Université Henri Poincaré - Nancy 1
IUT de Nancy-Brabois
Département GENIE des TELECOMMUNICATIONS et RESEAUX



2ème Année

TRAITEMENT du SIGNAL

TP FILTRAGE NUMERIQUE

Programme pour 2003-2004

Intervenants : Jean-Marc THIRIET, Hugues GARNIER

SOMMAIRE

- *APPLICATION DE MATLAB A L'ETUDE DES FILTRES NUMERIQUES*
- *NOTICE INFORMATIQUE*
- *INTERFACE GRAPHIQUE POUR LES DEVELOPPEMENTS SUR DSP*

TP1 : ANALYSE ET SYNTHESE DE FILTRES NUMERIQUES NON RECURSIFS

TP2 : UTILISATION DE PROCESSEURS DE TRAITEMENT NUMERIQUE DU SIGNAL

TP3 : ANALYSE ET SYNTHÈSE DE FILTRES NUMERIQUES RECURSIFS

APPLICATION DE MATLAB À L'ÉTUDE DES FILTRES NUMERIQUES

1. Représentation d'un filtre numérique

Un filtre numérique en général est représenté par sa fonction de transfert en z. Sous MATLAB, une fonction de transfert en z est caractérisée par deux vecteurs contenant les coefficients des puissances décroissantes de z^{-1}, du numérateur et du dénominateur de la fonction de transfert (FT) considérée.

Ex : La fonction de transfert en z suivante :

$$H(z) = \frac{1}{2} + \frac{1}{2}z^{-1}$$

sera représentée sous Matlab par :

```
»num=[0.5 0.5];
»den=[1 0];
```

2. Réponses temporelles et fréquentielles

La réponse à un échelon peut être obtenue à l'aide de la commande **`dstep :`**

```
»num=[0.5 0.5];
»den=[1 0];
»y=dstep(num,den) ;
»stem(y)
```

La fonction **`stem`** permet le tracé d'une séquence de valeurs discrète.

Dans le même esprit, la réponse impulsionnelle est tracée à l'aide de la commande **`impz(num,den).`**

On peut également, grâce à la fonction **`dlsim`**, visualiser la réponse temporelle d'un système à une entrée quelconque.

Pour obtenir la réponse en fréquence d'un filtre numérique, on utilisera la commande **`freqz`** dont la syntaxe est **`[H,w]=freqz(num,den,N, fe)`** où N est le nombre de points de calcul de la réponse fréquentielle entre [0 ; $f_e/2$]. La commande **`freqz(num,den,N,fe)`** permet de tracer le diagramme de Bode pour la zone utile du filtre (entre 0 et $f_e/2$) ; N peut être choisi égal à 512.

Le diagramme de Bode peut également être obtenu à l'aide de la commande **`dbode(num,den,Te).`** La commande **`[mag,phase]=dbode(num,den,Te,w)`** permet d'obtenir l'amplitude et la phase du système pour les pulsations contenues dans le vecteur w. On peut choisir une échelle logarithmique de fréquences par la fonction **`w=logspace(d1,d2,N)`**.

Autres exemples typiques :

Réponse impulsionnelle

Créer un fichier texte **`repimpz.m`** (*Menu File+ New M-File*) contenant les lignes suivantes :

```
[y,t]=impz(num,den,N);
stem(t,y)
```

Sauvegarder ce fichier (*Menu File+ Save as*) sous le nom **`repimpz.m`** sous votre répertoire. Dans la fenêtre *command*, exécuter les lignes suivantes :

```
»N=10
»num=[0.5 0.5];
»den=[1 0];
»repimpz
```

Réponse indicielle

De la même manière, on aurait pu créer **`repindz.m`** pour la réponse indicielle :

```
y=dstep(num,den,N);
t=0:N-1;
stem(t,y)
```

Tracé d'un diagramme de Bode

```
»num=[0.5 0.5];
»den=[1 0];
»w=logspace(-1,1);
»[amp,phase]=dbode(num,den,w);
```

```
»semilogx(w,phase) %permet l'affichage d'une courbe avec une échelle
logarithmique en abscisse et une échelle linéaire en ordonnée.
»loglog(w,amp) %permet l'affichage d'une courbe avec échelles
logarithmiques en abscisse et ordonnée.
```

On aurait pu également définir une variable :

```
»amplog = 20*log10(amp)
»semilogx(w,amplog)
```

On aurait pu également utiliser la commande `freqz` :

```
»num=[0.5 0.5];
»den=[1 0];
»fe=1 ;
»[H,f]=freqz(num,den,512,fe);
»amp=abs(H);
»phase=angle(H);
»figure(1)
»plot(f,amp),grid
»title('Réponse fréquentielle en amplitude pour
[0;fe/2]'),xlabel('fréquence (Hz)')
figure(2)
»subplot(2,1,1)
»semilogx(f,20*log10(amp)),grid
»title('Diagramme de Bode en amplitude'),
»ylabel('Amplitude en dB')
»subplot(2,1,2)
»semilogx(f,phase*180/(2*pi)),grid
»title('Diagramme de Bode en phase'),
»ylabel('Phase en °'),xlabel('fréquence (Hz)')
```

AVANT DE COMMENCER !

Vous trouverez ci-dessous la liste des logiciels et fichiers nécessaires pour les TP DSP et leur localisation sur les disques durs.

- Avant toute chose,
  - Vérifier que vous avez bien les connexions réseau sur votre machine et les disques configurés de la manière suivante :
    - H pour votre login, I pour le pub des étudiants, J pour program ; si ce n'est pas le cas, ouvrer une fenêtre DOS et taper les commandes suivantes :
      - `net use h: \\s2000-gtr\`*`nom_login_etudiant`*`$`
      - `net use i: \\s2000-gtr\pub_etudiants`
      - `net use j: \\s2000-gtr\program`
  - aller dans le répertoire "pub" des étudiants dans (disque i) et téléchargez le répertoire MATLAB2004.

S'il y a des problèmes, l'organisation des disques est la suivante :

Logiciel ou fichiers	Comment les utiliser	Comment vous dépanner
MATLAB	S'utilise directement à partir de l'icône MATLAB. Une fois dans Matlab, faire cd j: *votre_rep_tds* Pour que Matlab trouve les fichiers dans votre répertoire de travail	Matlab se trouve dans le répertoire Program\Matlab\bin\matlab.exe sur la machine s2000-gtr ou j:\Matlab\bin\matlab.exe
Kit de développement DSP (assembleur, débogueur, chargeur ("loader") A UTILISER DANS UNE FENETRE MS-DOS ou depuis l'interface graphique (cf. page suivante)	Depuis votre répertoire de travail (dans une fenêtre MS-DOS), l'assembleur (dsk3a ou dsk5a), le débogueur (dsk3d ou dsk5d) et le chargeur (dsk3l ou dsk5l) sont accessibles directement	Le kit de développement se trouve dans le répertoire Program\DSP\DSKC30\ ou Program\DSP\DSKC50\ sur la machine nt_gtr (ou depuis j:) Si le kit de développement est inaccessible, il faut ajouter les logiciels dans le chemin de DOS, pour cela vous pouvez exécuter envc30 ou envc50 (ces deux fichiers se trouvent dans \TDS\DSP dans le répertoire pub des étudiants).

1. ANALYSE DE FILTRES NUMERIQUES NON RECURSIFS (OU RIF)

Lorsque vous lancez Matlab, vérifiez que vous êtes dans votre répertoire, la commande `cd` permet de savoir dans quel répertoire on se trouve et la commande `cd h:`, par exemple permet de changer de disque.

