Une mémoire Flash non volatile de 512 Ko (256 Ko utilisables dans la configuration par défaut).
- ✓ Quatre LED et des commutateurs DIP accessibles par l'utilisateur.
- ✓ Configuration logicielle de la carte via des registres implémentés dans un dispositif logique complexe de type (CPLD) .
- ✓ Démarrage configurable avec plusieurs options.

✓ Connecteurs d'extension pour cartes mère.

✓ Émulation JTAG via un émulateur JTAG intégré dans la carte mère avec interface hôte [33].

2.6.3. Support matériel

Un schéma de principe simplifié de la carte DSK est illustré par les figures ci-dessous :

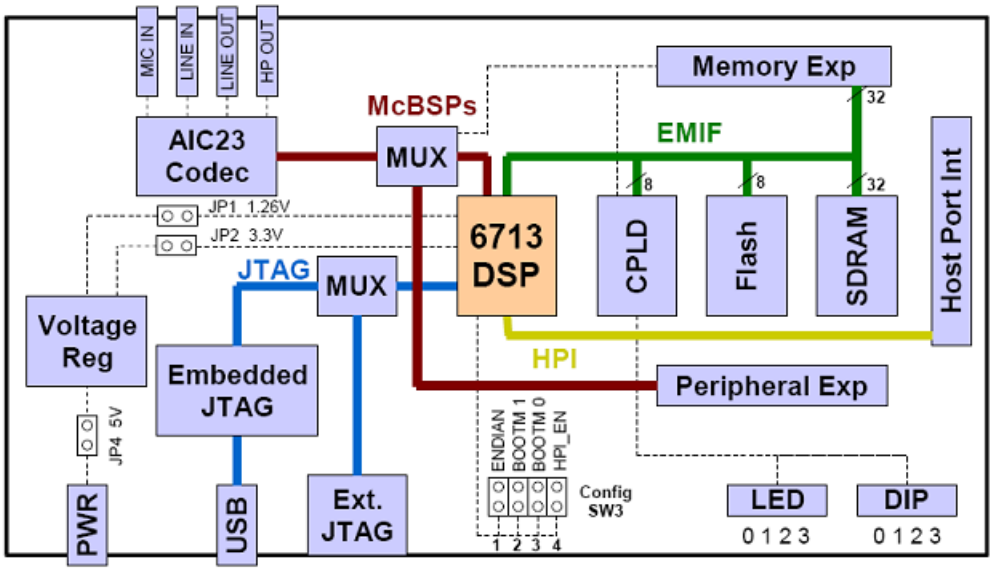

Figure 2.6 : Schéma fonctionnel du kit de démarrage DSP TMS320C6713(DSK)

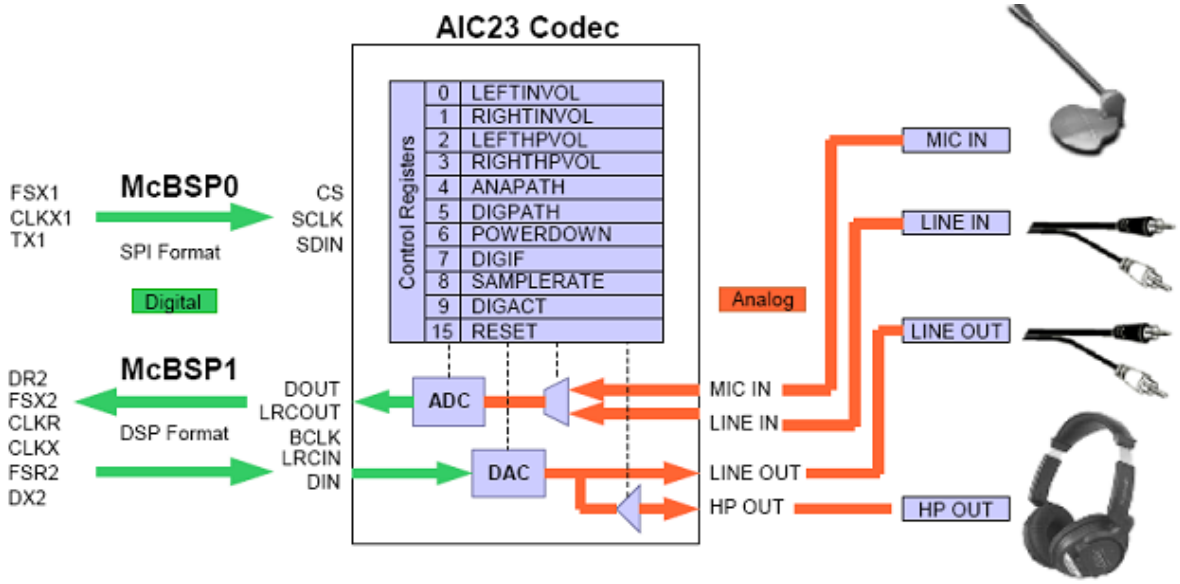

Figure 2.7 : Codec stéréo AIC23 avec prises stéréo Lin In, Lin out, MIC et casque [34]

2.6.4. L'architecture interne du Tms320c6713

La figure ci-dessous illustre l'architecture du DSP :

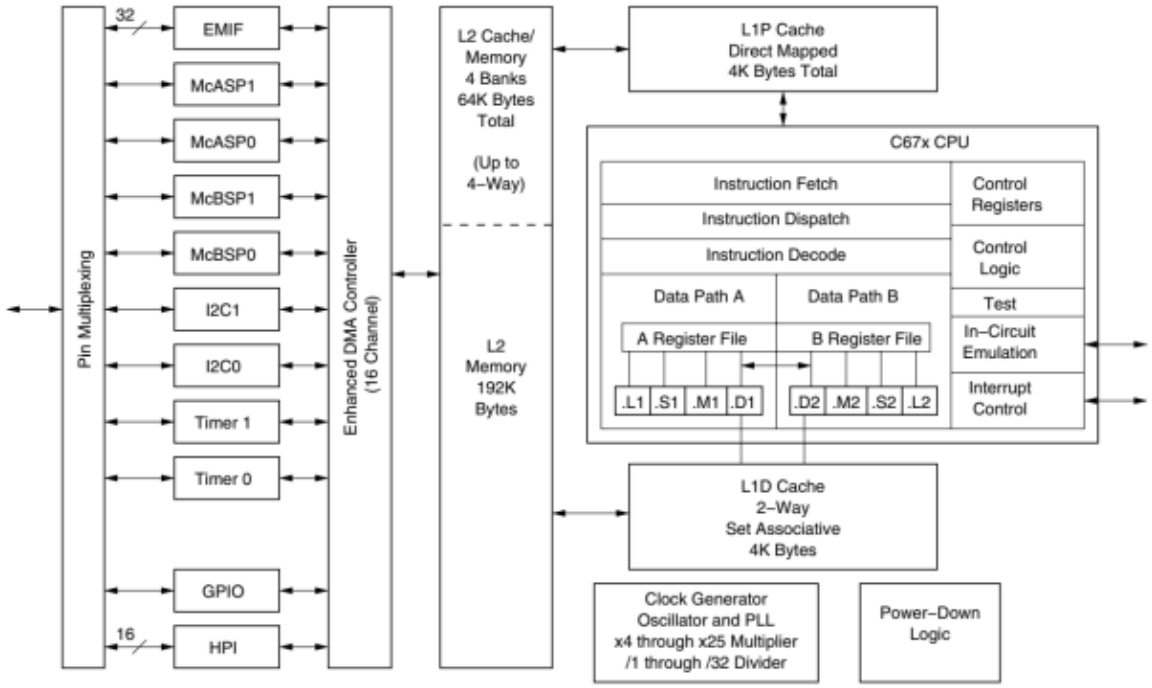

Figure 2.8 : Schéma fonctionnel du DSP TMS320C6713

2.6.5. Les périphériques du TMS320C6713

Le TMS320C6713 a plusieurs périphériques qui sont :

- Le contrôleur DMA : Il permet sans l'aide du CPU de transférer des données entre les espaces mémoire (interne, externe et des périphériques). Il a quatre canaux programmables et un autre canal auxiliaire.
- Le contrôleur EDMA : Il permet le transfert des données entre les espaces mémoire comme le DMA. Il a 16 canaux programmables.
- L'interface port hôte HPI. Il donne au processeur hôte un contrôle total pour un accès direct de l'espace mémoire du CPU et à la cartographie de la mémoire des périphériques du DSP.
- Deux McBSP qui sont des ports séries multicanaux protégés. Ils permettent la communication avec les périphériques externes. Ils ont la même structure. Ils supportent une communication full-duplex.

- L'interface de mémoire externe EMIF : Il permet l'interface avec plusieurs éléments (mémoires) externes.
- Les compteurs : Le DSP possède deux compteurs qui peuvent être synchronisés par une source interne ou externe et ils sont utilisés comme générateurs de pulsations, compteurs d'événements externes, interrupteurs du CPU après l'exécution de tâches et déclencheur du DMA/EDMA.
- Les interruptions : l'ensemble des périphériques contient jusqu'à 32 sources d'interruptions [35].

2.7. Environnement de développement (Code Composer Studio (CCS))

Le logiciel Code Composer Studio est un environnement de développement intégré (IDE) qui prend en charge les portefeuilles de microcontrôleurs (MCU) et de processeurs intégrés de TI. Le logiciel Code Composer Studio comprend une suite d'outils utilisés pour développer et déboguer des applications embarquées.

Le logiciel comprend un compilateur d'optimisation C/C++, un éditeur de code source, un environnement de génération de projet, un débogueur, un profileur et de nombreuses autre fonctionnalités.

L'IDE intuitif fournit une interface utilisateur unique qui vous guide à chaque étape du flux de développement applications.

Des outils et des interfaces familières vous permettent de démarrer plus rapidement que jamais. Le logiciel Code Composer Studio combine les avantages de la structure logicielle Eclipse avec les capacités de débogage intégrées avancées de TI, ce qui se traduit par un environnement de développement riche en fonctionnalités pour les développeurs intégrés [36].

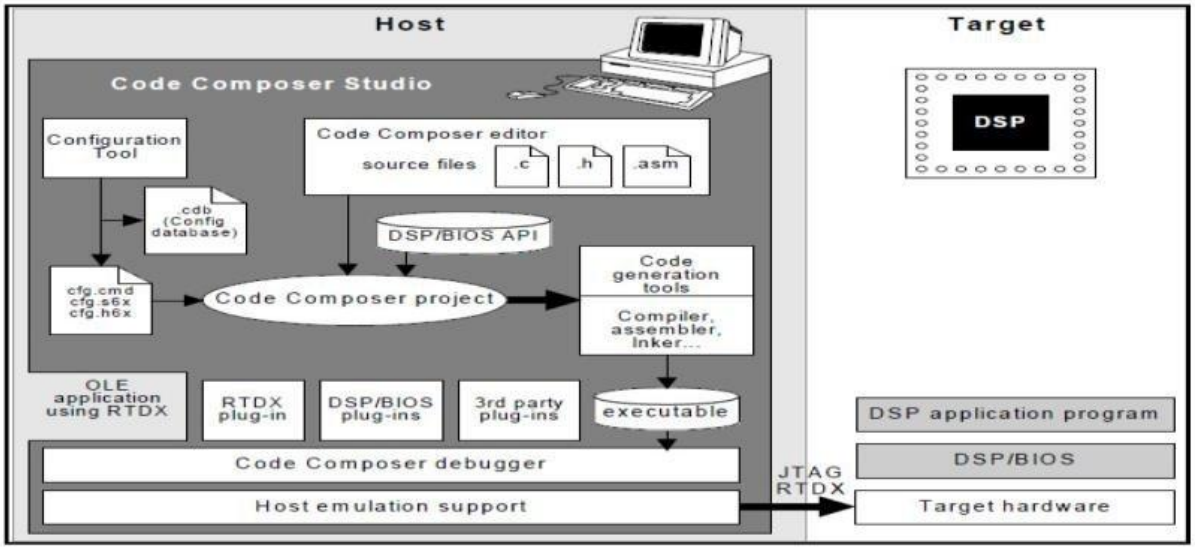

Figure 2.9 : Structure du système du développement du TMS320C6713

2.8. Configuration du développement et environnement

Les exécutables sur DSP sont élaborés en utilisant des environnements de développement intégrés (IDE) fournies par les fabricants des DSP, ils intègrent de nombreuses fonctions, telles que l'éditeur, le débogage, la gestion des fichiers de projet, et le profilage. L'environnement de développement pour Texas Instrument (TI) est appelé « Code Composer Studio ». Le constructeur TI met à disposition gratuitement le compilateur, l'assembleur et l'éditeur de liens optimiseur pour une utilisation non commerciale. La figure 2.9 montre un exemple d'un écran typique du Code Composer. Sur le côté gauche il y a la liste de tous les fichiers inclus dans le projet. Au centre de l'écran deux fenêtres affichent le code, comme un fichier C (process.c) et le code assembleur (fenêtre Disassemby).

