

Thème 2 : Filtrage numérique

Notions :

- Comprendre la différence entre les filtres RII et RIF
- comprendre l'intérêt d'un filtre anti-repliement
- (comprendre les formats virgule fixe, virgule flottante et QXX)
- comprendre la notion de retard de groupe
- (comprendre la génération d'un effet audionumérique : l'écho)

I- Introduction : Intérêt et types des filtres numériques

Un **filtre numérique** est un **algorithme de calcul** qui fait correspondre à une suite d'échantillons $x(n)$ une suite d'échantillons $y(n)$ (exemple : $y(n) = \frac{x(n) + x(n-1)}{2}$ ou encore $y(n) = \frac{y(n-1) + x(n)}{2}$). Les échantillons $x(n)$ proviennent de l'échantillonnage d'un signal $x(t)$

continu du temps. En toute rigueur nous devrions écrire les $x(n)$ sous la forme $x(nT_e)$ où $T_e = \frac{1}{F_e}$

est la période d'échantillonnage. C'est pour ne pas alourdir l'écriture qu'on les note simplement $x(n)$, idem avec les échantillons de sortie $y(n)$ au lieu de $y(nT_e)$. Par extension, on appelle souvent filtre numérique, l'ensemble de la chaîne de traitement (figure 1) : filtre antirepliement, échantillonneur-bloqueur, CAN, processeur et CNA. Le but est de réaliser une opération de filtrage comme on pourrait le faire avec des filtres actifs ou passifs.

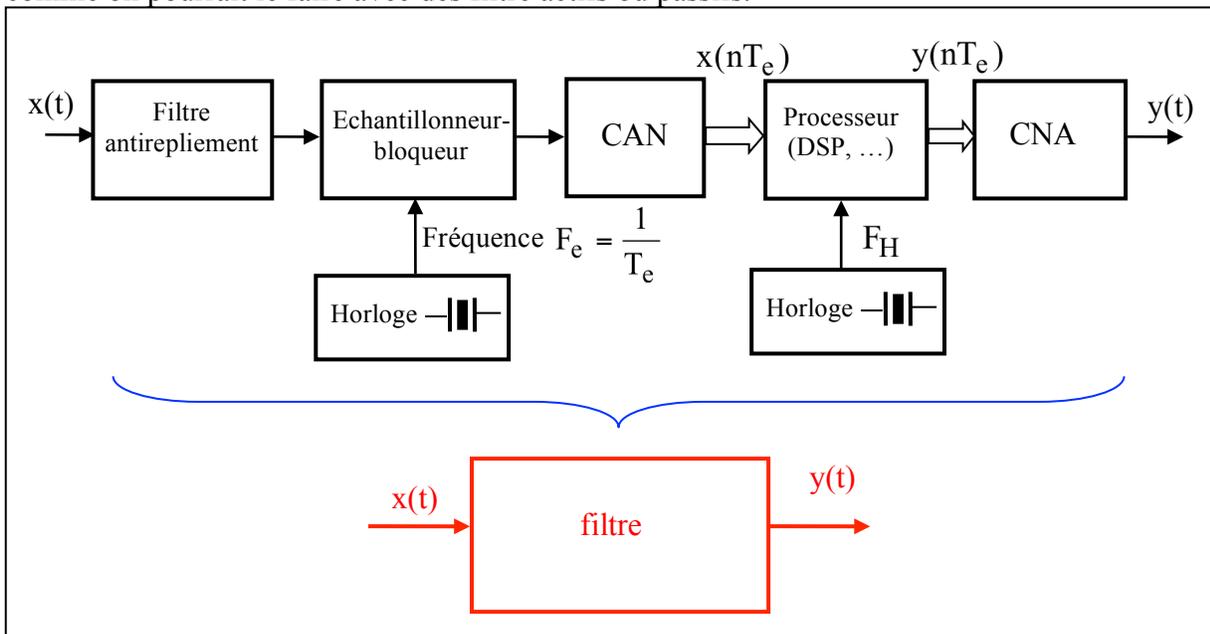


Figure 1 : Chaîne de traitement numérique d'un signal : filtrage numérique

Le filtrage numérique apporte une grande souplesse de traitement : les résistances et condensateurs des filtres actifs sont remplacés par des coefficients dans un algorithme de calcul. Le filtrage numérique permet de réaliser des filtres impossibles à réaliser avec des circuits analogiques, par exemple réaliser un **retard pur**. Bien sûr, il n'y a pas que des avantages, en effet, il faut échantillonner le signal $x(t)$ et donc satisfaire le **critère de Shannon**, c-à-d échantillonner à une fréquence F_e au moins deux fois supérieure à la fréquence maximale contenue dans le spectre de $x(t)$. Il est par conséquent difficile de faire du filtrage numérique aux hautes fréquences. Il faut par ailleurs convertir les échantillons en un nombre de bits fini ce qui introduit un **bruit de quantification**. Pour terminer, il faut disposer d'une méthode pour calculer les coefficients de la relation reliant les $x(n)$, $x(n-1)$, ... $y(n-1)$, $y(n-2)$, ... aux $y(n)$. Faire la synthèse d'un filtre numérique, c'est trouver les coefficients b_i et a_j de la relation :

$$y(n) = \sum_{i=0}^{i=M} b_i x(n-i) + \sum_{j=1}^N a_j y(n-j) \quad (1)$$

de telle manière que l'opération réalisée soit par exemple équivalente à un filtre passe-bas de fréquence de coupure 1kHz.