1. ANALYSE DE FILTRES NON RECURSIFS

1.1. ETUDE THÉORIQUE

Pour chacun des algorithmes suivants :

Algorithme n°0 : $s_n = e_n$

Algorithme n°1 : $s_n = \frac{1}{2}e_n + \frac{1}{2}e_{n-1}$

Algorithme n°2 : $s_n = -\frac{1}{2}e_n + \frac{1}{2}e_{n-1}$

Algorithme n°3 : $s_n = -\frac{1}{2}e_n + \frac{1}{2}e_{n-2}$

Algorithme n°4 : $s_n = \frac{1}{2}e_n + \frac{1}{2}e_{n-2}$

Algorithme n°5 : $s_n = \frac{1}{4}e_n + \frac{1}{2}e_{n-1} + \frac{1}{4}e_{n-2}$

Algorithme n°6 : $s_n = -\frac{1}{2}e_n + \frac{1}{2}e_{n-4}$

Déterminer (tracer les réponses temporelles et fréquentielle) :

a) le type de filtre (RIF, RII) et son ordre,

b) la fonction de transfert en z,

c) la réponse impulsionnelle,

d) la réponse indicielle,

e) la réponse fréquentielle, en précisant la (ou les) fréquence(s) de coupure et la zone utile. Il faudra par conséquent calculer cette fréquence de coupure.

f) le type du filtre (passe-bas, passe-haut, passe-bande, réjecteur,…).

Comparer les filtres passe-bas obtenus entre eux (calculer la ou les fréquence(s) de coupure, observe la raideur des filtres sur la réponse fréquentielle, relier la raideur à l'ordre, ...)

1.2. ANALYSE EN SIMULATION SOUS MATLAB

Pour chacun des filtres étudiés dans la partie étude théorique, les réponses impulsionnelle, indicielle et fréquentielle sont visualisables en tapant : `tp1`.

Pour visualiser les comportements temporels et fréquentiels de filtres quelconques, vous pourrez utiliser la commande `tprep`.

2. SYNTHESE DE FILTRES NON RECURSIFS PAR LA METHODE DE LA FENETRE

2.1 SYNTHÈSE EN SIMULATION SOUS MATLAB

On s'intéresse dans cette partie à la synthèse de filtres non récursifs par la méthode de la fenêtre à l'aide de la fonction Matlab *FIR1*. On considérera la conception d'un FNR passe-bas, passe-haut, passe-bande ou coupe-bande.

fir1 Synthèse de filtre à RIF par la méthode de la fenêtre

num = fir1(ordre,fn) permet la conception d'un filtre **passe-bas** à RIF d'ordre spécifié. Les coefficients du filtre sont retournés dans un tableau num de longueur ordre+1. La fréquence de coupure fn doit être réduite entre 0 et 1, le 1 correspondant à la moitié de la fréquence d'échantillonnage.
Si fn est un tableau de deux éléments, fn = [fn1 fn2], fir1 donne les coefficients d'un **passe-bande** avec fn1 la fréquence mini et fn2 la fréquence maxi de la bande passante.
num = fir1(ordre,fn,'high') permet de synthétiser un filtre **passe-haut**.
num = fir1(ordre,fn,'stop') permet de synthétiser un **réjecteur de bande** avec fn = [fn1 fn2].
Pour les passe-haut et réjecteur de bande, l'ordre doit être pair, donc le nombre de coefficients du filtre et de la fenêtre associée sont impairs.
La procédure fir1 utilise par défaut une fenêtre de Hamming. Toutefois, d'autres fenêtres sont disponibles : rectangulaire (Boxcar), Hanning, Bartlett, Blackman, Kaiser et Chebwin en indiquant un argument optionnel. Par exemple,
num = fir1(ordre,fn,bartlett(ordre+1)) pour la fenêtre de Bartlett. ou ***num = fir1(ordre,fn,'high',bartlett(ordre+1))***
num = fir1(ordre,fn,chebwin(ordre+1,Rr) pour la fenêtre de Chebyshev.

On peut ensuite refaire l'analyse de ces filtres pour étudier les diverses réponses et comparer avec un filtre analogique type afin de vérifier la qualité du filtre numérique trouvé. En pratique, on testera principalement la réponse fréquentielle (fichier `tprep`).

Dans certains cas, on peut avoir des problèmes avec le gain statique. Pour se ramener à un gain statique de 1, il faut que le gain pour p=0, autrement dit pour z=1 soit égal à 1, par conséquent il faudra que :

$$\left[\sum_i b_i . z^{-i}\right]_{z=1} = \sum_i b_i = 1$$

Il suffira, si nécessaire, de multiplier tous les coefficients par une constante, ce qui ne changera pas les caractéristiques fréquentielles du filtre.

2.2 TRAITEMENT DU SON PAR FILTRE NUMÉRIQUE

2.2.1 Extraction d'un accord de do majeur grave

Soit un son composé de la superposition d'un accord de do majeur grave (do=262 Hz, mi =330 Hz; sol=392 Hz) auquel est superposé un accord de do majeur plus aigu (do=524 Hz, mi =660 Hz; sol=784 Hz). L'ensemble est échantillonné à 2000 Hz.

Il est demandé de concevoir par la méthode de la fenêtre un filtre numérique non récursif permettant de supprimer le do majeur aigu. Quelques indications sont données dans le fichier sounr1.m que l'on pourra compléter, la fonction son_ft.m pourra également être utilisée. (Avant toute chose, veuillez copier dans votre répertoire de travail les fichiers sounr1.m, sounr2.m et son_ft.m, ces fichiers se trouvent dans le répertoire pub des étudiants).

2.2.2. Elimination du mi

Dans l'accord résultant (do majeur grave), on souhaite maintenant éliminer le mi central et ne conserver que le do et le sol. Il est demandé là encore de concevoir un filtre numérique non récursif permettant d'assurer cette fonctionnalité. Quelques indications sont données dans sounr2.m que l'on pourra compléter.

2.3 CONCEPTION DE FILTRES NUMÉRIQUES

PREPARATION

2.3.1. Concevoir un FNR passe-bas obéissant aux caractéristiques ci-dessous :
fréquence de coupure 1.875 kHz
zone de transition 625 Hz

ondulations en bande atténuée : > 51 dB
fréquence d'échantillonnage : 8 kHz

Préciser quels sont les types de fenêtres utilisables et les caractéristiques du filtre (ordre...), choisir une fenêtre et justifier le choix. Calculer à la main les deux premiers coefficients $h_N(0)$ et $h_N(1)$ du filtre.

2.3.2. Concevoir un FNR passe-bande obéissant aux caractéristiques ci-dessous :
bande passante 1500-2500 Hz
zone de transition 500 Hz
ondulations en bande passante : 0.1 dB
ondulations en bande atténuée : > 60 dB
fréquence d'échantillonnage : 10 kHz

Préciser quels sont les types de fenêtres utilisables et leurs caractéristiques.

2.3.3. Concevoir un FNR passe-haut obéissant aux caractéristiques ci-dessous :
fréquence de coupure idéale : 2400 Hz
zone de transition 1000 Hz
ondulations en bande passante : 0.01 dB
ondulations en bande atténuée : > 40 dB
fréquence d'échantillonnage : 20 kHz

On utilisera ici la fenêtre de Kaiser, préciser l'ordre et la valeur de β.

2.3.4. Concevoir un filtre réjecteur de bande ayant les caractéristiques suivantes :
zone de transition 0.04 (fréquence normalisée)
ondulations en bande atténuée : > 40 dB
fréquence de coupure : 0.2 (fréquence normalisée)

N.B. : les fréquences normalisées sont rapportées à la demi fréquence d'échantillonnage (fe/2), appelée également la fréquence de Nyquist.

SIMULATION

Sous Matlab, on pourra procéder de la manière suivante :

```
>>num=fir1(54,.23,'high',hamming(55)); %pour calculer les coefficients du filtre
>>tprep %pour visualiser l'ensemble des réponses temporelles et fréquentielles de ce
filtre
```

2.3.5. Choisissez les résultats obtenus avec une fenêtre de Hamming pour le filtre analysé en 2.3.1 et simuler sa réponse fréquentielle en tapant `tp2` dans la fenêtre *command* de Matlab. Vérifier que les spécifications demandées sont obtenues (fréquence de coupure, ondulations en bande passante, largeur de la zone de transition...).

2.3.6. Refaire les manipulations pour le filtre préparé en 2.3.2 pour les résultats obtenus pour une fenêtre de Blackman.