Au bas de l'écran de l'IDE, on trouve [37] :

a) La fenêtre de **compilation/Link**, qui donne les résultats de la dernière compilation du code.

b) La fenêtre **Watch**, qui affiche les valeurs prise par deux variables.

c) La fenêtre **Register**, qui affiche le contenu de tous les registres du DSP.

d) Autres outils utiles (affichage des graphes, visualisation d'images….).

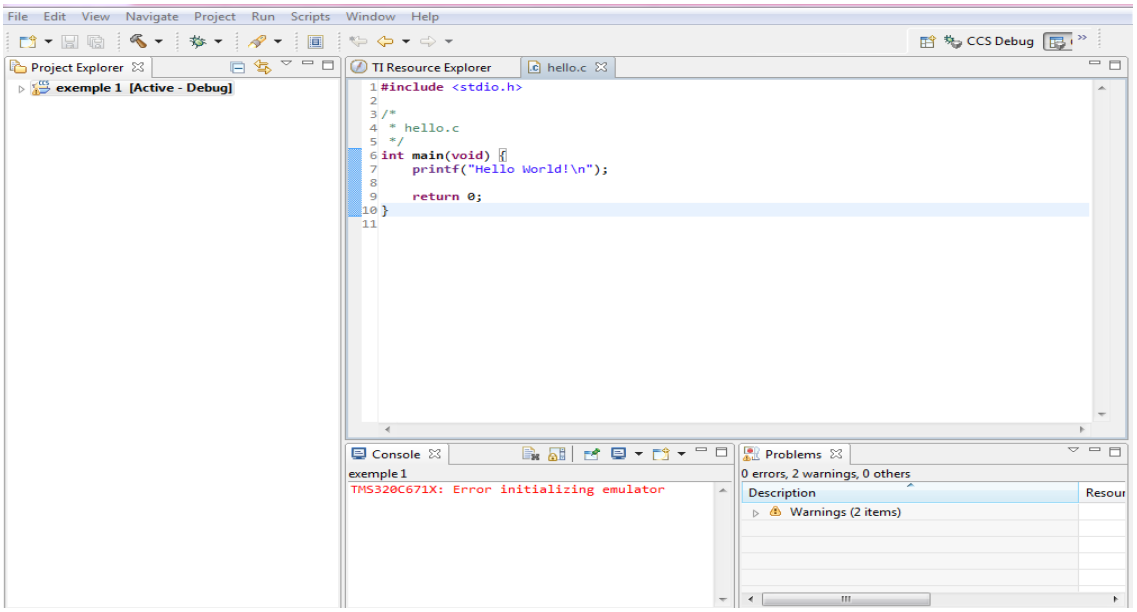

Figure 2.10 : Code Composer Studio

2.9. Conclusion

Dans ce chapitre, on a mis en avant les notions principales des DSP, ont ce basant sur l'architecture interne, et les caractéristiques du DSP TMS320C6713 qui sera exploité dans notre travail. Puis nous avons présenté l'outil software afin d'exploité notre carte DSP, cet outil est le Code Composer Studio CCS. C'est un environnement de développement intégré (IDE) utile pour créer et déboguer des programmes. De plus, il permet une analyse en temps réel des d'application.

Comme les DSP sont une partie importante du traitement numérique du signal, plus exactement ils sont un des supports clefs pour la réalisation pratique. De ce fait, dans le prochain chapitre, nous allons illustrer les étapes à suivre on exploitant cette carte pour appliquer différents types de filtres à un signal de la parole.

Chapitre III :

Résultats et Discussions

Sommaire

3.1. Introduction

Le filtrage est l'une des opérations de traitement des signaux les plus utilisées. Les DSP sont maintenant disponible pour implémenter des filtres numériques en temps réel. Le jeu d'instructions TMS320C6x et l'architecture le rend bien adapté pour de telles opérations de filtrage.

Dans ce chapitre, nous étudierons l'implémentation d'un filtre FIR sur un DSP, et nous appliquerons des filtres FIR au signal audio à l'aide de l'outil de conception de filtre Matlab et le code composer studio.

3.2. Outil Filtre Designer

Filter Designer est une interface utilisateur puissante pour la conception et l'analyse de filtres numériques RIF et RII.

Filter Designer nous permet de concevoir rapidement des filtres FIR ou IIR numériques en définissant des spécifications de performances de filtre, en important des filtres depuis notre espace de travail MATLAB ou en ajoutant, déplaçant ou supprimant des pôles et des zéros.

Filter Designer fournit également des outils d'analyse de filtres, tels que des tracés de réponse d'amplitude et de phase et des tracés de pôle zéro.

- Mise en route :

- Dans la fenêtre ***command window*** nous avons exécuté la commande : ***filterDesigner***
- La fenêtre suivante va s'ouvrir :

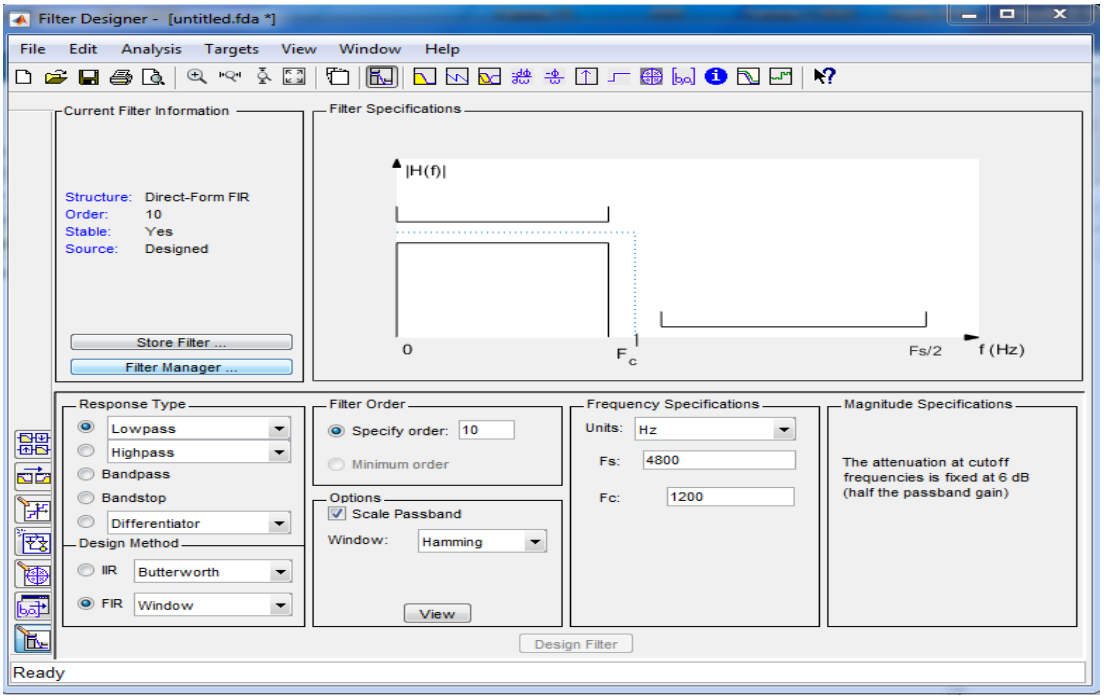

Figure 3.1 : Filtre passe-bas idéal

3.3. Affichage d'autres analyses

Une fois que nous avons conçu le filtre, nous pouvons afficher les analyses de filtre suivantes dans la fenêtre d'affichage en cliquant sur l'un des boutons de la barre d'outils :

Dans l'ordre de gauche à droite, les boutons sont :

- Magnitude response
- Phase response
- Magnitude and Phase responses
- Group delay response
- Phase delay response
- Impulse response
- Step response
- Pole-zero plot
- Filter Coefficients
- Filter Information

3.4. Exemples de conception des filtres numériques

3.4.1 Filtre passe-bas

On utilise pour la conception, l'outil Matlab « filterDesigner » avec une fenêtre de Hamming ; fréquence de coupure fc= 1200Hz et fréquenche d'échantillonnage fs=4800Hz.

Les figures ci-dessous représentent la réponse d'amplitude de ce filtre :

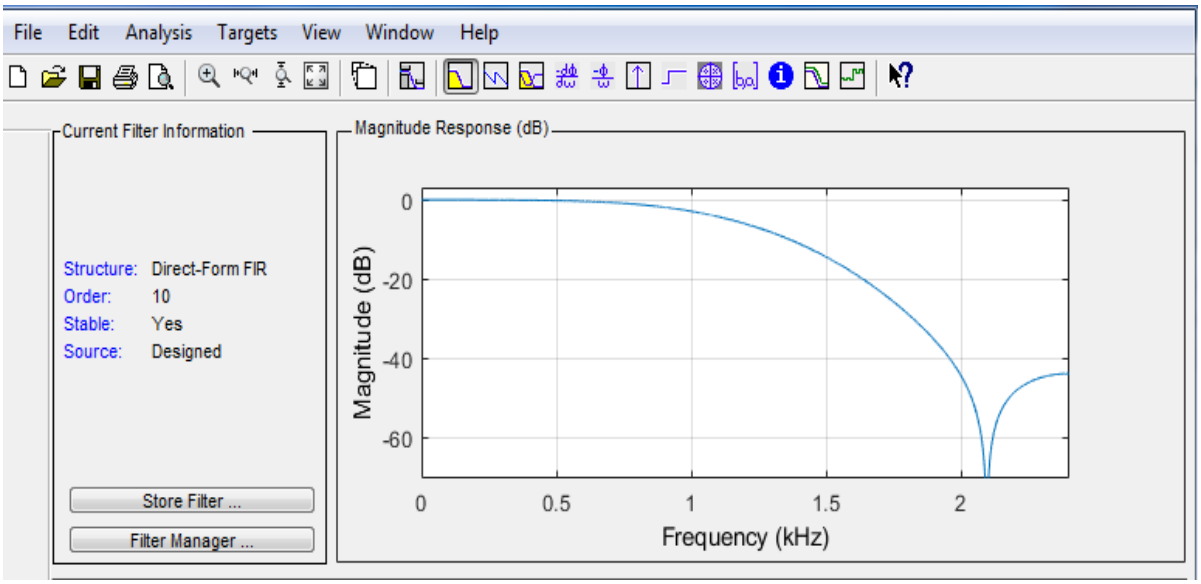

Figure 3.2 : Réponse d'amplitude d'un filtre RIF passe-bas avec fenêtre Hamming

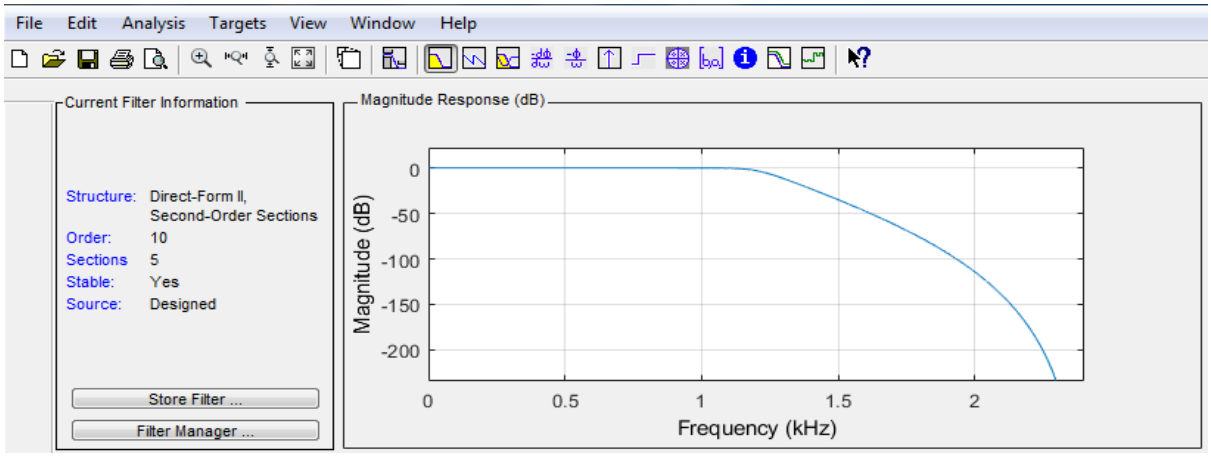

Figure 3.3 : Réponse en amplitude d'un filtre RII passe-bas avec fenêtre Hamming

La figure ci-dessous affiche la réponse en phase :