Question 1 : On a procédé à l'échantillonnage d'un signal continu $x(t)$ et on a obtenu la suite d'échantillons suivants :

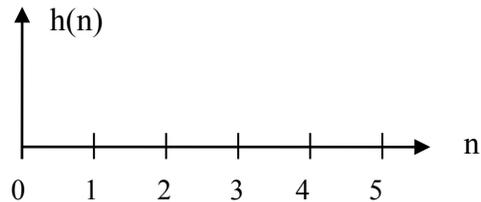
instants d'échantillonnage	0	T_e	$2T_e$	$3T_e$	$4T_e$	$5T_e$	$6T_e$	$7T_e$	etc
$x(n)$	1	2	1	2	1	2	1	2	---
$y(n)$									

L'algorithme implémenté dans le processeur est : $y(n) = \frac{x(n) + x(n-1)}{2}$, compléter la ligne des $y(n)$. Intuitivement, de quel type de filtre s'agit-il ? passe-bas, passe-haut, passe-bande ou réjecteur ? justifier votre réponse. Dessiner le signal $y(t)$ en sortie du CNA.

Le filtre $y(n) = \frac{x(n) + x(n-1)}{2}$ est un cas particulier de filtre **RIF** (**R**éponse **I**mpulsionnelle **F**inie, ou **FIR** en anglais pour **F**inite **I**mpulse **R**esponse). Il est appelé ainsi car un échantillon de sortie $y(n)$ ne fait appel qu'à un nombre fini d'échantillons d'entrée $x(n)$, $x(n-1)$, ... Ce type de filtre n'a pas d'équivalent en filtrage analogique.

Question 2 : On appelle **réponse impulsionnelle** $h(n)$ d'un filtre la réponse $y(n)$ à la suite d'échantillons d'entrée $x(n) = \delta(n)$, $\delta(n)$ est l'équivalent de l'impulsion de Dirac pour l'analogique. Déterminer les éléments $h(n)$ de la réponse impulsionnelle du filtre obéissant à l'équation $y(n) = \frac{x(n) + x(n-1)}{2}$ et compléter le graphe ci-dessous.

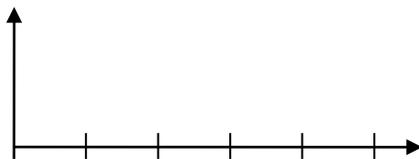
instants d'échantillonnage	0	T_e	$2T_e$	$3T_e$	$4T_e$	$5T_e$	$6T_e$	$7T_e$	etc
$x(n) = \delta(n)$	1	0	0	0	0	0	0	0	---
$y(n) = h(n)$									



A coté des filtres RIF, on trouve les filtres **RII** (**R**éponse **I**mpulsionnelle **I**nfinie, ou **IIR** en anglais pour **I**nfinite **I**mpulse **R**esponse). Dans ce cas un échantillon de sortie $y(n)$ dépend de tous les échantillons passés. L'algorithme $y(n) = \frac{y(n-1) + x(n)}{2}$ est un exemple de filtre RII. La relation $y(n) = \frac{y(n-1) + x(n)}{2}$ est une **relation de récurrence**, elle permet de calculer $y(n)$ connaissant $y(n-1)$. Ce type de filtre a un équivalent en analogique.

Question 3 : Déterminer les éléments $h(n)$ de la réponse impulsionnelle du filtre obéissant à l'équation de récurrence $y(n) = \frac{y(n-1) + x(n)}{2}$ et compléter le graphe ci-dessous.

instants d'échantillonnage	0	T_e	$2T_e$	$3T_e$	$4T_e$	$5T_e$	$6T_e$	$7T_e$	etc
$x(n) = \delta(n)$	1	0	0	0	0	0	0	0	---
$y(n) = h(n)$									



II- Virgules fixe et flottante

Il existe deux types de processeurs, les processeurs à **virgule fixe** (fixed point) et les processeurs à **virgule flottante** (floating point). Ces derniers ont une grande **dynamique**, ils permettent de manipuler des grands nombres au détriment de la **précision**, les premiers permettent de faire des calculs avec une grande précision mais une dynamique faible. C'est l'application qui détermine le type de processeur (par exemple : traitement audio ou TV, etc ...). Une opération de filtrage numérique consiste à faire des multiplications, des additions et des transferts en mémoire, on ne procède pas (perte de temps) à des tests d'overflow, il y a donc des risques de dépassement si on manipule des nombres dépassant la capacité des registres. Vous aurez l'occasion d'observer ce problème en TP car le DSP utilisé permet de travailler soit en virgule fixe soit en virgule flottante.

II-a/ Format envirgule fixe

Dans le format virgule fixe les nombres sont représentés en complément à deux.