2.3.7. Refaire les manipulations pour le filtre préparé en 2.3.3 pour les résultats obtenus pour une fenêtre de Kaiser.

2.3.8. Refaire les manipulations pour le filtre préparé en 2.3.4 pour les résultats obtenus pour une fenêtre de Hanning.

2. UTILISATION DE PROCESSEURS DE TRAITEMENT NUMERIQUE DU SIGNAL

1. DSP : COMPOSANTS MIRACLES ?

1.1 CARACTÉRISTIQUES ET PRINCIPAUX CONSTRUCTEURS

Des modems aux systèmes de freinage antibloquant (ABS) en passant par les récepteurs TV (satellite et bientôt terrestres), de plus en plus de produits incorporent des processeurs de traitement du signal, les fameux DSP ou Digital Signal Processor. Par exemple, le traitement du signal, autrefois mis en œuvre dans des applications « haut de gamme » relativement coûteuses (aéronautique, aérospatiale, ...) s'est démocratisé et continue à se démocratiser à l'image des micro-ordinateurs et ceci grâce au développement des DSP.

Les DSP sont caractérisés par leur puissance et leur rapidité de calcul. Le marché des DSP devrait dépasser les 8 milliards de dollars en l'an 2000 selon une étude américaine. Ce marché se répartit principalement entre :

- Texas Instruments (47%),
- Lucent Technologies (35%),
- Motorola (27%),
- Analog Devices (8%).

En 1997, la seule société Texas Instruments a vendu quelques 50 millions de solutions DSP rien que pour les radio-mobiles ...

1.2 DSP : PRINCIPALES APPLICATIONS

Les DSP trouvent des applications dans les domaines en pleine expansion tels que les communications sans fil, les modems ultra-rapides ou la gestion des réseaux ; mais ils sont utilisés également dans des secteurs ou pour des usages très variés : commande de moteurs, suppression de bruit dans les voitures, égalisation des sons, imagerie médicale, compression audio-vidéo.

Les systèmes de communication numériques reposent sur le traitement du signal à grande vitesse pour la compression, l'encodage, le filtrage, ... Les DSP fournissent en effet la puissance de calcul nécessaire à ces systèmes et ce, à un coût assez bas pour que ces communications puissent être largement répandues.

Dans l'automobile, les DSP trouvent des applications dans les systèmes électroniques de direction assistée, dans les systèmes de suspension, les systèmes de freinage antibloquant (ABS) et antidérapant, la régulation de vitesse avec système d'anticollision par radar...

Dans le domaine des ordinateurs, les DSP servent au niveau des disques durs où ils ont contribué aux gains massifs de capacité de stockage, grâce au contrôle extrêmement précis, que permettent les DSP, au niveau du positionnement des têtes de lecture/écriture.

Dans le domaine des télécommunications, les DSP occupent un rôle important et équipent les stations de base sans fil, les systèmes de vidéoconférences, les standards téléphoniques, mais aussi les récepteurs d'appel de poche, les répondeurs numériques, les téléphones multifonctions...

Au niveau industriel, les DSP sont présents dans les compresseurs et pompes à chaleur, les commandes numériques pour moteurs, pour robots...

En ce qui concerne les équipements de loisirs, les DSP servent dans les récepteurs TV numériques, les appareils photo et camescope numérique, les jeux vidéo, les guitares électroniques...

Côté domotique enfin, signalons que dans certaines machines à laver, un système à base de DSP gère la vitesse du moteur pour réduire le bruit et mieux gérer l'énergie.

2 ENVIRONNEMENTS MATERIEL ET LOGICIEL

2.1 PRÉSENTATION MATÉRIELLE DES DSP

Les deux types de processeurs de traitement numérique du signal sont soit des **TMS320C50** (à virgule fixe), ou des **TMS320C31** (à virgule flottante), de chez Texas Instruments. Ces DSP sont montés sur une carte (starter kit) contenant une alimentation électrique (un transformateur en-dessous de la carte dans le boîtier). Cette carte intègre des convertisseurs analogique-numérique (CAN) et numérique-analogique (CNA) (TLC32040CFN) et une horloge interne. Le circuit d'interface analogique comprend le CAN précédé par un filtre anti-repliement et le CNA suivi par un filtre passe-bas de reconstruction.

Les signaux que l'on peut entrer et récupérer de la carte sont par conséquent des signaux analogiques ; les connecteurs utilisés sont de type BNC. Les signaux que l'on entre doivent être compris entre **-5 et 5V**. Les CAN et CNA sont des convertisseurs 14 bits.

La carte DSP est reliée au PC via une liaison RS232 pour le C50 ou l'interface parallèle pour le C31. Les programmes sont développés avec un éditeur ASCII puis assemblés sur le PC. Une fois assemblé, un programme peut être téléchargé via la ligne RS232, ou l'interface parallèle selon le cas, pour être utilisé.

La fréquence d'échantillonnage peut se régler (cf. page b-12 du petit manuel et dernière partie de ce TP). La fréquence d'échantillonnage maximale est de 19.2 kéch/s ; sa valeur par défaut est d'environ 10 Khz.

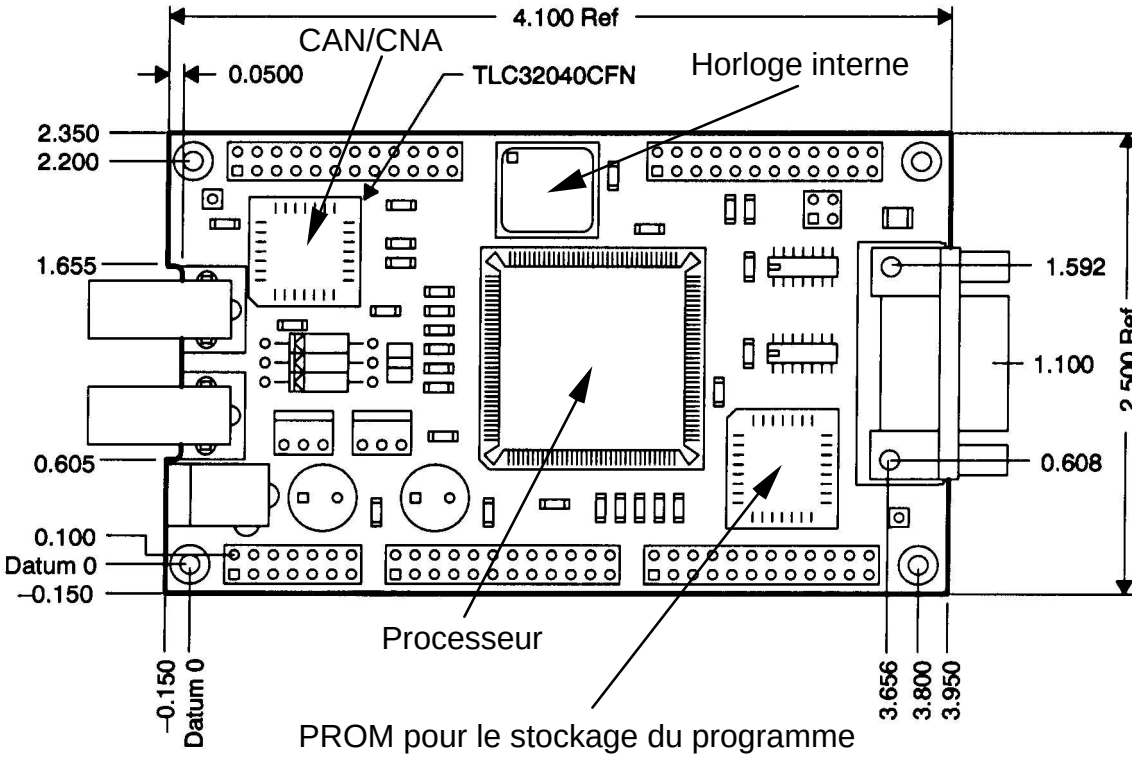


Schéma du DSP Starter Kit avec le C50

Le matériel nécessaire pour pouvoir utiliser le DSP durant ce TP est le suivant :

- 1 DSP,
- 1 GBF,
- 1 PC avec Matlab et le kit de développement DSP
- 1 câble RS232 pour les C50 ou parallèle pour les C31,
- 1 « T » pour câbles coaxial,
- 3 câbles coaxiaux.

2.2 PRÉSENTATION LOGICIELLE DES DSP TMS320

Avant de commencer, téléchargez le contenu du répertoire correspondant à votre type de DSP et son contenu dans votre répertoire de travail (`TP2004C3` pour le C30) et (`TP2004C5` pour le C50).