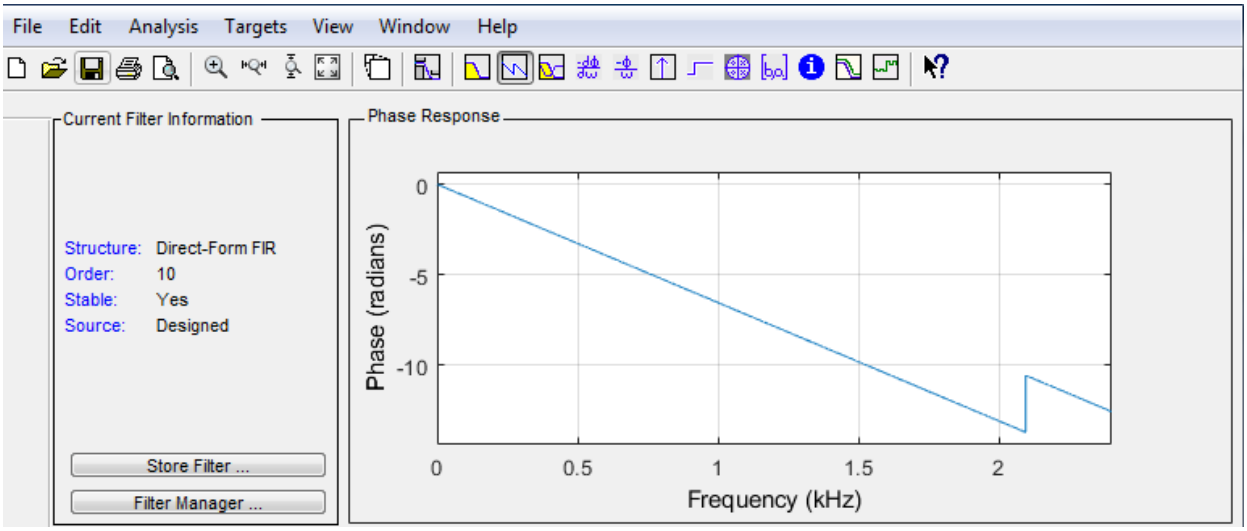

Figure 3.4 : Réponse de phase d'un filtre RIF passe-bas avec fenêtre Hamming

La figure 3.5 et la figure 3.6 représentent les réponses en amplitude et en phase.

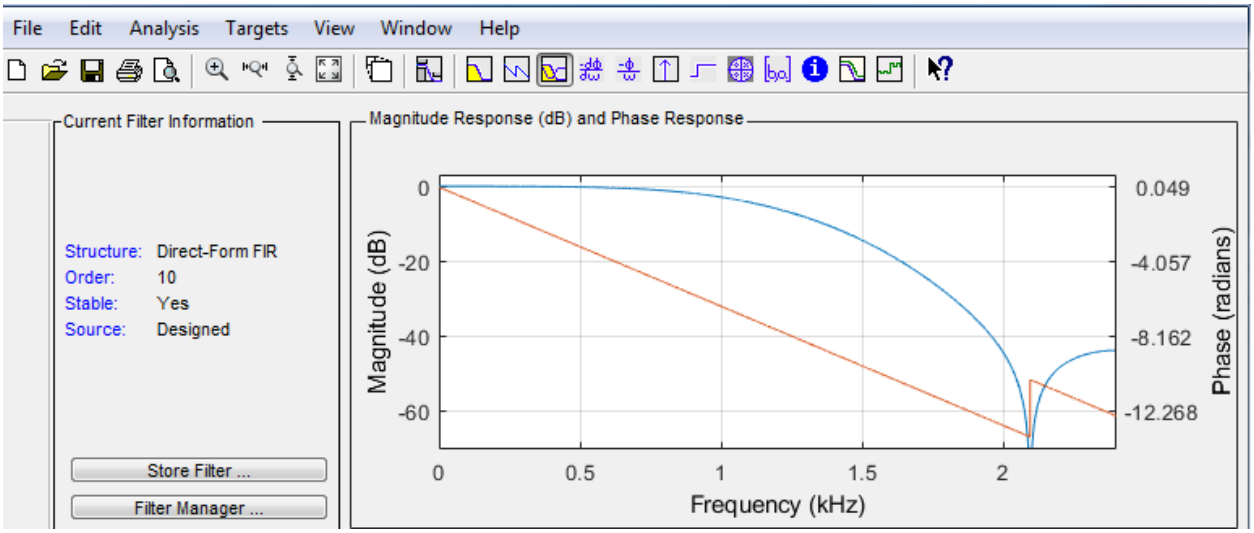

Figure 3.5 : Réponses en amplitude et en phase d'un filtre RIF passe-bas avec fenêtre Hamming

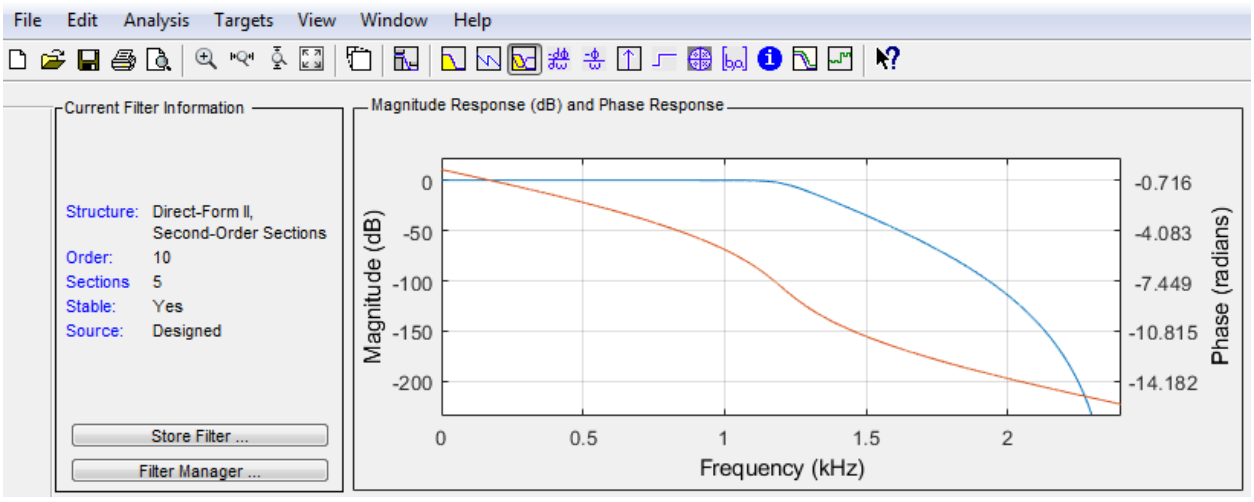

Figure 3.6 : Réponses en amplitude et en phase d'un filtre RII passe-bas avec fenêtre Hamming

Les réponses impulsionnelles pour un filtre passe bas de type RIF et RII sont tracés par les deux figures ci-dessous :

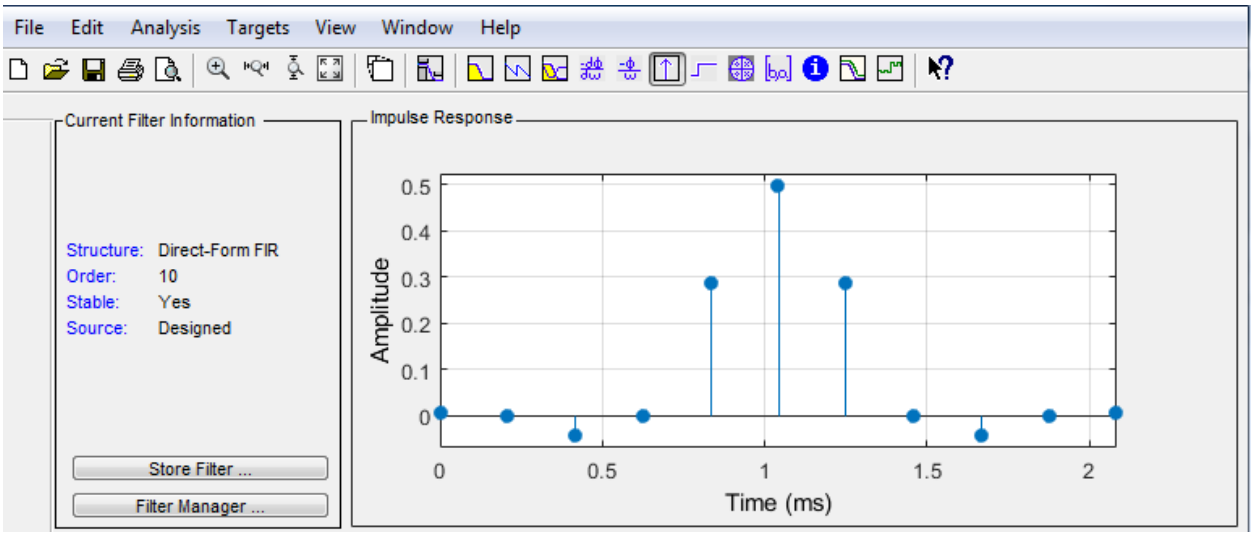

Figure 3.7 : Réponse impulsionnelle d'un filtre RIF passe-bas

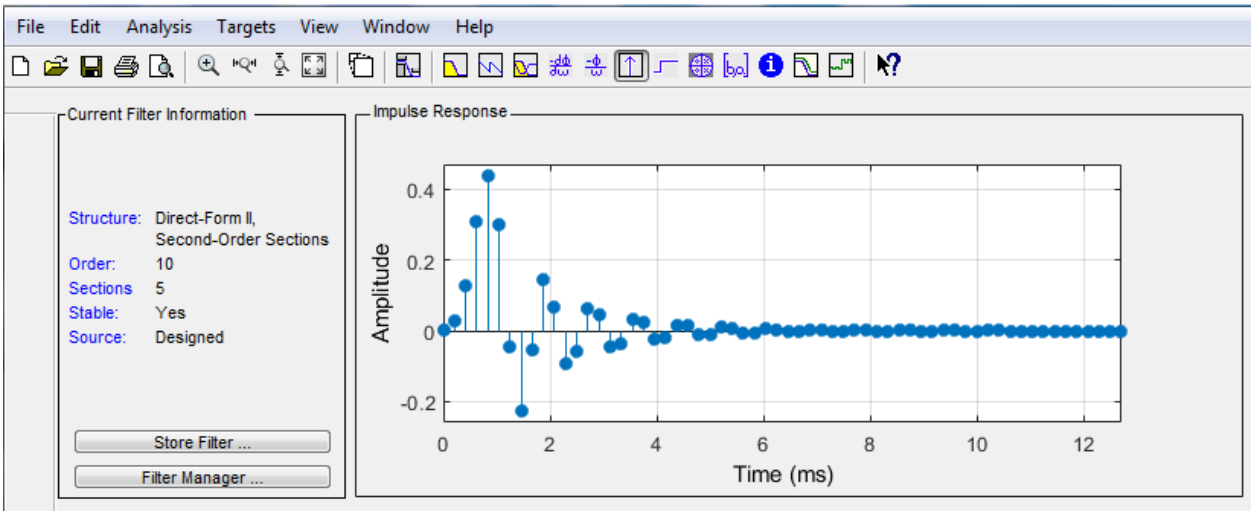

Figure 3.8 : Réponse impulsionnelle d'un filtre RII passe-bas

La fenêtre principale du filter deseigner nous permet aussi de représenter le diagramme des pôles-zéros (voir figure 3.9 et figure 3.10)