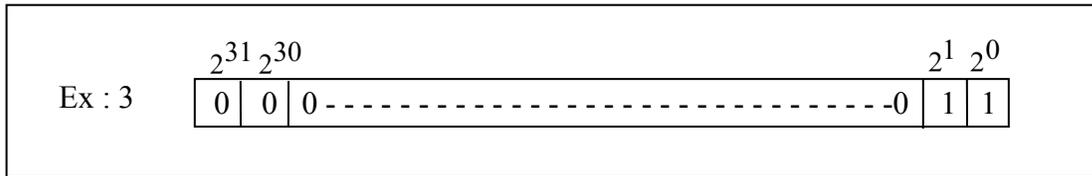


Figure 2 : Exemple de registre 32 bits en virgule fixe

Question 4 : On effectue des opérations sur des registres de 32 bits, 1) quels sont le plus grand nombre positif, le plus petit nombre positif et le plus petit nombre négatif manipulables.

II-b/ Format en virgule flottante

Dans le cas d'un registre de 32 bits (simple précision) par exemple, on réserve 1 bit pour le signe **s**, 8 bits pour l'exposant **e** et 23 bits pour la mantisse **f** (standard IEEE-754). Le nombre se lit ainsi :

$$(1)^s 2^{(e-127)} 1.f$$

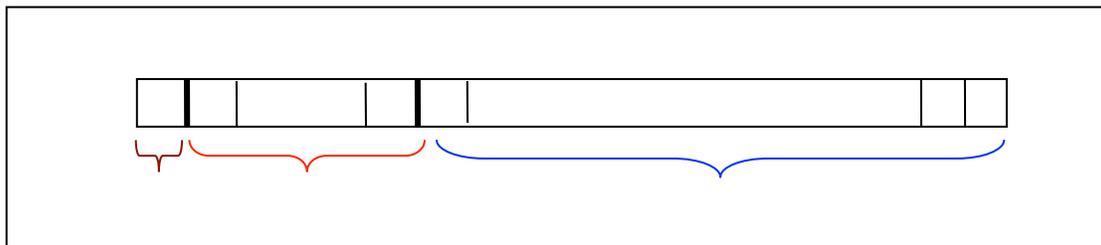


Figure 3 : Exemple de registre 32 bits en virgule flottante

Exemples :

Question 7 : Représenter le spectre du signal du signal $x(t)$ entre 0 et 15kHz. Il n'est pas nécessaire de calculer les amplitudes de chaque raie, faites juste un graphe à main levée avec des amplitudes approximatives, par contre indiquez les fréquences suivant l'axe horizontal.

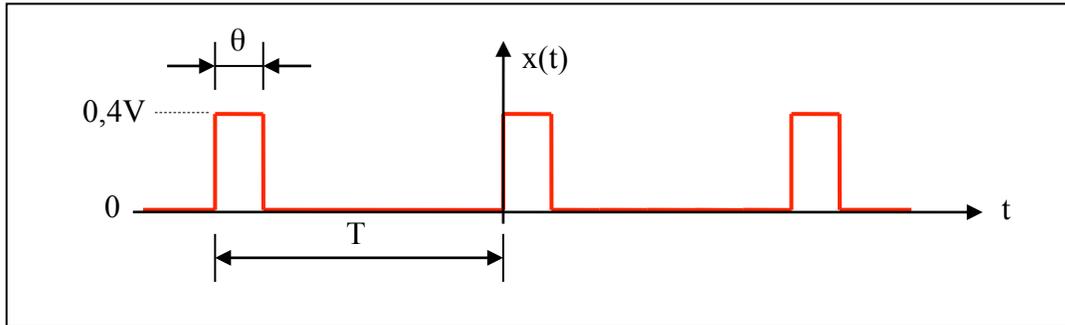


Figure 4 : Signal $x(t)$ appliqué au filtre numérique de la figure 1

Question 8 : On réalise 4 expériences avec deux filtres numériques passe bande de largeur de bande différente, avec ou sans filtre antirepliement. On demande pour les 4 cas de tracer le spectre du signal $y(t)$ obtenu. Le filtre antirepliement à capacités commutées coupe à 3353Hz quand il est inséré dans la chaîne de traitement.

Pour chaque cas, vous devez tracer : le spectre *i)* après le filtre antirepliement s'il y en a un, *ii)* après échantillonnage et *iii)* après filtrage et conclure.

- **Cas n°1 :**
 - Filtre numérique passe bande
 - Fréquence coupure basse = 2540Hz
 - Fréquence de coupure haute = 2560Hz
 - Avec filtre antirepliement
- **Cas n°2 :**
 - Filtre numérique passe bande
 - Fréquence coupure basse = 2540Hz
 - Fréquence de coupure haute = 2560Hz
 - Sans filtre antirepliement
- **Cas n°3 :**
 - Filtre numérique passe bande
 - Fréquence coupure basse = 2500Hz
 - Fréquence de coupure haute = 3100Hz
 - Avec filtre antirepliement
- **Cas n°4 :**
 - Filtre numérique passe bande
 - Fréquence coupure basse = 2500Hz

- Fréquence de coupure haute = 3100Hz
- Sans filtre antirepliement

IV- Comparaison des filtres RII et RIF : retard de groupe

Les filtres numériques RIF ont, sous certaines conditions, une **phase linéaire**, ce qui est impossible à réaliser avec un filtre analogique. Ecrivons alors la phase φ (en rd), voir figure 5, sous la forme $\varphi = -\tau\omega$ où ω (en rds^{-1}) est la pulsation et τ (en s), homogène à un temps, est le **retard de groupe**. Supposons maintenant un signal $x(t)$ constitué de plusieurs fréquences, par

exemple $x(t) = e(t) + u(t)$, avec $e(t) = \sum_{i=1}^M s_i \cos(\omega_i t)$ un signal basse fréquence et

$u(t) = \sum_{i=M+1}^{\infty} s_i \cos(\omega_i t)$ un signal haute fréquence. On cherche à conserver $e(t)$ et à éliminer $u(t)$,

pour cela on choisit un filtre RIF de fréquence de coupure F_c telle que $f_M < F_c < f_{M+1}$.