Le processeur peut être programmé en assembleur spécifique. Pour cela, sont disponibles trois utilitaires directement accessibles sous DOS (ou pourront être appelés via l'interface graphique, cf. notice informatique) :

- ***DSK5A*** (DSK3A pour le C31) permet d'assembler un source *.ASM en un source compilé pour le TMS320 *.DSK (de manière à ne pas avoir de problème, il faudra utiliser la **version 1.02** de l'assembleur sur C50 ; sur **C31**, il est impératif de copier le fichier `C3XMMRS.ASM` dans votre répertoire de travail).
  Ex : `DSK5A TOTO` ; assemblage de `TOTO.ASM`

- ***DSK5L*** (DSK3L pour le C31) est un "loader" (chargeur) qui permet de télécharger un programme assemblé dans le processeur afin de l'utiliser. Le chargeur charge le fichier `*.DSK`, qui est un exécutable pour le DSP, dans la mémoire du DSP puis demande l'exécution de celui-ci.
  Ex : `DSK5L TOTO` ; chargement et exécution de `TOTO.DSK`
  Autre exemple : `DSK3L TOTO LPT3` (à tester, selon le PC...)

- ***DSK5D*** (DSK3D pour le C31) est un "debugger" qui permet de visualiser le contenu des registres et de la mémoire du processeur pendant l'exécution d'un programme. Il permet, comme son nom l'indique, de débugger les programmes assemblés.

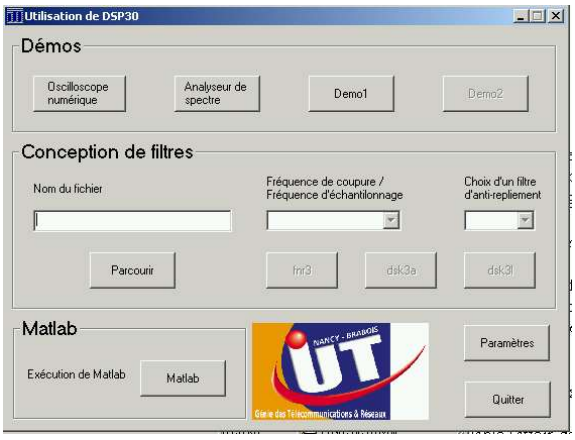
En effet, son utilisation permet plusieurs modes :
- le mode exécution,
- le mode pas à pas,
- le placement de points d'arrêts,
- la visualisation des variables et données aux adresses choisies en mémoire,
- la visualisation de l'évolution des registres.

N.B : Sur les outils du C50 (dsk5l et dsk5d), il est possible de quitter par la combinaison Ctrl+Pause ce qui a pour effet de rendre la main au PC tandis que le DSP continue à travailler seul ; sur le C31, la sortie normale des programmes (dsk3l et dsk3d) a pour effet de laisser également le DSP travailler seul.

INTERFACE GRAPHIQUE POUR LE DEVELOPPEMENT SUR DSP

Afin de simplifier l'utilisation des DSP, une interface graphique a été développée (cf. figure ci-dessous).

Cette interface permet d'une part d'appeler Matlab automatiquement mais surtout d'appeler les différents logiciels de développement d'un programme pour le DSP.
Cette interface se compose de trois parties :



- Sur le haut, quatre boutons « demos » (seul les boutons actifs correspondent à une démo existante),
- Dans la partie « Conception de filtres », trois champs à remplir ("nom du fichier", sans extension, choix d'un couple "fréquence d'échantillonnage-fréquence du filtre d'anti-repliement", validation ou non validation du "filtre d'anti-repliement") qui devront être remplis avant d'appeler les trois programmes de développement pour DSP. En-dessous se trouvent trois boutons à cliquer pour exécuter chacune des trois phase de développement d'un programme pour DSP :
  - fnr3 (ou fnr5) qui permet de créer le source du programme assembleur en fonction des coefficients sauvegardés sous Matlab (voir TP 2), ainsi que des valeurs choisies dans les champs de cette partie de l'interface ; il faut impérativement avoir rempli tous ces champs au préalable),
  - dsk3a (ou dsk5a) qui permet de transformer le source assembleur en un programme implantable sur le DSP
  - dsk3l (ou dsk5l) qui permet d'envoyer le programme sur le DSP et de lancer son exécution.

Pour pouvoir utiliser l'interface graphique, il faut au préalable l'avoir configurée.

Pour cela, il faut cliquer sur le bouton « paramètres » afin de configurer les chemins d'accès aux différents programmes que l'on va utiliser. Normalement, vous ne devez configurer que la partie concernant votre propre répertoire, les autres chemins (démos, utilitaires DSP, Matlab) ayant déjà été configurés.

3. PRISE EN MAIN DES DSP TMS320

- lancer l'interface graphique.
- régler le GBF en prenant un signal sinusoïdal d'amplitude 5V crête à crête et de fréquence 200 Hz et visualiser le signal d'entrée sur la voie 1 de l'oscilloscope.
- brancher le signal issu du GBF sur l'entrée du DSP.
- brancher la sortie du DSP à la voie 2 de l'oscilloscope. **ATTENTION** : le signal envoyé par le GBF au boîtier DSP devra **ETRE COMPRIS** entre **-5 et 5 VOLTS MAXI**.
- relier le DSP sur l'ordinateur via la ligne RS232, ou l'interface parallèle puis assurer l'alimentation électrique du DSP.

3.1. QUELQUES APPLICATIONS DU DSP (pour le C50 uniquement)

Les DSP peuvent servir à différents types d'application. En traitement du signal, outre le filtrage numérique qui sera abordé dans la suite, les principales applications sont :

- générateur de signaux
- oscilloscope numérique
- analyseur de spectre

Fichiers	Description
Répertoire **FUNCTION**	Générateur de signaux sur le PC. **Exécuter *FUNC.EXE*.** On a le choix entre un générateur de bruit ("Noise") et de sinusoïde ("SineGen"). Cliquez sur "SineGen". Les touches 0, 1, 2 et 3 permettent de sélectionner respectivement les unités, les dizaines, les centaines et les milliers pour faire varier la fréquence par appui sur les touches ↑ et ↓ Générer une sinusoïde de 1000 Hz, la visualiser à l'oscilloscope, cliquez sur 3 puis sur ↑ (la fréquence augmente à 2000 puis 3000 Hz), cliquez sur 2 puis sur ↑ (la fréquence passe à 3100, 3200, ...). Pour quitter cette démo, appuyez sur la touche "Esc".
Répertoire **OSCOPE**	Oscilloscope numérique sur le PC. **Exécuter *OSCOPE.EXE*.** L'appui sur C permet de d'avoir une liste des commandes. → et ← pour faire varier la fréquence (équivalent au potentiomètre Time/Div d'un oscilloscope), ↑ et ↓ pour les variations d'amplitude (équivalent au potentiomètre Volt/Div d'un oscilloscope Pour quitter cette démo, appuyez sur la touche "Esc".
Répertoire **SPEC**	Analyseur de spectre (FFT) sur l'écran du PC. **Exécuter *SPEC50.EXE*.** Observer les phénomènes de repliement en augmentant la fréquence du signal du GBF. Rappel : la fréquence d'échantillonnage du DSP est par défaut de 10 Khz. L'écran du PC affiche donc la partie utile du spectre du signal d'entrée. **Attention** : les touches Q et A sont inversées. Quitter en appuyant sur la touche "A".

N.B. : Si un programme ne fonctionne pas (notamment SPEC50), débrancher l'alimentation électrique du DSP, attendre quelques secondes et rebrancher.