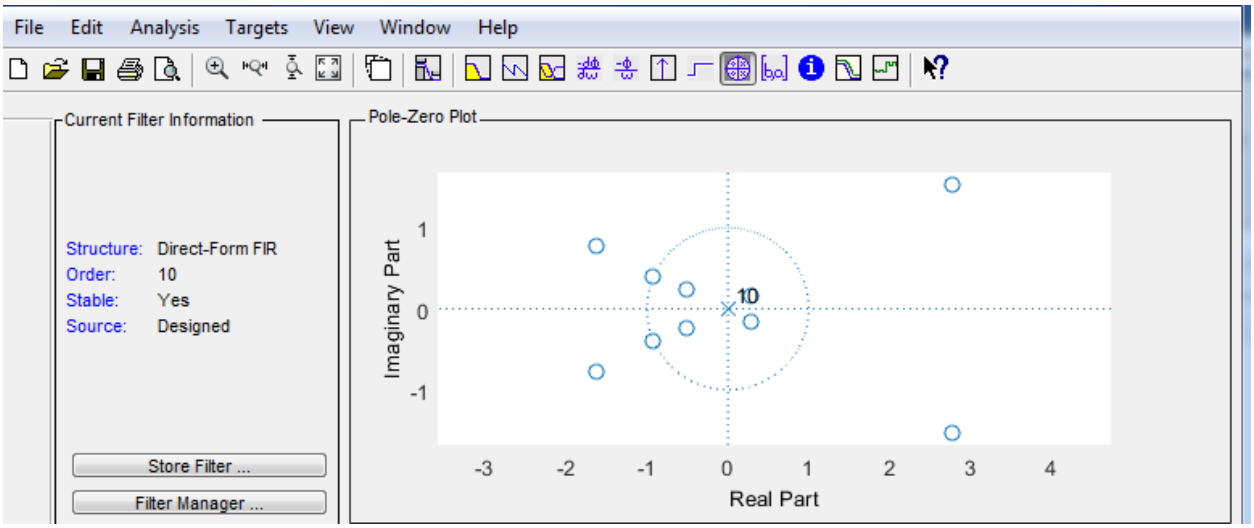

Figure 3.9 : Les pôles-zéros d'un filtre RIF passe-bas

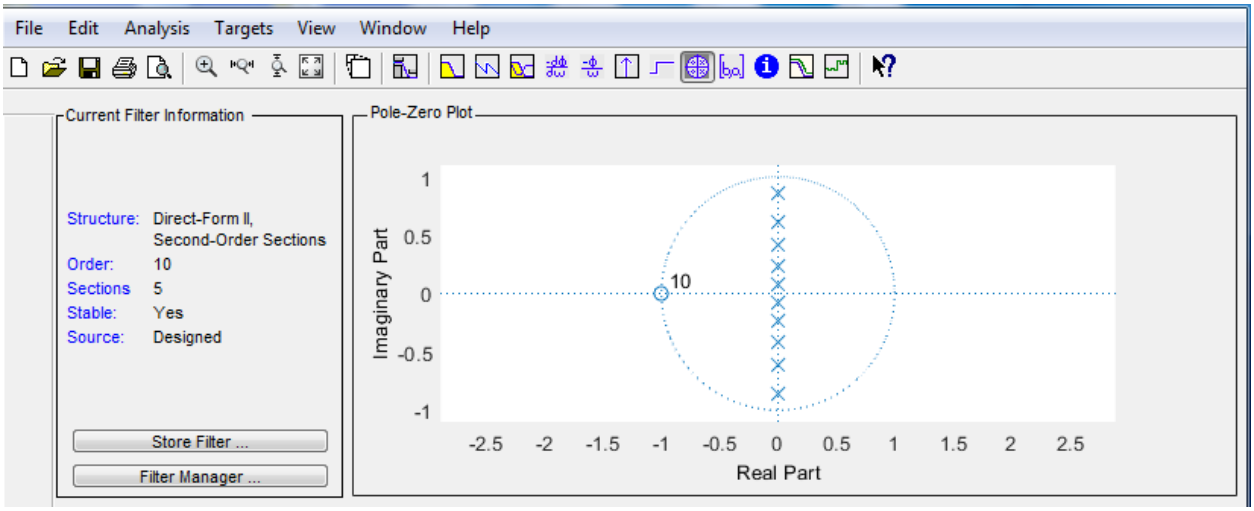

Figure 3.10 : Les pôles-zéros d'un filtre RII passe-bas

On peut aussi afficher les coefficients de notre filtre, le numérateur dans le cas d'un filtre RIF.

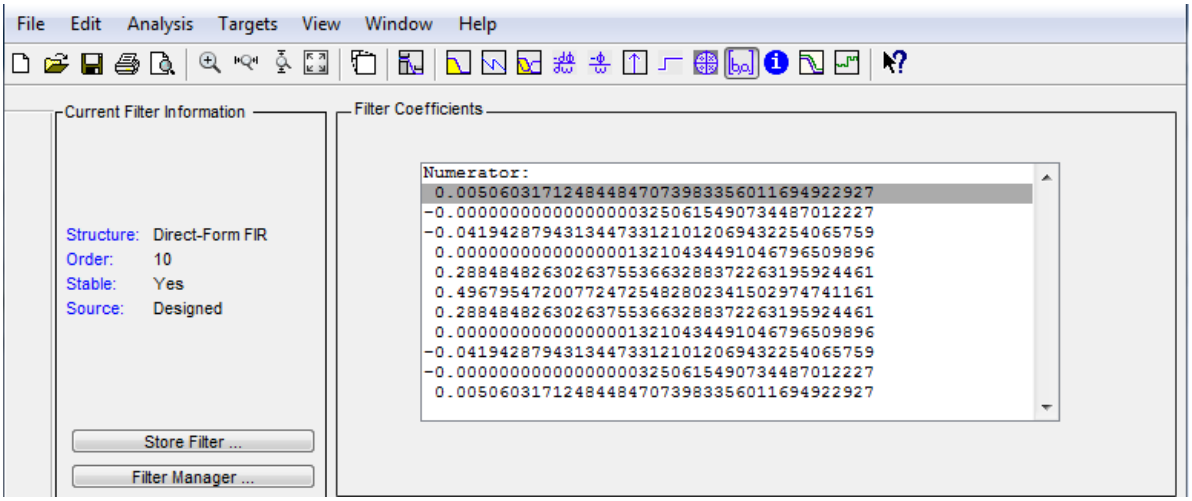

Figure 3.11 : Les Coefficients de filtre RIF passe-bas

3.4.2 Filtre passe-haut

On utilise pour la conception, de l'outil Matlab « filterDesigner » avec une fenêtre de Hamming et des fréquences de coupure et d'échantillonnage respectivement : fc= 1200Hz et fs=4800Hz.

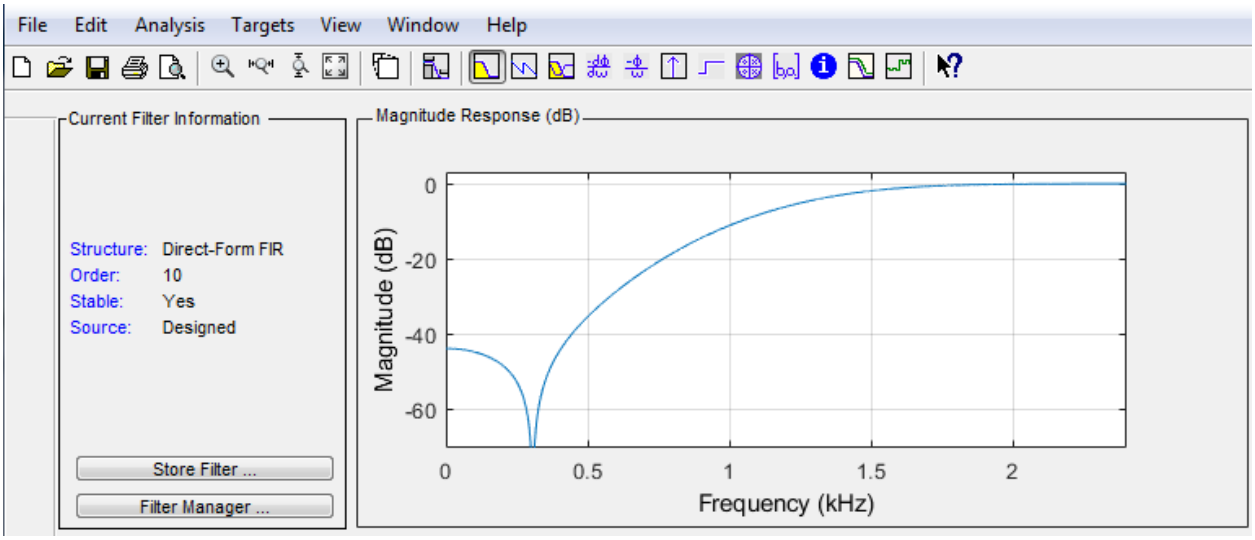

Figure 3.12 : Réponse d'amplitude d'un filtre RIF passe-haut avec fenêtre Hamming

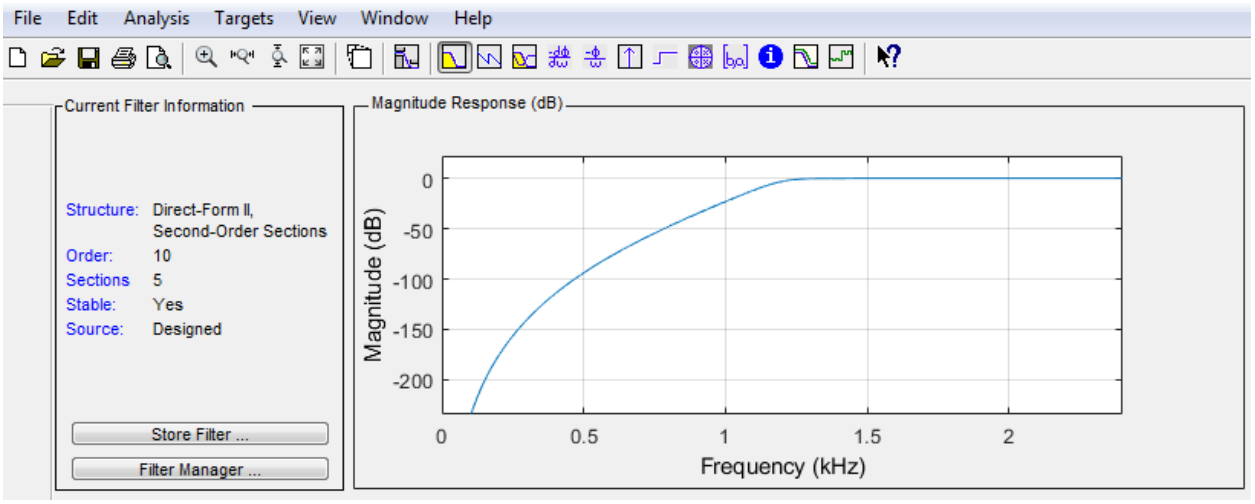

Figure 3.13 : Réponse d'amplitude d'un filtre RII passe-haut avec fenêtre Hamming

3.4.3 Filtre passe-bande

On utilise pour la conception, par l'outil Matlab « filterDesigner » avec une fenêtre de Hamming fc1= 8400Hz et fc2=13200Hz fs=48000Hz (voir figure 3.14 et figure 3.15).