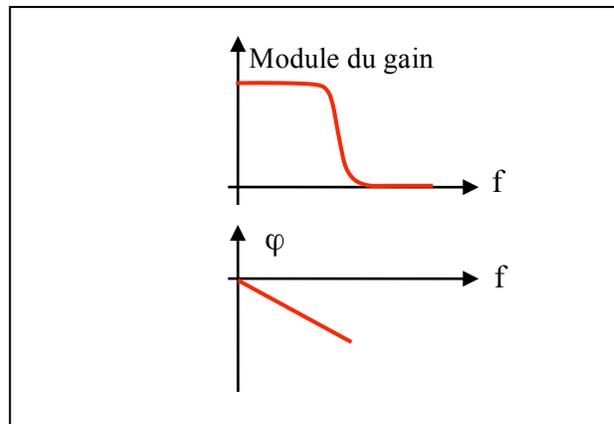


Figure 5 : Les filtres RIF ont, sous certaines conditions, une phase linéaire dans la bande passante

NB : Quand les filtres ne sont pas à phase linéaire (ex : RII et certains RIF), le retard de groupe τ dépend de ω , il est défini par : $\tau(\omega) = -\frac{d\varphi}{d\omega}$. Dans ce cas chaque fréquence composant le signal est retardée d'une quantité différente ce qui a pour effet de déformer le signal utile, on dit alors qu'il y a une **distorsion de phase**.

Question 9 : Montrer que le signal $y(t)$ obtenu après filtrage est égal à $e(t - \tau)$ si la phase du filtre est linéaire dans la bande passante F_c , autrement dit le signal utile est **retardé** mais non **déformé** comme il le serait avec un filtre RII de phase non linéaire.

NB : Dans un filtre RIF, l'échantillon $y(n)$ ne dépend que d'un nombre fini d'échantillons $x(n)$, l'équation (1) ne contient pas de coefficients a_j , seulement des b_i d'où $y(n) = \sum_{i=0}^{i=M} b_i x(n-i)$.

Ces filtres sont **toujours stables** contrairement aux filtres RII qui peuvent devenir instables. La phase linéaire des filtres RIF est un avantage, alors pourquoi des filtres RII ? L'inconvénient des filtres RIF est qu'il nécessite un nombre élevé de coefficients b_i pour obtenir une pente d'atténuation (en dB/décade) importante dans la zone de transition, c'est-à-dire entre la bande passante et la bande d'arrêt. Un nombre élevé de coefficients b_i est synonyme de mémoire importante et surtout de temps de calcul important. Pour faire du filtrage en temps réel, il faut que le temps de calcul soit inférieur à la période d'échantillonnage.

V- Génération d'un effet audionumérique : l'écho

Réaliser un écho simple d'un signal $x(t)$, c'est générer un signal $y(t) = x(t) + x(t - \alpha)$ où α est un retard. Un filtre RIF est parfaitement adapté pour cette réalisation. Aux échantillons $x(n)$ de $x(t)$, il faut donc faire correspondre des échantillons $y(n) = x(n) + x(n - k)$.

Question 10 : On veut simuler un écho correspondant à la situation de la figure 6 ; un micro distant de $d = 33m$ d'une paroi réfléchissante. La vitesse du son dans l'air est égale à $330ms^{-1}$, la fréquence d'échantillonnage $F_e = 8116,88Hz$, en déduire la valeur de k dans la relation $y(n) = x(n) + x(n - k)$. Les échantillons $x(n)$ sont numérisés sous 16bits, de quelle mémoire en kilo octets a-t-on besoin pour réaliser l'écho.

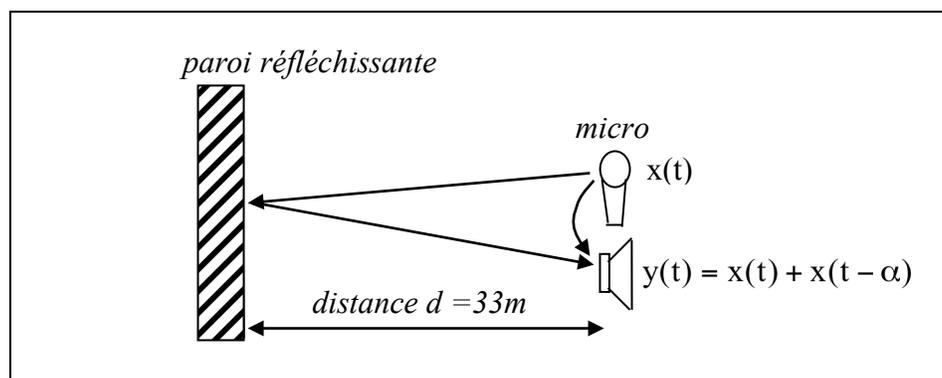


Figure 6 : Génération d'un écho par une paroi réfléchissante

NB : D'autres effets audionumériques peuvent être réalisés sur le même principe ; réverbération, chorus, flanger, phaser, ...