3.2. QUELQUES APPLICATIONS DU DSP (pour le C30 uniquement)

Les DSP peuvent servir à différents types d'application. En traitement du signal, outre le filtrage numérique qui sera abordé dans la suite, les principales applications sont :

- oscilloscope numérique
- analyseur de spectre

Fichiers	Description
Fichier **DSK_OSC**	Oscilloscope numérique sur le PC. **Exécuter *DSK_OSC.EXE*.** Les touches F1 à F8 permettent de faire varier le contenu des registres TA, TB, RA et RB de l'AIC et donc de jouer sur la fréquence d'échantillonnage et la configuration du filtre d'anti-repliement. Les touches 1 à 6 permettent de configurer le registre de contrôle de l'AIC. ***Pour valider le fonctionnement de l'oscilloscope numérique, actionner la touche 5,*** la touche 6 permet de valider ou d'invalider le filtre d'anti-repliement. **Attention** : les touches Q et A sont inversées. Quitter en appuyant sur la touche "A".
Fichiers **DSK_FFT et DSK_FFTB**	Analyseur de spectre (FFT) sur l'écran du PC. **Exécuter *DSK_FFT.EXE ou DSK_FFTB.EXE*.** Observer les phénomènes de repliement en augmentant la fréquence du signal du GBF. Rappel : la fréquence d'échantillonnage du DSP est par défaut de 46 Khz. L'écran du PC affiche donc la partie utile du spectre du signal d'entrée. **Attention** : les touches Q et A sont inversées. Quitter en appuyant sur la touche "A".

Si le DSP ne fonctionne pas bien (en passant d'un programme à l'autre), le plus simple est de lancer l'exécution d'un programme tel que dsk_osc et de réinitialiser le DSP (appui sur R).

3.3. UTILISATION DE L'ANALYSEUR DE SPECTRE - VERIFICATION DES INFLUENCES DE L'ECHANTILLONNAGE SUR LA REPONSE FREQUENTIELLE

En utilisant l'analyseur de spectre (SPEC50 sur le C50, DSK_FFT ou DSK_FFTB sur le C31), effectuer les manipulations suivantes :

- Générer un signal sinusoïdal de 400 Hz. Observer et commenter les raies obtenues : fréquences, atténuations en dB). Que représentent ces raies ?
- Refaire les observations avec un signal triangulaire, puis un signal rectangulaire, de 400 Hz également.
- Revenir au signal sinusoïdal. Faire les mêmes observations que ci-dessus pour un signal de fréquence 9600 Hz, puis 10400 Hz.
- Quelle est la fréquence d'échantillonnage utilisée par le DSP ?
- Observer le repliement de spectre sur l'analyseur.

4. APPLICATION SUR LE DSP DE FILTRES RIF

Cette partie permet d'implanter et de tester sur le DSP les filtres non récursifs synthétisés précédemment. Les fichiers utiles se trouvent dans le répertoire `\TDS\DSP\` dans le répertoire "pub" des étudiants (recopiez, si nécessaire, les fichiers dans votre répertoire de travail).

Procédure pour tester des filtres provenant de Matlab sur le DSP

La procédure à suivre est la suivante :

A. Lancer l'interface graphique (DSP30.EXE ou DSP50.EXE selon le type de DSP) après l'avoir configurée si nécessaire.

B. Lancer Matlab depuis l'interface graphique
- définir dans la variable ***num*** les coefficients du numérateur du filtre *(ex : num=[0.5 0.5] ou num=fir1(52,0.4375)).*
- définir dans la variable ***den*** les coefficients du dénominateur du filtre, si besoin.
- exécuter la commande ***sauve5*** pour le C50 (ou ***sauve3*** pour le C31) qui permet de sauvegarder les coefficients du filtre dans un fichier spécifique ***nom_fichier.sau*** pour le C50 ou ***nom_fichier.sa3*** pour le C31, dont le nom est demandé à l'utilisateur (le fichier généré devra avoir obligatoirement l'extension **.sau** pour le C50 ou **.sa3** pour le 31).

C. Depuis l'interface graphique
- remplir la zone de droite (nom du fichier, choix du couple fréquence d'échantillonnage-fréquence de coupure du filtre d'anti-repliement, "avec" ou "sans" filtre d'anti-repliement)
- cliquez sur le bouton ***fnr3*** ou ***fnr5***. Ceci va générer un source pour le DSP (extension automatique .asm).
- assembler le programme en cliquant sur le bouton ***dsk3a*** ou ***dsk5a.***
- Téléchargez le programme dans le DSP en cliquant sur le bouton ***dsk3l*** ou ***dsk5l***.
- tester le filtre conçu.

Régler le GBF en prenant un signal sinusoïdal d'amplitude 5V crête à crête et de fréquence appropriée et visualiser le signal d'entrée sur la voie 1 de l'oscilloscope.

Brancher l'entrée du DSP au GBF.

Brancher la sortie du DSP à la voie 2 de l'oscilloscope.

1. Tester le filtre synthétisé en 2.3.1. lors du TP précédent.

Observer les phénomènes d'oscillations en bande passante et en bande atténuée, mesurer la largeur de la zone de transition, donner une estimation de la fréquence de coupure, conclusion.

2. Refaire les manipulations pour le filtre préparé en 2.3.2.

3. Refaire les manipulations pour le filtre préparé en 2.3.3.

4. Refaire les manipulations pour le filtre préparé en 2.3.4.

ETUDE DES PROBLEMES DE REPLIEMENT (ALIASING)

Pour cette étude, on se basera sur le fichier ***direct5*** sur le C50 (***direct3*** sur le C31) qu'il faudra récupérer dans le répertoire `\TDS\DSP\` dans le répertoire "pub" des étudiants (recopiez, si nécessaire, les fichiers dans votre répertoire de travail) avant de l'assembler et de l'exécuter. Ce programme retransmet le signal d'entrée à la sortie avec une fréquence d'échantillonnage à trouver.

- Pour un signal sinusoïdal de 200 Hz, mesurer le signal de sortie en voie 2 : amplitude, fréquence, déphasage entre l'entrée et la sortie (Nota: il sera sûrement plus aisé de mesurer le retard du signal 2 par rapport au signal 1, donnez l'expression mathématique liant le retard τ au déphasage ϕ).
- Refaire successivement la manipulation avec un signal de même amplitude et de fréquences 1000 Hz, 4000 Hz.
- Faire varier continûment la fréquence du signal d'entrée de 1000 à 11000 Hz. Que se passe-t-il ?
- Refaire successivement la manipulation avec un signal de même amplitude et de fréquences 6000 Hz, 9000 Hz, 9800 Hz. Comparer les résultats obtenus pour F_{signal} = 9000 Hz à ceux obtenus dans le cas où F_{signal} = 1000 Hz ; points communs, différences, pourquoi ?

- Que vaut la fréquence d'échantillonnage ?
- Refaire rapidement certaines de ces observations avec un signal d'entrée carré.

Questions générales

1. Donner quelques exemples d'application de filtres numériques.
2. Quel intérêt pour votre formation d'appréhender ces problèmes ?
3. Pourquoi le CAN est-il précédé d'un filtre anti-repliement ?
4. Pourquoi le CNA est-il suivi par un filtre passe-bas de reconstruction (ou de lissage) ?

MISE EN EVIDENCE DES PROBLÈMES D'ÉCHANTILLONNAGE SUR DSP

Cette partie permet d'étudier les problèmes d'échantillonnage à l'aide des algorithmes de filtrage.

Initialisation du CAN : déclaration de la fréquence d'échantillonnage. Elle est donnée par deux formules :

En transmission :	En Réception :
fréquence du processeur	Fréquence du processeur
2*TA*TB	2*RA*RB

Remarques : les registres TA, RA, TB et RB sont des registres de 6 bits. Leur valeur maximale est donc 63, sauf TA qui doit être compris entre 4 et 31. La fréquence du processeur est de 10 MHz dans le cas du C50, programmable dans le cas du C31 (souvent 12,8 MHz dans les programmes que l'on utilise).

Le filtre d'anti-repliement est en fait un filtre passe-bande dont les fréquences de coupure sont calibrées par le contenu du registre TA.

Fréquence basse :	Fréquence haute :
fréquence du processeur*0.3	Fréquence du processeur*3.6
288*2*TA	288*2*TA

Il faudra choisir judicieusement le couple TA,TB pour déterminer la valeur de la fréquence d'échantillonnage et celle des fréquences de coupure du filtre d'anti-repliement, l'idéal étant que la fréquence haute du filtre d'anti-repliement soit un peu en-dessous de la demi-fréquence d'échantillonnage…

- Calculer la fréquence d'échantillonnage et la fréquence de coupure du filtre anti-repliement en fonction des paramètres donnés dans `direct5.asm` (ou `direct3.asm`).