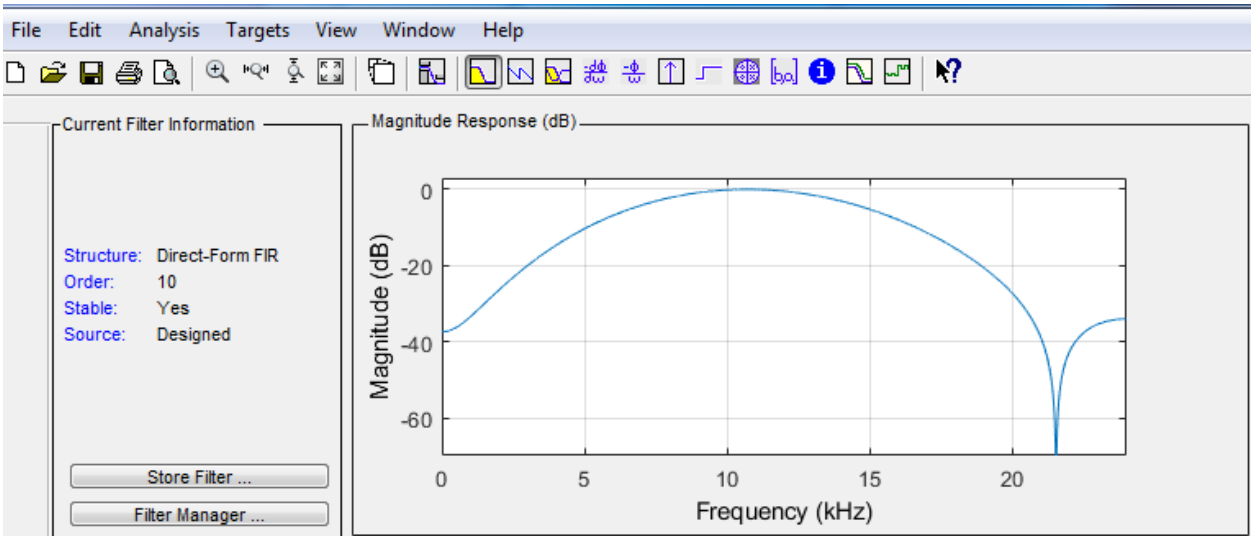

Figure 3.14 : Réponse d'amplitude d'un filtre RIF passe-bande avec fenêtre Hamming

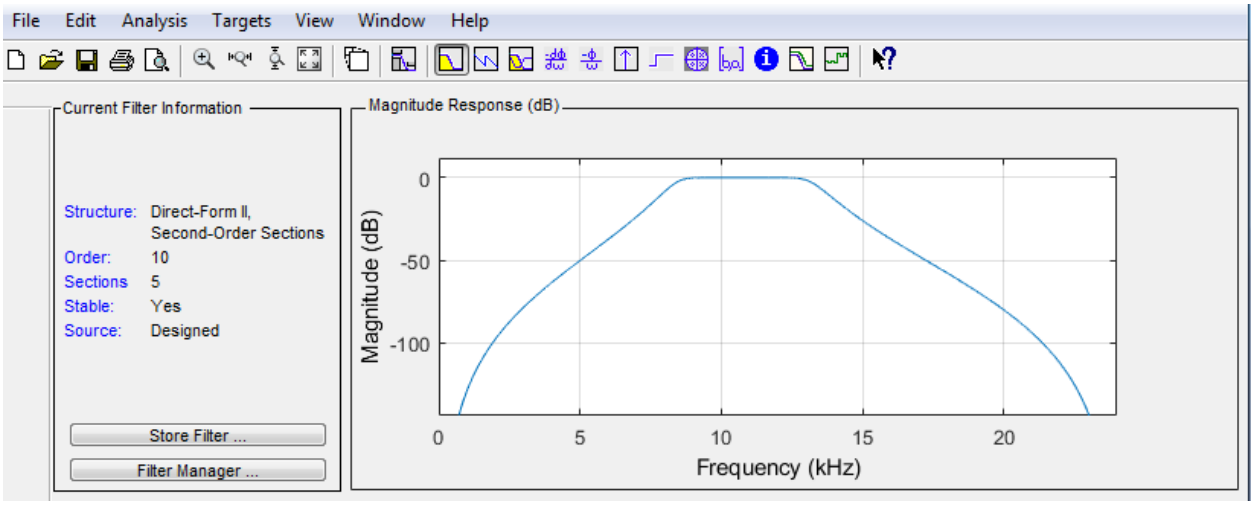

Figure 3.15 : Réponse d'amplitude d'un filtre RII passe-bande avec fenêtre Hamming

3.4.4 Filtre coupe-bande

On utilise pour la conception, par l'outil Matlab « filterDesigner » avec une fenêtre de Hamming fc1= 8400Hz et fc2=13200Hz fs=48000Hz.

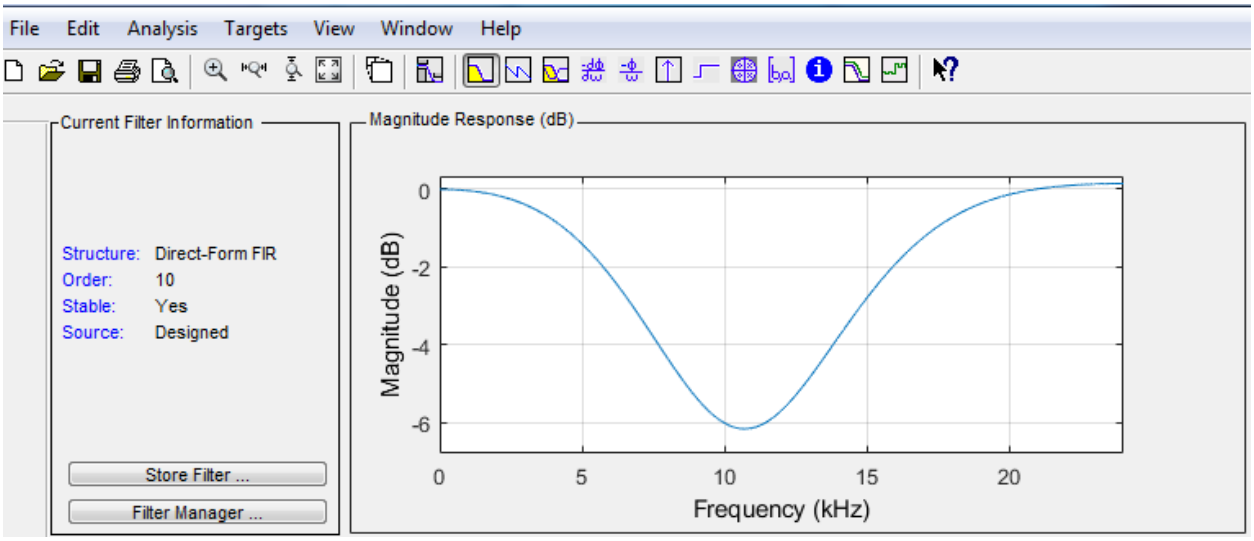

Figure 16 : Réponse d'amplitude d'un filtre RIF coupe-bande avec fenêtre Hamming

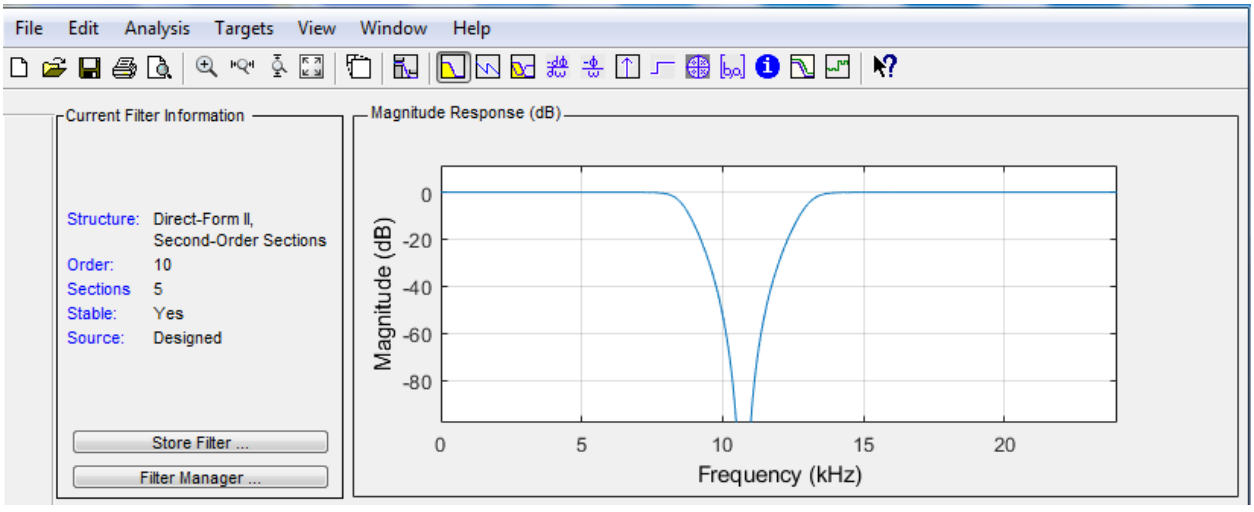

Figure 3.17 : Réponse d'amplitude d'un filtre RII coupe-bande avec fenêtre Hamming

3.5. Résultats et discussion

3.5.1. Filtre passe-bas

On utilise pour la conception d'un filtre RIF passe-bas, l'outil Matlab « filterDesigner » avec une fenêtre de blackman ; et avec une fréquence de coupure fc= 10800Hz et fréquence d'échantillonnage fs=44100Hz.

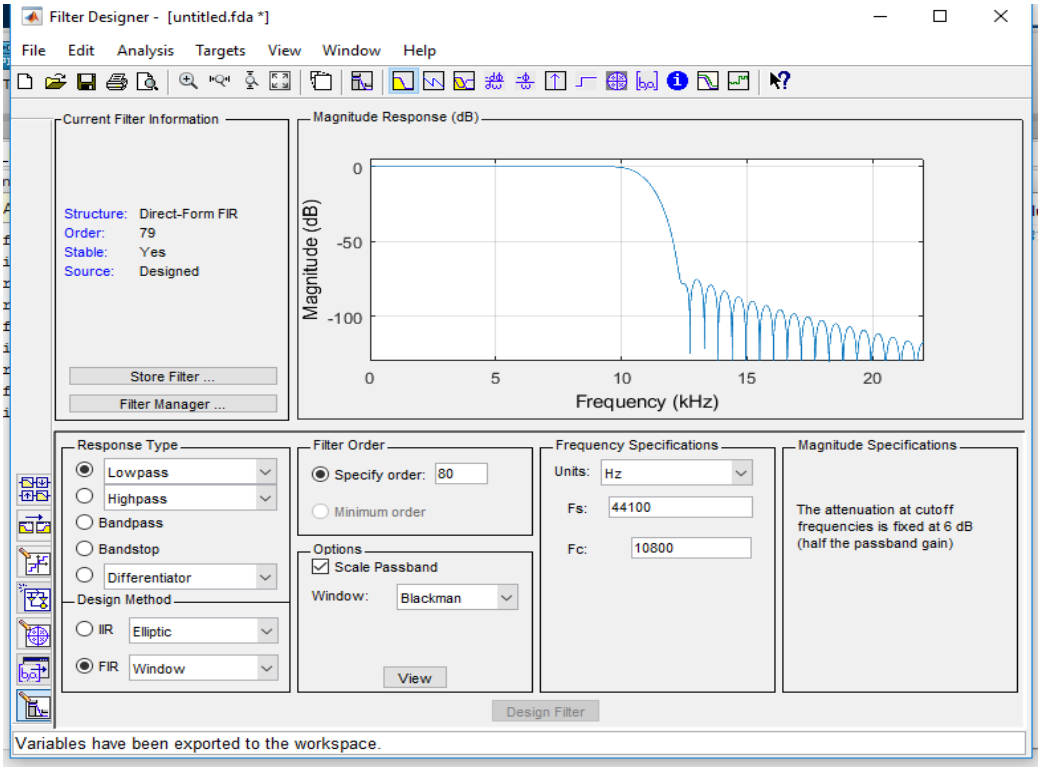

Figure 3.18 : Filtre RIF passe-bas la réponse fréquentielle en utilisant Matlab « filterDesigner » avec fenêtre de Blackman

Pour faire l'exportation des coefficients de notre filtre vers l'espace de travail de Matlab, nous devons sélectionner les coefficients dans le menu « Export » pour enregistrer les coefficients du filtre, et nous attribuons les noms des variables à l'aide du numérateur (pour les filtres RIF).