Thème 2 :

Filtrage numérique

I - Le matériel et les logiciels utilisés

Les expériences proposées sont articulées autour du Starter Kit TMS320C31 de Texas Instruments (www.ti.com). Deux générateurs de signaux fournissent les différents signaux nécessaires aux expériences proposées. Les signaux sont analysés à l'aide d'un oscilloscope numérique avec FFT (Fast Fourier Transform). Le synoptique de la maquette est donné à la figure 1.

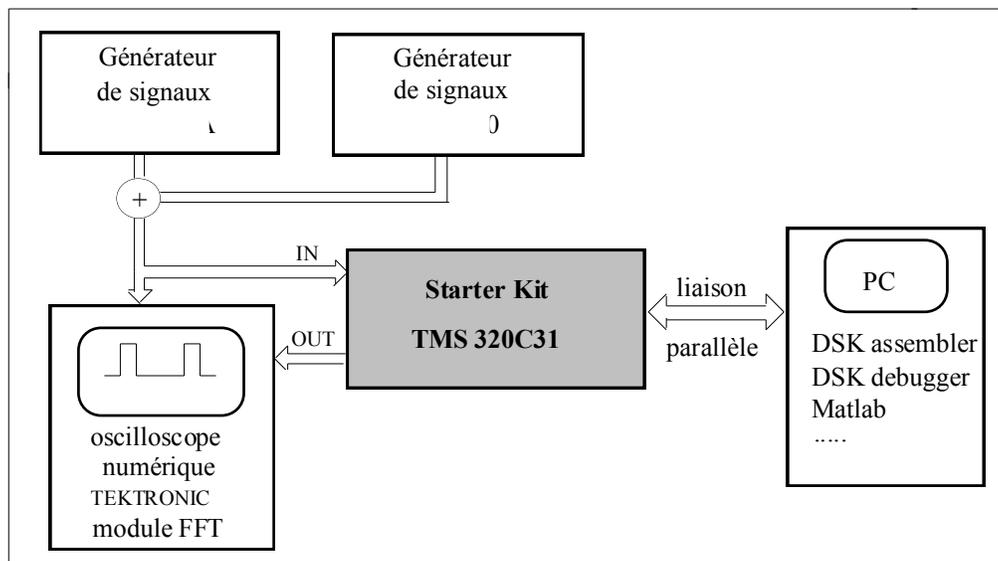


Figure 1 : Synoptique de la maquette

Au **DSP (Digital Signal Processor)** TMS320C31, il est associé le circuit d'interface **AIC (Analog Interface Circuits)** TLC32040C dont le schéma synoptique est donné en figure 2. La communication entre le DSP et l'AIC est assurée par les signaux \overline{FSR} , DR, DX et \overline{FSX} . Nous faisons ci-dessous une analyse succincte de l'AIC.

L'AIC gère 5 tâches :

- **amplification** du signal d'entrée, cela permet d'adapter l'amplitude du signal d'entrée à la pleine échelle du CAN
- **conversions** analogique/numérique (CAN) et numérique/analogique (CNA) sur 14 bits
- **filtrage anti-repliement (FAR) et de reconstruction (FR)**, ces filtres sont à capacités commutées avec des retards de groupe constant
- **transfert des données** entre l'AIC et le DSP

- **synthèse des différentes fréquences** : fréquences d'échantillonnage des convertisseurs (F_{can} et F_{cna}), fréquences d'échantillonnage des filtres à capacités commutées (F_1 et F_2), fréquence de transmission des données entre l'AIC et le DSP (F_{shift})

Les programmes sont chargés dans le processeur via la liaison parallèle sur LPT1. Les coefficients des filtres RII et RIF sont automatiquement calculés en utilisant la boîte à outils « Digital Signal Processing » de Matlab.

A titre d'exemple, on donne ci-dessous la ligne de code permettant de calculer par exemple les coefficients d'un filtre RII de type Butterworth d'ordre 2 et de fréquence de coupure $f_{-3dB} = 500\text{Hz}$ quand la fréquence d'échantillonnage $F_e = 4000\text{Hz}$.

>> [B,A]=butter(2,0.25) % le premier argument est l'ordre du filtre, le second est égal à $\frac{f_{-3dB}}{(F_e / 2)}$

$$\begin{aligned} B &= 0.0976 \quad 0.1953 \quad 0.0976 \\ A &= 1.0000 \quad -0.9428 \quad 0.3333 \end{aligned}$$

Les vecteurs B et A retournent les coefficients des polynômes en Z du numérateur et du dénominateur par ordre décroissant, d'où la transformée en Z du filtre suivante:

$$\frac{Y(z)}{X(z)} = \frac{0.0976 + 0.1953z^{-1} + 0.0976z^{-2}}{1 - 0.9428z^{-1} + 0.333z^{-2}}$$

La relation de récurrence s'écrit donc :

$$y(n) = 0.0976x(n) + 0.1953x(n-1) + 0.0976x(n-2) + 0.9428y(n-1) - 0.333y(n-2)$$

Il faut ensuite écrire un programme dans le langage du TMS320C31 et l'implémenter dans la mémoire du DSP. Dans le cas présent les programmes ont été écrits en assembleur, il existe bien entendu des outils de développement pour écrire les programmes en langage évolué mais ces outils coûtent chers.