- Brancher l'entrée du DSP au GBF en prenant un signal carré d'amplitude 1V et de fréquence 0.3 KHz et visualiser le signal d'entrée sur la voie 1 de l'oscilloscope.

- Faire varier la fréquence du signal d'entrée de 0 à F_e (et même un peu plus) sans filtre anti-repliement, puis avec celui-ci. Conclure sur l'utilisation du filtre anti-repliement.

- Faire varier la fréquence d'échantillonnage, calculer les valeurs à donner à TA/RA et à TB/RB pour que la fréquence d'échantillonnage soit proche de 18 kHz et la fréquence de coupure du filtre anti-repliement égale à $F_e/2$. Donner les valeurs exactes obtenues pour ces deux fréquences.

En pratique, il ne sera pas nécessaire de modifier les contenus des registres, l'interface graphique permettra de tester certains exemples pré-définis.

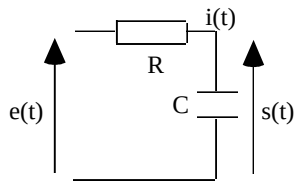
3. ANALYSE ET SYNTHÈSE DE FILTRES NUMÉRIQUES RÉCURSIFS (OU RII) SOUS MATLAB

Lorsque vous lancez Matlab, vérifiez que vous êtes dans votre répertoire, la commande `cd` permet de savoir dans quel répertoire on se trouve et la commande `cd h:`, par exemple permet de changer de disque.

1. PREPARATION THÉORIQUE

La transformation bilinéaire permet d'obtenir la fonction de transfert et/ou l'équation de récurrence d'un filtre numérique ayant, tout au moins dans une plage de fréquences données, quasiment le même comportement fréquentiel qu'un filtre analogique de référence.

On cherche à synthétiser un filtre numérique ayant le même comportement fréquentiel qu'un filtre analogique de type RC avec R=40 Ω et C = 1 μF. On prendra F_e = 200 000 Hz ou $\omega_e = 1.25\times10^6$ rd/s.



- Donner la fonction de transfert du filtre analogique en précisant la fréquence/pulsation de coupure et le gain statique.

- Calculer la fonction de transfert du filtre numérique par utilisation de la transformation bilinéaire : déterminer la valeur de k : le facteur d'échelle, donner la valeur de p_n. Donner la fonction de transfert du filtre numérique.

Remarques :

R1 : Dans la plupart des situations réelles, la fonction de transfert du filtre analogique de référence ne sera pas connue; c'est un gabarit type qui sera utilisé. La technique consistera alors à :

- synthétiser le filtre analogique à l'aide d'une des méthodes connues (Butterworth, Chebyshev1 ou 2, Cauer …)
- puis obtenir le filtre numérique en utilisant la transformation bilinéaire.

R2 : Lorsqu'on effectue la synthèse d'un filtre "ayant des pôles" (*filtre numérique récursif* ou *filtre analogique*) et notamment si l'ordre (donc le nombre de coefficients) est élevé, des problèmes d'arrondis numériques des coefficients pourront avoir lieu. Dans ce cas, il est ***très important*** de vérifier les propriétés du filtre (réponse impulsionnelle, réponse fréquentielle, avec `tp3` par exemple) avant de l'implanter. Dans certains cas, il est possible que le filtre ne soit pas stable… La solution consistera alors à réduire (éventuellement de manière considérable) l'ordre du filtre, typiquement un ordre de 8 ou 10 est déjà bien suffisant, sachant que l'on pourra jouer sur la position des pôles si l'on désire améliorer les performances.

2 TRAITEMENT DU SON PAR FILTRE NUMÉRIQUE

Il est proposé de refaire la manipulation de traitement de son du TP précédent, cette fois-ci avec des filtres récursifs. Les synthèses seront effectuées en prenant si possible les mêmes caractéristiques de gabarit de filtre, afin de comparer les performances entre les filtres non récursifs et les filtres récursifs obtenus.

2.1 Extraction d'un accord de do majeur grave

Soit un son composé de la superposition d'un accord de do majeur grave (do=262 Hz, mi =330 Hz; sol=392 Hz) auquel est superposé un accord de do majeur plus aigu (do=524 Hz, mi =660 Hz; sol=784 Hz). L'ensemble est échantillonné à 2000 Hz.

Il est demandé de concevoir un filtre numérique récursif permettant de supprimer le do majeur aigu. Quelques indications sont données dans le fichier sourec1.m que l'on pourra compléter, la fonction `son_ft.m` pourra également être

utilisée. (Avant toute chose, veuillez copier dans votre répertoire de travail les fichiers `tp3.m`, `sourec1.m`, `sourec2.m` et `son_ft.m`, ces fichiers se trouvent dans le répertoire pub des étudiants).

2.2 Elimination du mi

Dans l'accord résultant (do majeur grave), on souhaite maintenant éliminer le mi central et ne conserver que le do et le sol. Il est demandé là encore de concevoir un filtre numérique non récursif permettant d'assurer cette fonctionnalité. Quelques indications sont données dans `sourec2.m` que l'on pourra compléter.

3. ANALYSE EN SIMULATION SOUS MATLAB D'UN FILTRE PUREMENT RÉCURSIF DU PREMIER ORDRE

Simulation de la fonction de transfert d'un filtre purement récursif du premier ordre de fonction de transfert :

$$H(z) = \frac{1}{1 - az^{-1}}$$

A quelles conditions ce filtre est-il stable ?

Faire varier la valeur de a (<-1, comprise entre -1 et 1, =1, >1) afin d'étudier la stabilité de ce filtre et l'allure des réponses impulsionnelles et indicielles en fonction de a, utiliser la procédure tp3. Dans cet exercice, il est intéressant de regarder ce qui se passe lorsque a est proche de 1, pourquoi ?

Conclure sur l'appellation de filtre à réponse impulsionnelle infinie.

Par rapport aux filtres RIF, ces filtres présentent l'inconvénient d'être à RII mais l'avantage d'avoir une fonction de transfert comprenant un dénominateur. Cette dernière caractéristique permet de pouvoir faire varier plus de paramètres pour la conception des filtres. Il est par conséquent possible d'obtenir une réponse aussi satisfaisante que pour un filtre RIF avec un ordre beaucoup moins élevé.

Définir de quel type de filtre il s'agit (passe-bas, passe-haut, coupe-bande, passe-bande) en précisant la valeur de la fréquence de coupure en fonction de a pour quelques cas.

Dans les conclusions, on abordera les problèmes suivants : stabilité, réponse en amplitude, déphasage. Conclure sur l'intérêt et les caractéristiques de filtres récursifs.

4. SYNTHESE DE FILTRES RÉCURSIFS PAR LA TRANSFORMATION BILINÉAIRE

On pourra utiliser la procédure Matlab ***bilinear*** où den et num sont les vecteurs contenant respectivement les coefficients du numérateur et les coefficients du dénominateur du filtre numérique récursif. Cette procédure permet d'obtenir les coefficients (et donc la fonction de transfert) du filtre numérique à partir des coefficients (et donc la fonction de transfert) du filtre analogique.

```
BILINEAR transformation bilinéaire avec calibration fréquentielle (frequency prewarping)

[num,den] = bilinear(numa,dena,Fe,Fr);

numa et dena sont des tableaux contenant les coefficients du numérateur, respectivement
du dénominateur, de la fonction de transfert du filtre analogique, numa(p)/dena(p), en
puissances décroissantes en p.
Fe est la fréquence d'échantillonnage en Hz.
Si Fr (fréquence de référence) est spécifiée (en Hz), cette fonction applique une
calibration fréquentielle avant d'effectuer la transformation bilinéaire de telle
manière qu'il y ait correspondance exacte entre les valeurs des réponses fréquentielles
du filtre analogique et du filtre numérique, en ce point.
```

Lorsque l'on ne connaît pas de filtres analogiques de référence, il est possible d'utiliser directement les méthodes de synthèse de filtres (analogiques ou numériques) butter, cheby1 ou 2, ellip … (voir plus loin).

2.1 : Trouver les coefficients du numérateur et du dénominateur du filtre récursif étudié dans la partie "préparation théorique" à l'aide de Matlab et simuler le filtre ainsi obtenu.