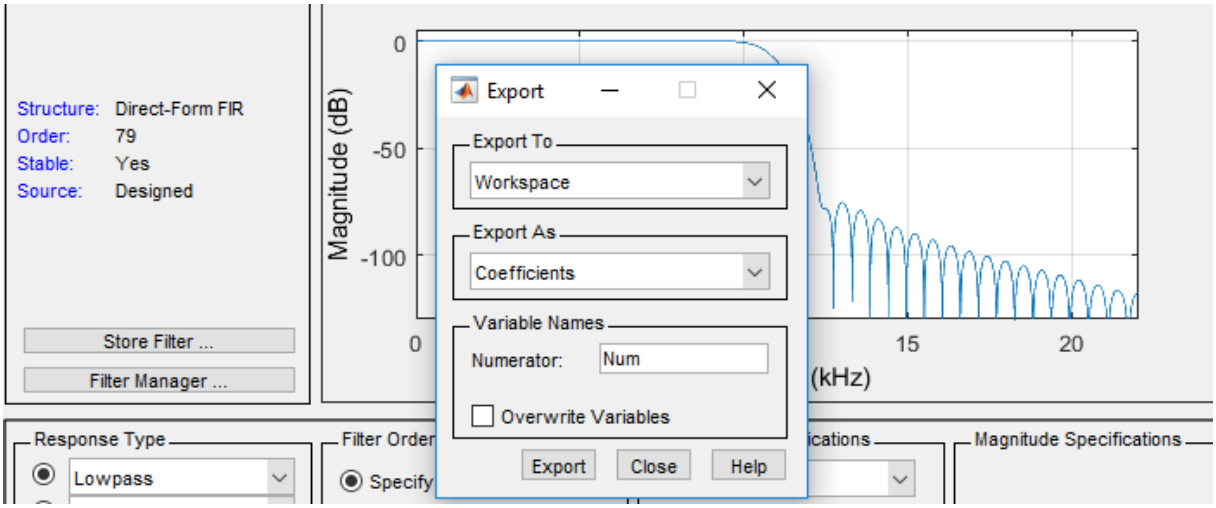

Figure 3.19 : Exportation de coefficients

Nous utilisons la fonction ***dsk_fir67*** pour créer un fichier de paramètres compatible avec le langage de programmation C.

La figure ci-dessous illustre une partie de notre programme Matlab.

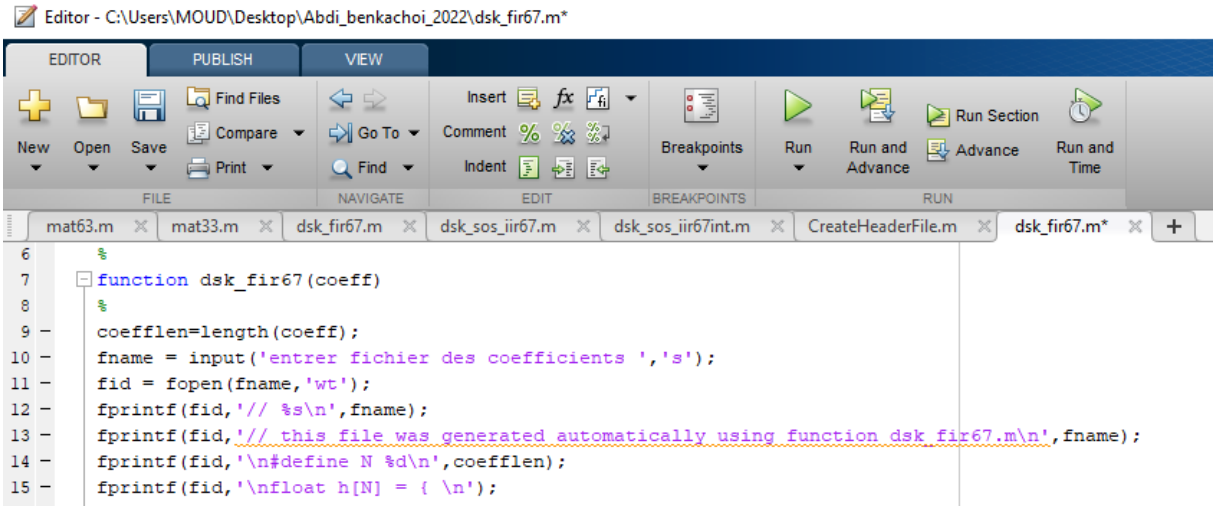

Figure 3.20 : La fonction ***dsk_fir67*** utilisée par Matlab

Nous devons appeler la fonction ***dsk_fir67*** sur la command window :

Figure 3.21 : Appel de la fonction ***dsk_fir67***

La figure ci-dessous illustre les coefficients de notre filtre RIF exploitable par code composer studio :

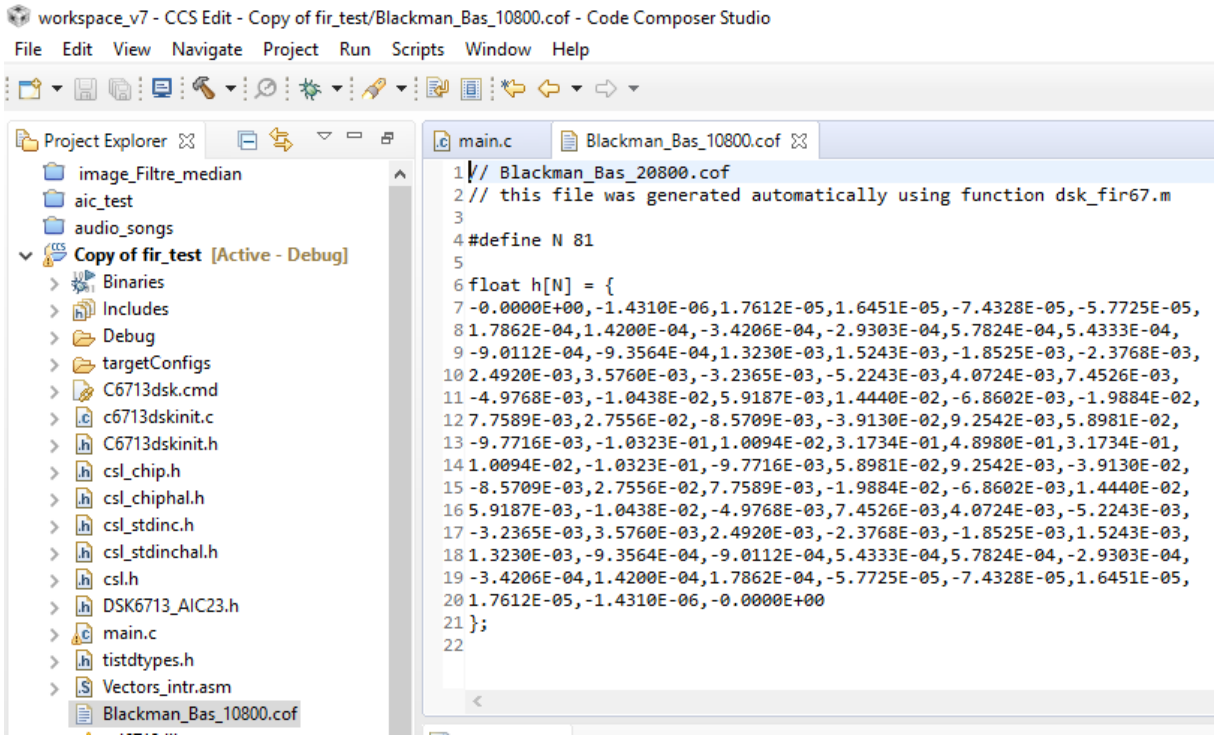

Figure 3.22 : Les coefficients de filtre RIF passe-bas

Il faut mentionner qu'un fichier de commande ****.cmd*** d'éditeur de liens « Linker » doit être ajouté à notre projet, afin de réserver la mémoire IRAM et les registres pour notre calcul.

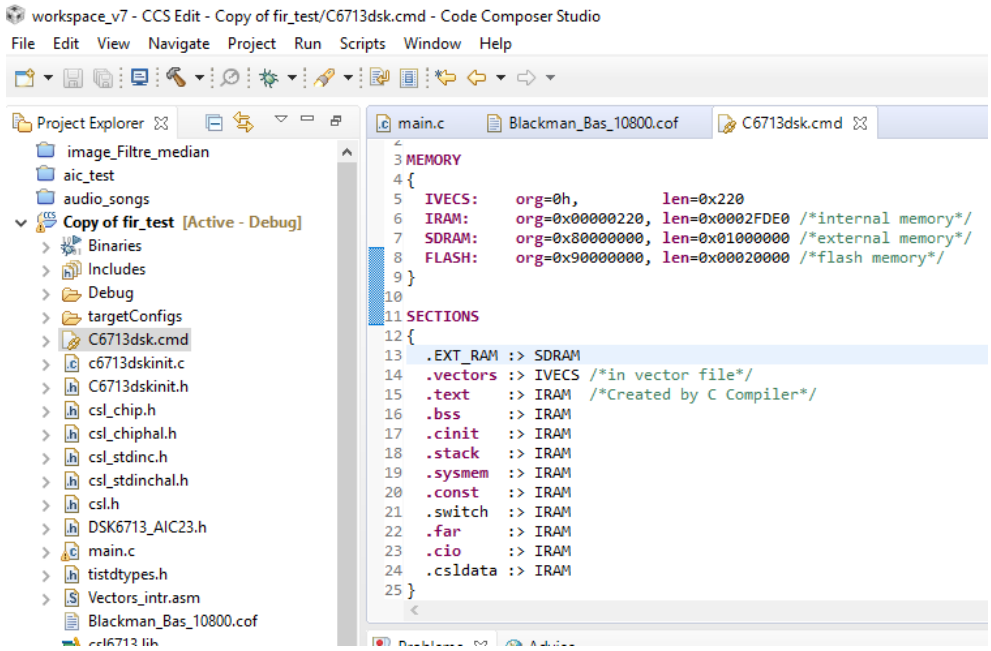

Figure 3.23 : Le fichier cmd « Linker »

La figure 3.24 représente notre fichier C, qui contient les instructions afin de réaliser un filtrage numérique, appliqué à un signal audio.

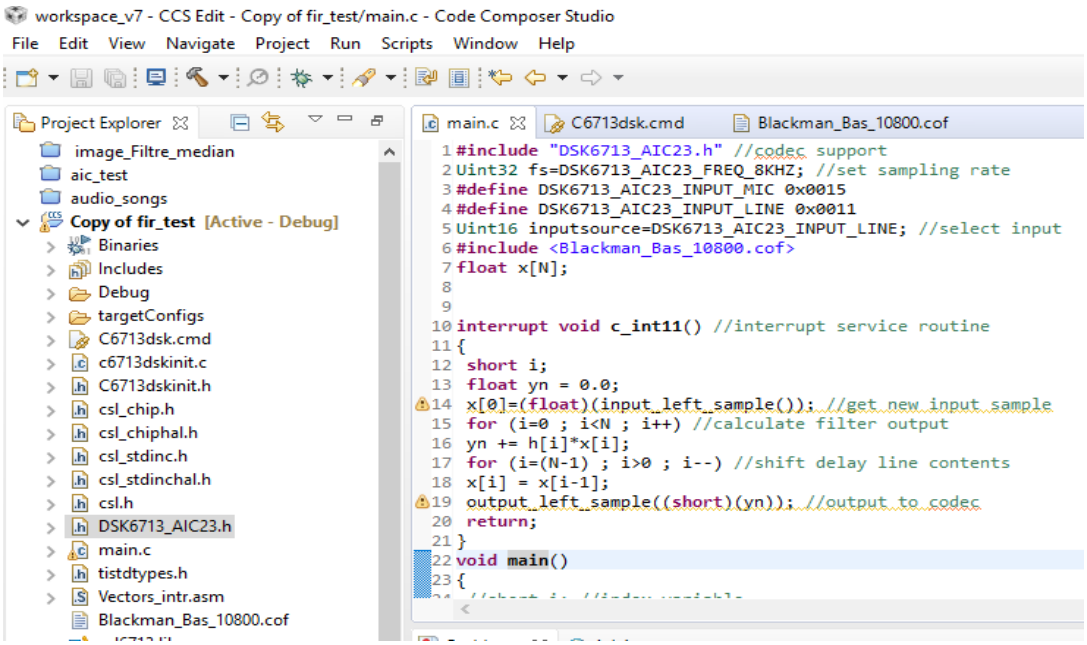

Figure 3.24 : Programme en C

3.5.2. Filtre passe-haut

On utilise pour la conception d'un filtre RIF passe-haut, l'outil Matlab « filterDesigner » avec une fenêtre de blackman ; et avec une fréquence de coupure fc= 10800Hz et fréquence d'Echantillonnage fs=44100Hz.