Ce travail a fait l'objet de plusieurs stages de fin d'étude au département GEII (D. Arnaud 1996, P. Champion 1997 et plus récemment S. Glaizal 2006 qui a réécrit l'interface homme- machine sous Matlab. S. Glaizal, après une Licence Prof. SARI 2007, travaille actuellement chez Audemat à Mérignac spécialiste de télévision numérique (www.audemat.com))

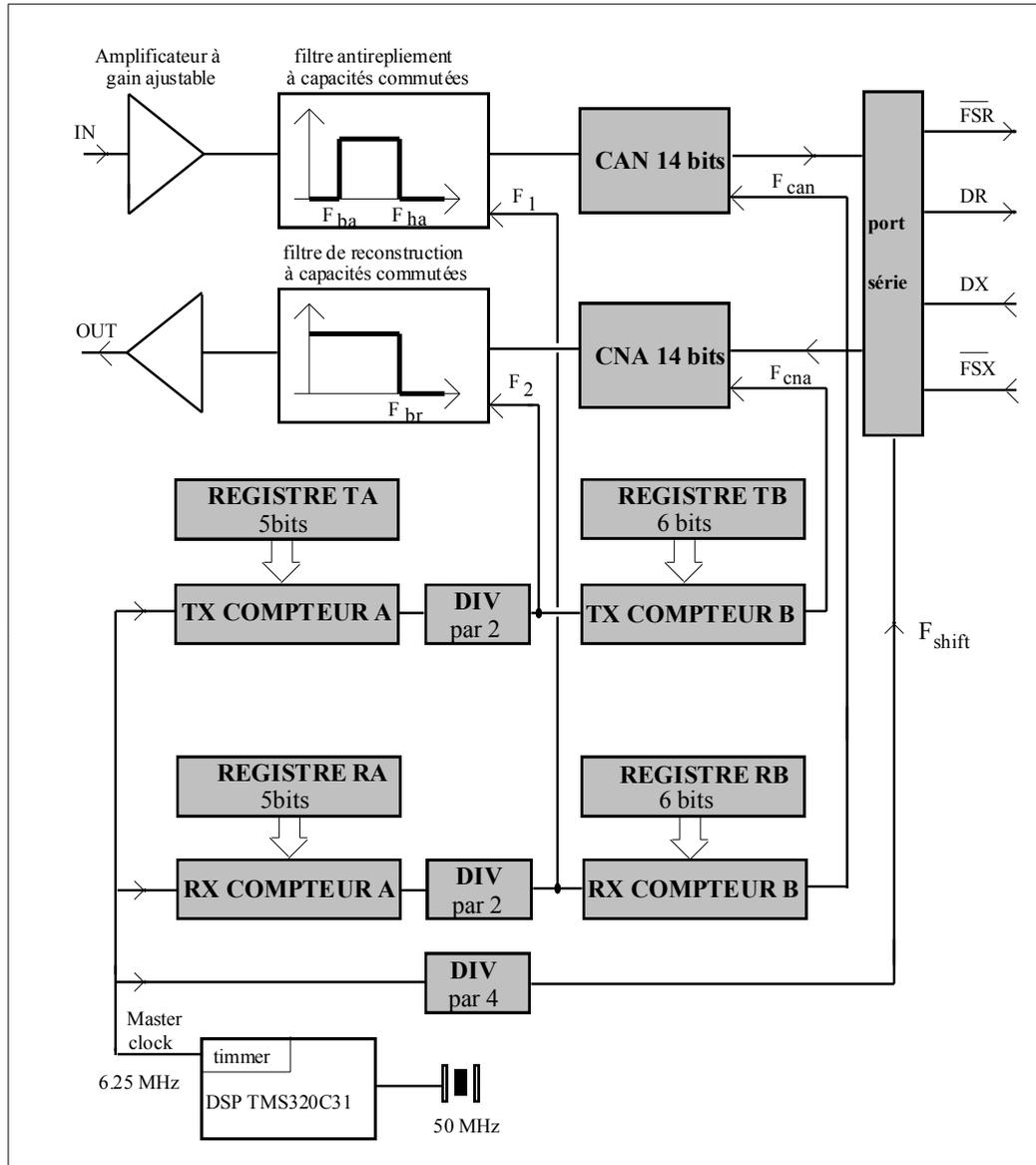


Figure 2 : Synoptique simplifié du circuit TLC32040C

II- Comparaison des performances des formats en virgule fixe et virgule flottante

Ouvrer Matlab et naviguer dans le menu déroulant.

1- Générer un filtre RII passe-bande du 4^{ème} ordre de type **Butterworth**, de fréquence de coupure basse $F_{cb}=1950$ Hz et de fréquence de coupure haute $F_{ch}=2050$ Hz. Choisir l'option avec **filtre antirepliement**.

a- Choisir l'option **virgule fixe**, entrer sur l'entrée IN un signal sinusoïdal de fréquence 2kHz de q.q. mV. Assurer vous qu'il s'agit bien d'un filtre passe bande. Augmenter progressivement l'amplitude.

Noter l'amplitude V_c à partir de laquelle le signal de sortie OUT devient "chaotique".