Rappel : la procédure `tp3` permet d'effectuer la représentation des réponses.

2.2 Refaire la manipulation pour l'exercice de TD, si le temps le permet.

2.3 Utilisation des méthodes de synthèse de filtres analogiques combinées avec la transformation bilinéaire

2.3.1 Outils disponibles

Ordres des filtres

MATLAB nous permet d'exécuter des procédures nous permettant d'obtenir l'ordre optimal du filtre numérique afin de résoudre un problème. Il faut ensuite utiliser les procédures explicitées dans la partie précédente pour effectuer la synthèse.

BUTTORD Butterworth filter order selection.
[N, fn] = buttord(fp, fa, Ap, Aa) returns the order N of the lowest order digital Butterworth filter that loses no more than Ap dB in the passband and has at least Aa dB of attenuation in the stopband. The passband runs from 0 to fp and the stopband extends from fa to 1.0, the Nyquist frequency. BUTTORD also returns fn, the Butterworth natural frequency to use with BUTTER to achieve the specifications.
[N, Wn] = buttord(fp, fa, Ap, Aa, 's') does the computation for an analog filter.
When Ap is chosen as 3 dB, the fn in BUTTER is equal to fp in BUTTORD.

Sur le même principe on a :

BUTTORD	Ordre optimal pour un filtre de Butterworth	BUTTER
CHEB1ORD	Ordre optimal pour un filtre de Chebyshev type I	CHEBY1
CHEB2ORD	Ordre optimal pour un filtre de Chebyshev type II	CHEBY2
ELLIPORD	Ordre optimal pour un filtre elliptique	ELLIP

Filtres de Butterworth

BUTTER Butterworth digital filter design.
[num,den] = butter(N,fn) designs an N'th order lowpass digital Butterworth filter and returns the filter coefficients in length N+1 vectors num and den. The cut-off frequency fn must be 0.0 < fn < 1.0, with 1.0 corresponding to half the sample rate.
If fn is a two-element vector, fn = [f1 f2], BUTTER returns an order 2N bandpass filter with passband f1 < f < f2.
[num,den] = butter(N,fn,'high') designs a highpass filter.
[num,den] = butter(N,fn,'stop') is a bandstop filter if fn = [f1 f2].

N.B. : L'utilisation du paramètre 's' permet de concevoir des filtres analogiques pour la procédure butter et toutes les autres :
ex : [num,den] = butter(N,Wn,'high','s')

Filtres de Chebychev type I

CHEBY1 Chebyshev type I digital filter design.
[num,den] = cheby1(N,Ap,fn) designs an N'th order lowpass digital Chebyshev filter with Ap decibels of ripple in the passband. CHEBY1 returns the filter coefficients in length N+1 vectors num and den. The cut-off frequency fn must be 0.0 < fn < 1.0, with 1.0 corresponding to half the sample rate. Use Ap=0.5 as a starting point, if you are unsure about choosing Ap.
If fn is a two-element vector, fn = [f1 f2], CHEBY1 returns an order 2N bandpass filter with passband f1 < f < f2.
[num,den] = cheby1(N,Ap,fn,'high') designs a highpass filter.
[num,den] = cheby1(N,Ap,fn,'stop') is a bandstop filter if fn = [f1 f2].

Filtres de Chebychev type II (ou inverses)

CHEBY2 Chebyshev type II digital filter design.
[num,den] = cheby2(N,Aa,fn) designs an N'th order lowpass digital Chebyshev filter with the stopband ripple Aa decibels down. CHEBY2 returns the filter coefficients in length N+1 vectors num and den. The cut-off frequency fn must be 0.0 < fn < 1.0, with 1.0 corresponding to half the sample rate. Use Aa = 20 as a starting point, if you are unsure about choosing Aa.
If fn is a two-element vector, fn = [f1 f2], CHEBY2 returns an order 2N bandpass filter with passband f1 < f < f2.
[num,den] = cheby2(N,Aa,fn,'high') designs a highpass filter.
[num,den] = cheby2(N,Aa,fn,'stop') is a bandstop filter if fn = [f1 f2].

Filtres elliptiques (ou de Cauer)

ELLIP Elliptic or Cauer digital filter design.
[num,den] = ellip(N,Ap,Aa,fn) designs an N'th order lowpass digital elliptic filter with Ap decibels of ripple in the passband and a stopband Aa decibels down. ELLIP returns the filter coefficients in length N+1 vectors num and den. The cut-off frequency fn must be 0.0 < fn < 1.0, with 1.0 corresponding to half the sample rate. Use Ap = 0.5 and Aa = 20 as starting

points, if you are unsure about choosing them.
If fn is a two-element vector, fn = [f1 f2], ELLIP returns an order 2N bandpass filter with passband f1 < f < f2.
[num,den] = ellip(N,Ap,Aa,fn,'high') designs a highpass filter.
[num,den] = ellip(N,Ap,Aa,fn,'stop') is a bandstop filter if fn = [f1 f2].

Filtres de Yulewalk

YULEWALK Recursive filter design using a least-squares method.
[num,den] = yulewalk(N,F,M) finds the N-th order recursive filter coefficients num and den such that the filter:

```
                          -1              -(n-1)
B(z)   b(1) + b(2)z + .... + b(n)z
---- = -------------------------
                          -1              -(n-1)
A(z)    1   + a(1)z + .... + a(n)z
```

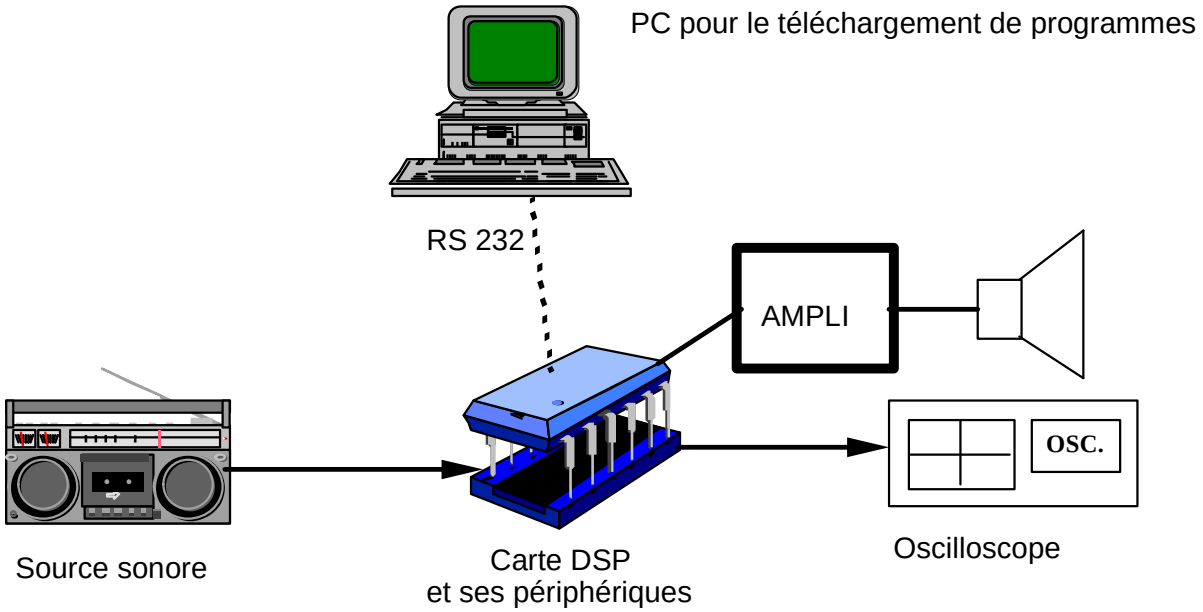
matches the magnitude frequency response given by vectors F and M. Vectors F and M specify the frequency and magnitude breakpoints for the filter such that PLOT(F,M) would show a plot of the desired frequency response. The frequencies in F must be between 0.0 and 1.0, with 1.0 corresponding to half the sample rate. They must be in increasing order and start with 0.0 and end with 1.0.

2.4.2 Manipulations

Synthétiser les filtres numériques étudiés dans la partie "Préparation" du TP 2 à l'aide de la méthode de Butterworth. Tout d'abord donner l'ordre à l'aide de BUTTORD puis synthétiser avec BUTTER et simuler avec tp3. Refaire ensuite les manipulations avec les autres méthodes (Tchebychev I et II, Cauer).