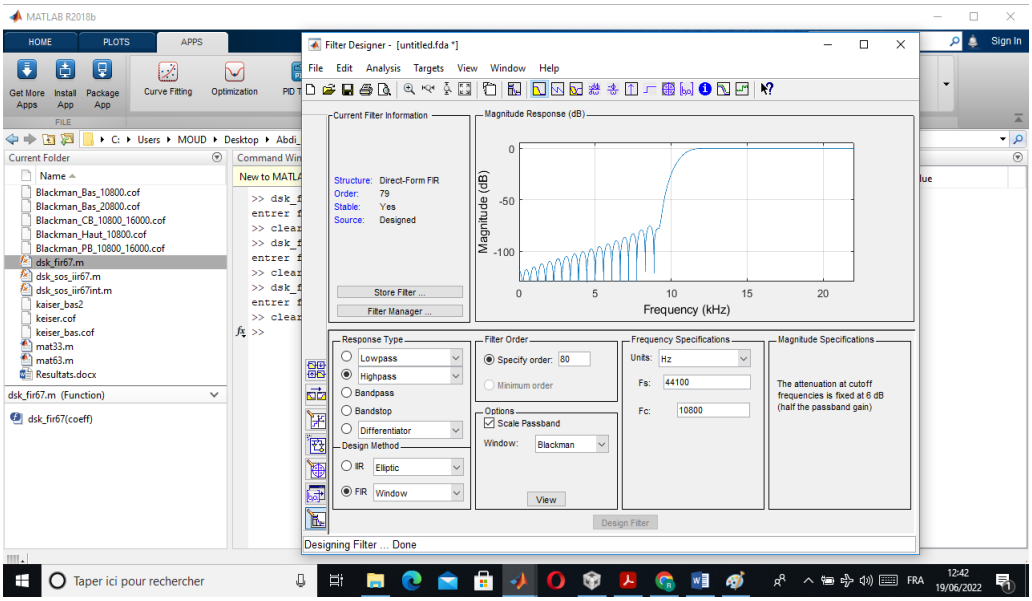

Figure 3.25 : Réponse fréquentielle en utilisant

Matlab « filterDesigner » avec fenêtre de Blackman pour filtre RIF passe-haut**.**

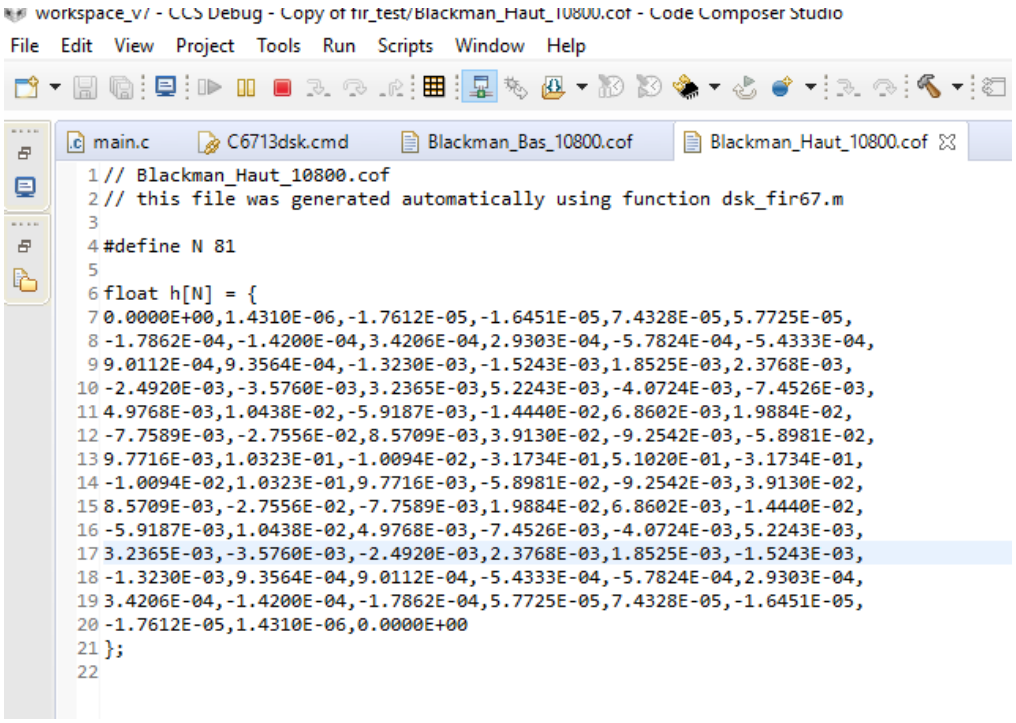

Figure 3.26 : Les coefficients de filtre RIF passe-haut

3.5.3. Filtre passe-bande

On utilise pour la conception d'un filtre RIF passe-bande, l'outil Matlab « filterDesigner » avec une fenêtre de blackman ; et avec une fréquence de coupure fc1= 10800Hz, fc2= 16000HZ et fréquence d'Echantillonnage fs=44100Hz.

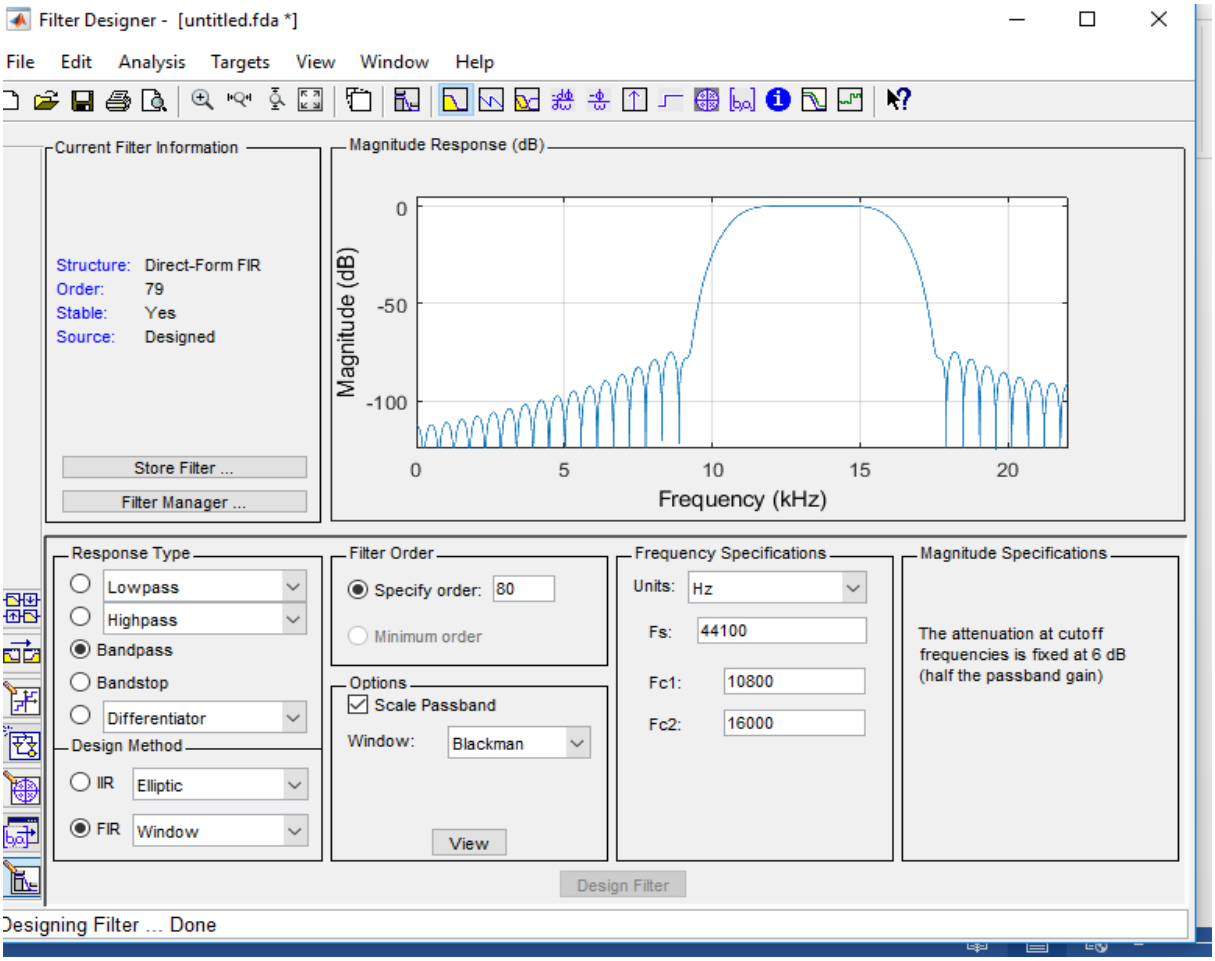

Figure 3.27 : Réponse fréquentielle d'un filtre RIF passe bande avec fenêtre de Blackman

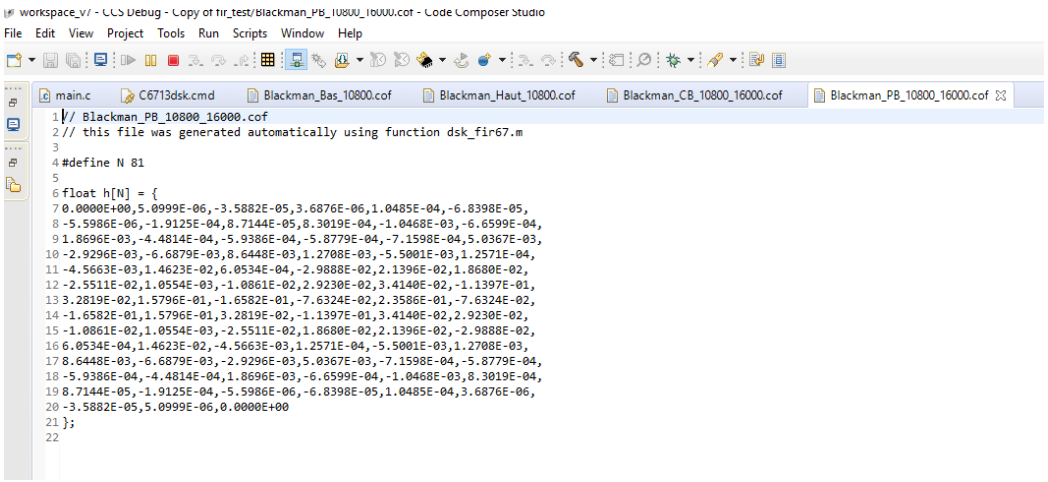

Figure 3.28 : Les coefficients de filtre RIF passe-bande

3.5.4. Filtre coupe-bande

On utilise pour la conception d'un filtre RIF coupe-bande, l'outil Matlab « filterDesigner » avec une fenêtre de blackman ; et avec une fréquence de coupure fc1= 10800Hz, fc2= 16000HZ et fréquence d'Echantillonnage fs=44100Hz.