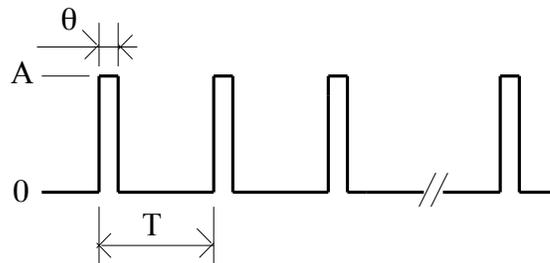
b- choisir l'option **virgule flottante**, faire la même expérience (ne pas dépasser la pleine échelle du convertisseur c'est à dire 3V crête). Que constatez vous ? Expliquer la différence de comportement.

c- En virgule fixe, le « chaos » apparaît pour une tension d'autant plus faible que la largeur de bande est étroite. En effet, une bande passante étroite est synonyme de coefficient de surtension élevé et donc de manipulation de grands nombres.

Reprendre les deux manipulations précédentes en fixant une fréquence de coupure basse $F_{cb} = 1980$ Hz et une fréquence de coupure haute $F_{ch} = 2020$ Hz. Pour quelle nouvelle valeur de tension, le signal devient-il « chaotique » ?

III- Rôle du filtre antirepliement sur la forme du signal de sortie

Configurer le générateur de signaux pour obtenir le signal périodique $s(t)$ représenté ci-dessous, prendre : $1/T = 2,55$ kHz, $T/\theta = 5$ et $A = 0.4$ V.



2- Observer, avec l'outil FFT de l'oscilloscope, le spectre du signal en sortie du générateur. Vérifier l'accord avec les valeurs théoriques du pic à $f=0$ et à $f_0, 2f_0$ puis $3f_0$.

Pour rappel : Le signal $s(t)$ est périodique et admet comme développement en série de Fourier :

$$s(t) = \frac{A\theta}{T} + \frac{2A\theta}{T} \sum_{n=1}^{\infty} \frac{\sin\left(\frac{n\pi\theta}{T}\right)}{\left(\frac{n\pi\theta}{T}\right)} \cos(n\omega t) \text{ avec } \omega = \frac{2\pi}{T}$$

3- Pour récupérer le fondamental du signal, on utilise un filtre RII passe-bande du 4^{ème} ordre de type **Butterworth**, de fréquence de coupure basse $F_{cb} = 2540$ Hz et de fréquence de coupure haute $F_{ch} = 2560$ Hz. On utilise le processeur en **virgule flottante**. Observez le signal de sortie dans les deux cas suivants :

- avec filtre antirepliement
- sans filtre antirepliement

Expliquer pourquoi les signaux sont identiques dans les deux cas.

Vérifier que le système réalise bien un filtre passe-bande en faisant varier la fréquence du générateur, noter les deux fréquences de coupure à -3dB.

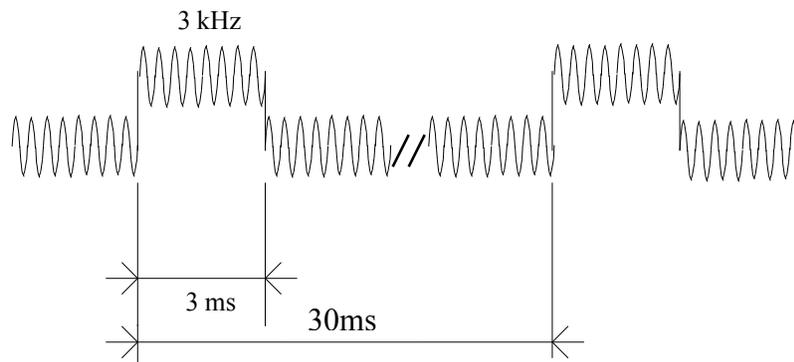
4- Faire la même expérience avec un filtre RII passe-bande du 4^{ème} ordre de type **Butterworth**, de fréquence de coupure basse $F_{cb} = 2500$ Hz et de fréquence de coupure haute $F_{ch} = 3100$ Hz. On utilise le processeur en **virgule flottante**. Observez le signal de sortie dans les deux cas suivants :

- avec filtre antirepliement
- sans filtre antirepliement

En l'absence de filtre antirepliement, observer le spectre du signal de sortie avec l'outil FFT et expliquer la présence des deux raies observées. Justifier votre résultat par rapport à ceux du TD correspondant.

IV- Comparaison des filtres RII et RIF

A l'aide des générateurs de signaux et d'un additionneur à résistances, générer un signal rectangulaire additionné d'un signal sinusoïdal de fréquence 3kHz comme ci-dessous :



On se propose de supprimer la composante de fréquence à 3 kHz à l'aide d'un filtre passe-bas. Le défi consiste donc à éliminer le signal à 3 kHz en modifiant le moins possible la forme du pulse.

5- Générer un filtre RII de type Butterworth passe-bas du huitième ordre avec une fréquence de coupure à -3dB de 2 kHz en utilisant le processeur en **virgule flottante** et **sans filtre antirepliement**.

NB : Pourquoi sans filtre antirepliement ?

En fait, si on l'utilise, on perd une grande partie du spectre utile.

L'absence de filtre antirepliement, génère du repliement mais compte tenu de la fréquence d'échantillonnage égale à 8116,88Hz $\gg \frac{1}{30 \times 10^{-3}} = 33\text{Hz}$, le repliement peut être considéré

comme faible.

Observer les signaux IN DSP et OUT DSP en synchronisme et expliquer la forme obtenue du signal de sortie.

6- Générer maintenant un filtre RIF à phase linéaire passe-bas avec 150 coefficients et une fréquence de coupure à 2kHz, en utilisant le processeur en **virgule flottante** et sans filtre antirepliement.

A partir de la courbe de phase du filtre donnée par Matlab, déterminer le **retard de groupe** τ du filtre numérique ainsi généré.

Observer les signaux IN DSP et OUT DSP en synchronisme, expliquer la forme du signal de sortie et mesurer le retard de OUT DSP. Le retard correspond à la somme du retard de groupe τ et du retard de groupe du filtre de reconstruction soit environ égal à 1ms. Pour information, le filtre antirepliement est aussi à retard de groupe constant, environ égal à 1,2ms.

7- Conclure quant à l'intérêt des filtres RIF. Justifier votre réponse en vous appuyant sur les résultats du TD.

V- Génération d'un effet audionumérique : l'écho

Commuter l'interrupteur d'entrée de la carte du DSP pour que le signal d'entrée soit celui du **microphone à électret** puis brancher en sortie du DSP l'**amplificateur de puissance classe D** chargé par le haut-parleur.

Insérer un **atténuateur de 10dB** entre la sortie du DSP et l'entrée de l'amplificateur de puissance (voir figure 3).

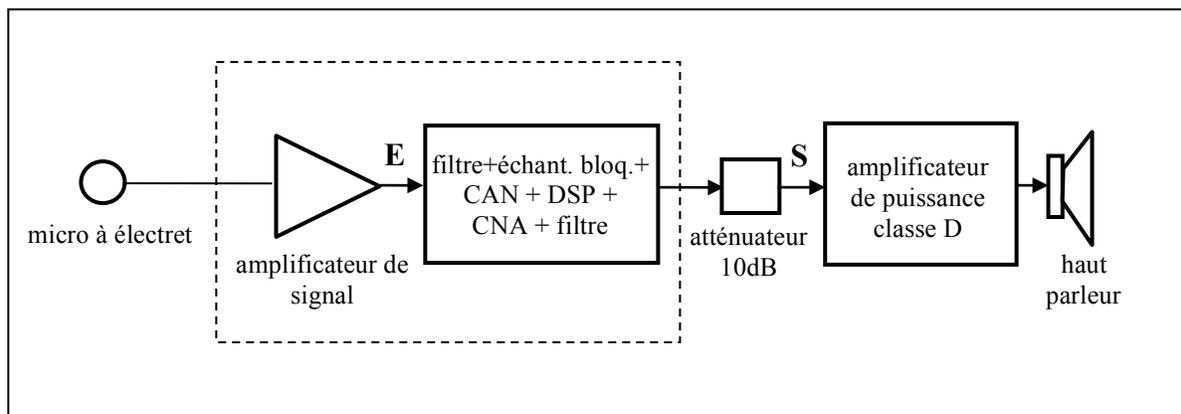


Figure 3 : Chaîne de traitement numérique audio

8- Dans le menu, choisissez « écho ». Parler devant le micro en prononçant une lettre de l'alphabet et constater l'écho. Observer avec l'oscilloscope les signaux E et S, mesurer le temps de retard. L'écho généré simule une paroi réfléchissante située à quelle distance du micro (voir figure 4) ?

9- Enregistrer le signal E de quelques lettres de l'alphabet ou de quelques mots (droite, gauche, ...).

Est-ce qu'une simple analyse temporelle permet de faire de la reconnaissance vocale d'après vous ? Faites maintenant une analyse spectrale, avec l'outil FFT de l'oscilloscope, du signal E correspondant à quelques lettres de l'alphabet ou quelques mots. Est-ce qu'une simple analyse spectrale permet de faire de la reconnaissance vocale d'après vous ? Conclusion.

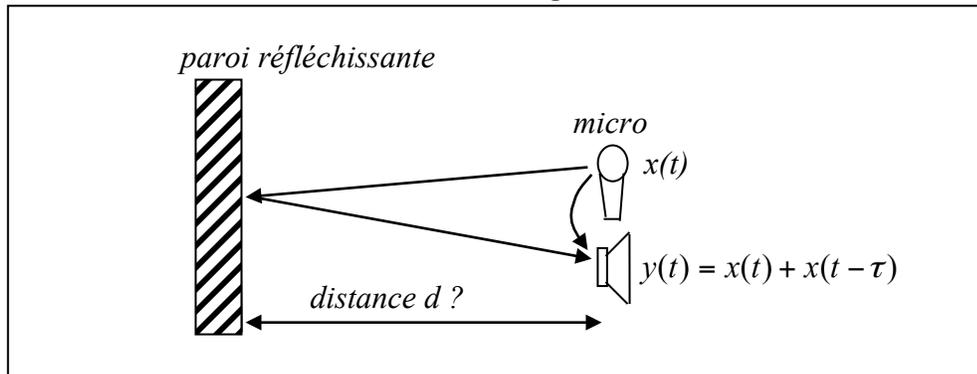


Figure 4 : Simulation d'un écho

Référence :

Pour en savoir plus sur les amplificateurs classe D, consulter l'article « Amplificateurs audio fréquence, classe A, AB et D » par G. Couturier sur le site de la revue pédagogique J3ea (<http://www.j3ea.org>)