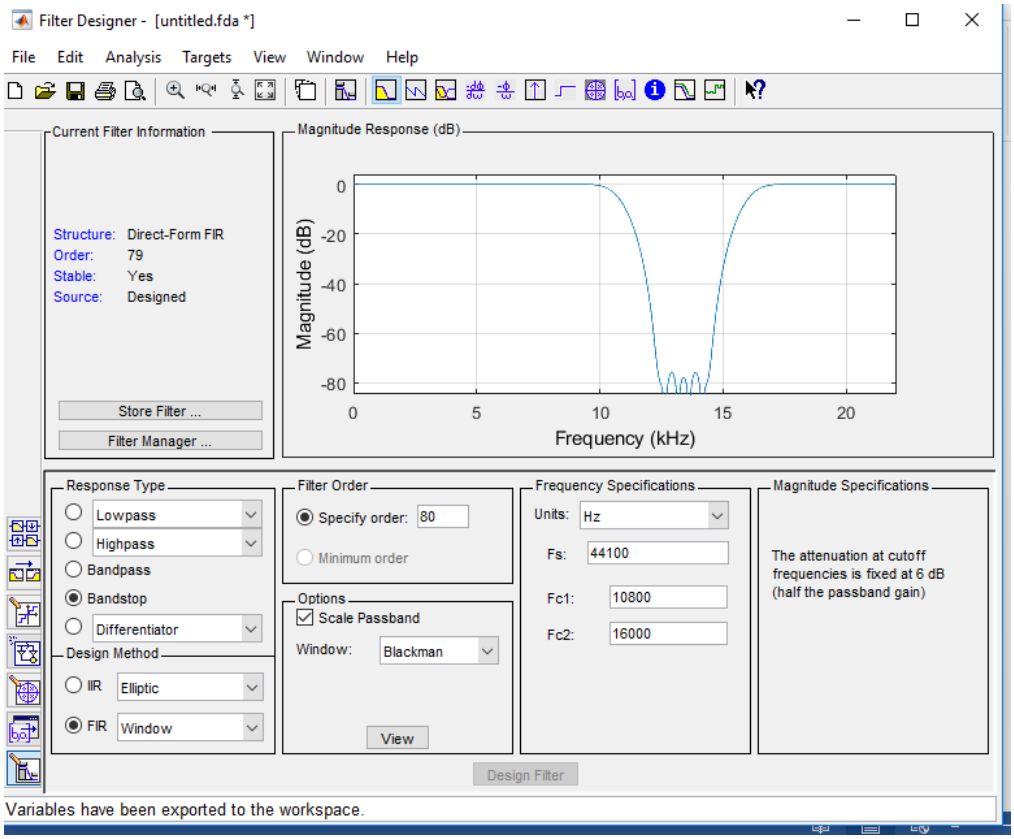

Figure 3.29 : Filtre FIR coupe bande (réponse fréquentielle d'amplitude)

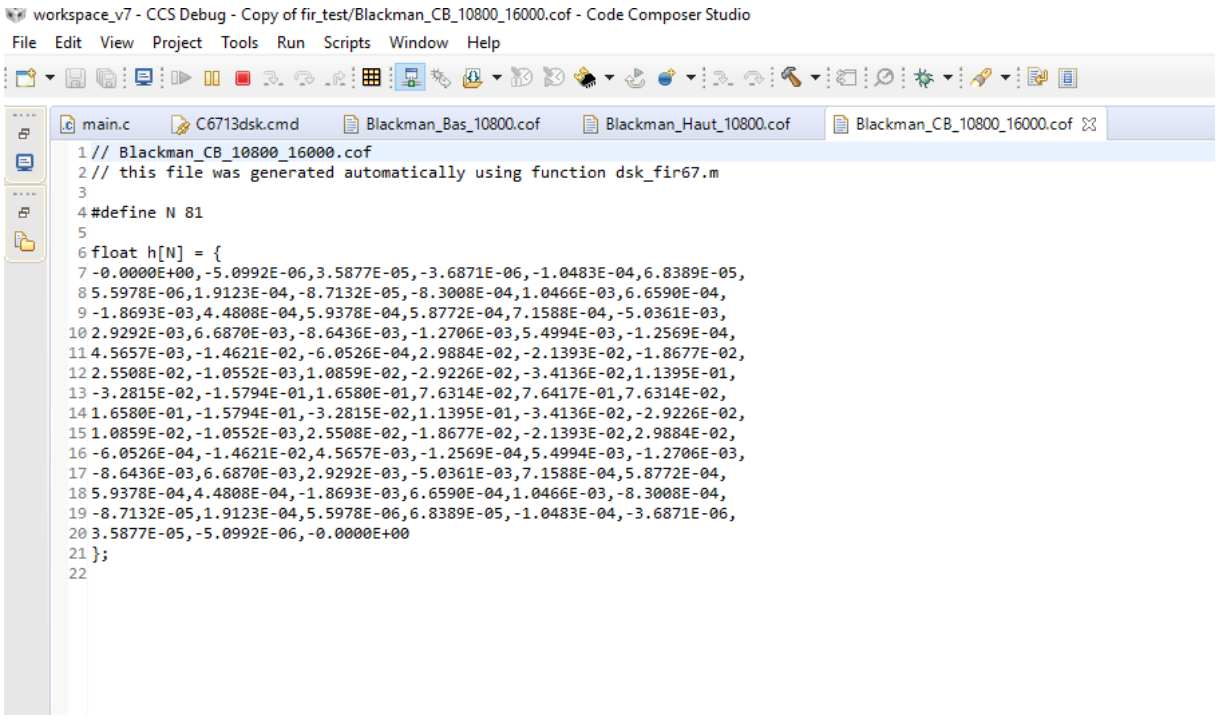

Figure 3.29 : Les coefficients de filtre RIF coupe-bande

3.6. Conclusion

Dans ce chapitre nous avons illustré les étapes à suivre pour que nous puissions faire la conception pratique de quatre filtre numérique passe bas, passe haut, passe bande et coupe bande de type RIF par la méthode de la transformée de Fourier et fenêtrage (fenêtre Blackman comme exemple). Ensuite nous appliquons nos filtres sur un signal audio en temps réel.

Pour cet objectif réalisé nous avons exploité la carte DSP TMS3206713 DSK pour implémenter nos filtres numériques. Cette application démontre que le jeu d'instructions TMS320C6x et son architecture le rend bien adapté pour de telles opérations de filtrage.

Conclusion Générale

Conclusion Générale

Les DSP sont utilisés dans la plupart des applications du traitement numérique du signal en temps réel. On les trouve dans les modems (modem RTC, modem ADSL).les téléphones mobiles, les appareils multimédia (lecteur MP3), les récepteurs GPS... Ils sont également utilisés dans des systèmes vidéo, les chaines de traitement de son, partout où l'on reçoit un signal complexe que l'on doit modifier à l'aide du filtrage. Etant donné l'explosion des données digitale dans le monde d'aujourd'hui.

L'utilisation des processeurs DSP est devenue crucial. La plupart des kits de développement disponibles sur le marché présentent un coût élevé. La carte du kit TMS320C6713 DSK lowcost avait été utilisé afin de filtré un signal audio en temps réel. L'outil software Code Composer Studio CCS, est un environnement de développement intégré (IDE) utile pour créer et déboguer des programmes (en langage C ou en assembleur). De plus, il permet une analyse en temps réel des applications réalisées.

Afin de réalisé notre projet, nous avons implémenté quatre filtres numériques RIF (filtre passe-bas, filtre passe-haut, filtre passe bande, filtre coupe bande) par la méthode de transformée de Fourier et fenêtrage par la fenêtre de Blackman sur la carte DSP TMS320C6713 DSK.

Notre application réaliser démontre que le jeu d'instructions TMS320C6x et son architecture le rend bien adapté pour de telles opérations de filtrage en temps réel.

Dans un travail de futur, nous estimons d'implémenter des modulations numériques comme l'OFDM sur une carte DSP.

Bibliographies

[1] L.Schwartz-Méthodes mathématiques pour les Sciences Physiques.Ed.Hermann;

1961.

[2] https://www.astro.rug.nl/~vdhulst/SignalProcessing/Hoorcolleges/college03.pdf .(consulté le 4/03/2022)

[3] http://perso.univ-lemans.fr/~nerrien/Phy308aA_5_C1.Pdf .(consulté le 4/03/2022)

[4] M.Beallanger ‹‹traitement numérique du signal›› 9^{e}édition by F Truchetet.

[5] B.Picinbono – Eléments deThéorie du Signal.Ed.Dunod, 1977.

[6] P.Gaillard ‹‹Analyse et traitement du signal››1990.

[7] www.isir.upmc.fr › UserFiles › File › zarader › Cours de traitement du signal ZARADER J.L.2008/2009 (consulté le 11/03/2022)

[8]https://www.mathenvideo.fr/category/bts-2nde-annee/la-transformee-de-fourier-discrete/tfd-inverse/ (consulté le 20/04/2022)

[9] https://nicolas.thiery.name/DESS/Notes/Cours4.pdf (consulté le 16/03/2022)

[10]http://w3.cran.univ-lorraine.fr/perso/hugues.garnier/Enseignement/TdS/I-TdS-Analyse_filtres_num.pdf (consulté le 17/04/2022)

[11] http://eavr.u-strasbg.fr/~laroche/student/MasterI (consulté le 19/04/2022)

[12] http://www.info2.uqam.ca/~boukadoum_m/MIC4220/Notes/6-MIC4220_RIF.pdf (consulté le 21/03/2022)

[13]http://benoit.decoux.free.fr/ENSEIGNEMENT/SIGNAL/CNAM/CNAM_cours_support_3.pdf (consulté le 23/03/2022)

[14] http://dspace.univ-setif.dz (consulté le 21/03/2022)

[15] https://stringfixer.com/fr/Linear_phase (consulté le 22/03/2022)

[16] https://www.larousse.fr/dictionnaires/francais/faiblesse/32677 (consulté le 22/03/2022)

[17]http://w3.cran.univ-lorraine.fr/perso/hugues.garnier/Enseignement/TdS/J-TdS_Conception_filtres_num.pdf (consulté le 01/04/2022)

[18]https://elearning-deprecated.univ-annaba.dz/pluginfile.php/61329/mod_resource/content/
(consulté le 01/04/2022)

[19] http://www.info2.uqam.ca/~boukadoum_m/MIC4220/Notes/7-MIC4220_RII_prof.pdf (consulté le 04/04/2022)

[20]https://www.f-legrand.fr/scidoc/docimg/numerique/filtre/rii/rii.html (consulté le 29/03/2022)

[21] https://stringfixer.com/fr/Bilinear_transform (consulté le 04/04/2022)

[22] https://homepages.laas.fr/adoncesc/SystemEmbed/Filtrage.pdf(consulté le 28/03/2022)

[23] https://www.emse.fr ›~dutertrePDF Conversions analogique – numérique. (consulté le 25/04/2022)

[24]https://www.insyte.website/pedagogie/sequencePeda/interagirAvecEnvironnement/cours/ Conversion Analogique – Numérique. Pdf (consulté le 07/04/2022)

[25]https://www.emse.fr/~dutertre/documents/cours-convertisseurs.pdf(consultéle 10/04/2022)

[26]https://elearningdeprecated.univannaba.dz/pluginfile.php/62576/mod_resource/content/1/Chapitre%20DSP-fezari-a.pdf (consulté le 10/04/2022)

[27]https://studylibfr.com/doc/1614079/i---sp%C3%A9cificit%C3%A9s-et caract%C3%A9ristiques-des-dsp (consulté le 12/04/2022)

[28] https://www.technologuepro.com/cours-dsp/chapitre-2-Caracteristiques-des-DSP.pdf(consulté le 11/04/2022)

[29] Wade.J.G : <<codage et traitement de signal l'exemple des systèmes vidéo numérique>>, Paris : Masson ,1991 (consulté le 19/04/2022)

[30] Arab Naima, diplôme d'ingénieur d'état : <<implémentation d'un DSP TMS320LF2407A en vue d'autopiloté une machine synchrone à aimants permanents >>, université de tizi ouzou, 2007. (consulté le 21/04/2022)

[31] Mémoire de fin d'études de Master Thème Implémentation d'algorithmes DSP sur la carte ARDUINO pour un signal ECG. Académique Spécialité : Instrumentation Université Mouloud Mammeri de Tizi-Ouzou

https://www.ummto.dz/dspace/bitstream/handle/ummto/6564/BraikSabrina_GharoutOuiza.pd

[32] M. Correvon(2010). Introduction aux DSP orientés & applications industrielle - *Notes du cours* [Présentation PDF] Repéré dans le site www.iai.heig-vd.ch.

[33]https://www.google.com/search?q=themarvinprojet.free.fr%3EdspLes+DSP+Digital+Signal+Processing+.... (consulté le 20/04/2022)

[34] https://cnx.org/contents/F6DqKhgX@6.3:AOtE5jR9@2/Code-Composer-Studio-v3-3-DSP-BIOS-and-C6713-DSK (consulté le 01/05/2022)

[35] Rulph Chassaing(2005). Digital Signal Processing and Applications with the C6713 and C6416. USA, New Jersey: John Wiley & Sons, Inc.

[36] https://www.ti.com/tool/CCSTUDIO (consulté le 01/05/2022)

[37] Implémentation d'un corrélateur d'image sur TMS320C6713 DSK, PFE2007,UDL, Algérie

[38] S. Tretter, Communication System Design Using DSP Algorithms: With Laboratory Experiments for the TMS320C6701 and TMS320C6711, Kluwer Academic Publishers, 2003.