DSP : Application au traitement du signal sonore

On pourra utiliser le processeur de traitement du signal pour traiter le signal sonore. Pour cela, il faudra construire un petit amplificateur interface entre la sortie de la carte DSP et le haut-parleur. L'ampli permettra de sortir le signal sur HP. La radio ou le magnétophone servira de signal d'entrée.



1. Branchement de la maquette

- Installer la maquette entre deux ordinateurs reliés à des DSP.
- Relier les fiches bleue et rouge du magnétophone sur les entrées "banane" noire et rouge.
- Brancher le magnétophone au secteur.
- Relier un câble BNC entre l'entrée du premier DSP et l'entrée E1 de la maquette.
- Relier un câble BNC entre la sortie du premier DSP et la sortie S1 de la maquette.
- Relier un câble BNC entre l'entrée du second DSP et l'entrée E2 de la maquette.
- Relier un câble BNC entre la sortie du second DSP et la sortie S2 de la maquette.
- Relier l'alimentation sur la maquette (masse puis +9V). Allumer l'alimentation
- Mettre en marche le magnétophone, essayer la liaison directe puis les liaisons via DSP1 et DSP2 après avoir exécuté un programme sur chacun des deux DSP.

2. Tests des programmes de filtrage

Essayer les programmes de filtrage LOPASS, HIGHPASS, BANDPASS, BANDSTOP ainsi que les filtres que vous avez synthétisés pendant les TP.
Essayer les problèmes d'échantillonnage (comparaison FIL1 et FIL1B).
Essayer les programmes OSCOPE, SPEC50 et FUNC.

2.4.2. D'AUTRES APPLICATIONS

D'autres exemples d'application sont disponibles pour le C50 (voir fiche annexe pour le C31) dans le répertoire `Program\DSP\DSKC30\DEMOS` sur la machine nt_gtr (ou depuis j:). A partir de ce répertoire, tester les trois démos ci-dessous :

Fichiers	Description
Fichier **DSK_OSC**	Oscilloscope numérique sur le PC. **Exécuter *DSK_OSC.EXE*.** Les touches F1 à F8 permettent de faire varier le contenu des registres TA, TB, RA et RB de l'AIC et donc de jouer sur la fréquence d'échantillonnage et la configuration du filtre d'anti-repliement (voir `tprep`). Les touches 1 à 6 permettent de configurer le registre de contrôle de l'AIC. ***Pour valider le fonctionnement de l'oscilloscope numérique, actionner la touche 5,*** la touche 6 permet de valider ou d'invalider le filtre d'anti-repliement. **Attention** : les touches Q et A sont inversées. Quitter en appuyant sur la touche "A".
Fichiers **DSK_FFT et DSK_FFTB**	Analyseur de spectre (FFT) sur l'écran du PC. **Exécuter *DSK_FFT.EXE ou DSK_FFTB.EXE*.** Observer les phénomènes de repliement en augmentant la fréquence du signal du GBF. Rappel : la fréquence d'échantillonnage du DSP est par défaut de 46 Khz. L'écran du PC affiche donc la partie utile du spectre du signal d'entrée. **Attention** : les touches Q et A sont inversées. Quitter en appuyant sur la touche "A".

Si le DSP ne fonctionne pas bien (en passant d'un programme à l'autre), le plus simple est de lancer l'exécution d'un programme tel que dsk_osc et de réinitialiser le DSP (appui sur R).

3 ANNEXE AU TP 7 : FILTRES RECURSIFS SUR LE C31

De la même manière que pour les filtres non récursifs, il est possible de récupérer les résultats d'un filtre obtenu sous Matlab pour l'implanter sur le C31. Pour cela, on sauvegarde les coefficients sous Matlab par la procédure SAVRII23.M dans un fichier ayant l'extension **.ii3**. Ensuite sous DOS, le programme rii23.exe génère le source assembleur.

Restrictions pour l'utilisation de ces programmes :

- Avant de sauvegarder des coefficients, vérifier que le filtre obtenu est stable (à l'aide de la commande `tprep`).
- Le nombre de coefficients doit être de trois pour le numérateur et trois pour le dénominateur.

ARCHIVES

Il est possible d'éliminer le filtre anti-repliement sur le DSP en mettant la variable AIC_CTR à 8 ou de le valider en mettant AIC_CTR à 9 ; cette variable se trouve au début des programmes source des fichiers assembleur qui peuvent être modifiés à l'aide d'un éditeur. Modifier la valeur de AIC_CTR et expliquer les changements au niveau du signal de sortie du GBF.

Pour le C31, la variable C_REG peut de la même manière être modifiée : elle vaut 00000011 en binaire lorsque le filtre anti-repliement n'est pas activé, et 00000010 lorsqu'il est actif.

NOTICE INFORMATIQUE

Vous trouverez ci-dessous la liste des logiciels et fichiers nécessaires pour les TP DSP et leur localisation sur les disques durs.

- Avant toute chose,
  - créez dans votre login un répertoire de travail (`h:\tds\dsp\`)
  - allez dans le répertoire "pub" des étudiants dans \TDS\DSP ???????????????????? et téléchargez les fichiers dans ce répertoire de travail. Selon le type de DSP avec lequel vous travaillez, téléchargez le contenu du répertoire correspondant dans votre répertoire de travail (`TP2003C3` pour le C30) et (`TP2003C5` pour le C50).
  - configurez le fichier `dsp30.txt` (ou `dsp50.txt` selon le type de DSP) : pour cela il faudra écrire les chemins correctement notamment pour les parties "emplacement des fichiers sauvés sur Matlab"

```
%Emplacement des exécutables%
j:\dsp\dskc30\

%Emplacement des fichiers sauvés sur matlab%
z:\thil\tds\dsp\

%chemin de matlab %
J:\MATLAB\BIN\MATLAB.EXE

%lecteur courant%
z:\thil\tds\dsp\
```

- S'il n'existe pas, connectez un lecteur réseau j: sur le répertoire Program de la machine Nt_gtr

Les démonstrations disponibles se trouvent dans le répertoire suivant :

Les démos des applications possibles des DSP	se trouvent dans le répertoire `j:\DSP\DSKC30\DEMOS` ou `j:\DSP\DSKC50\DEMOS` sur la machine nt_gtr. Les démos sont des programmes exécutables qu'il convient de lancer directement depuis leurs répertoires dans une fenêtre DOS.

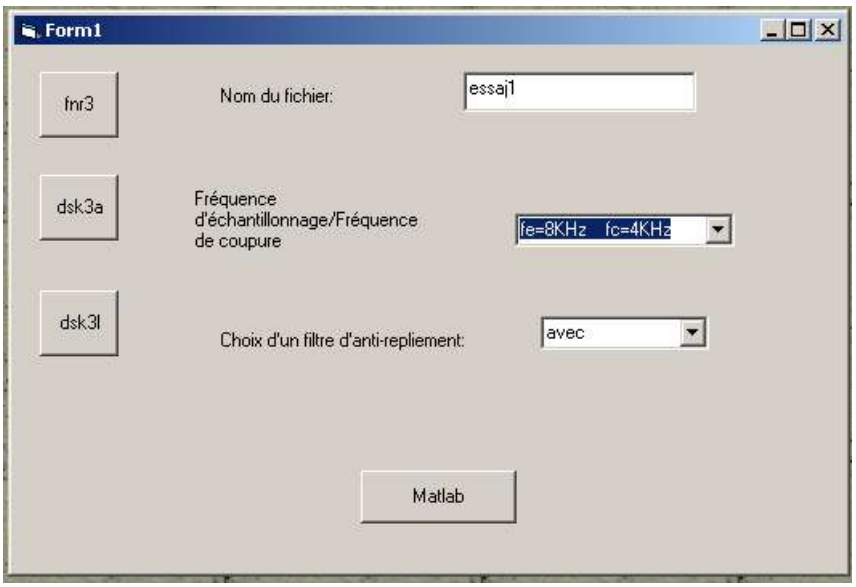
S'il y a des problèmes, l'organisation des disques est la suivante :

Logiciel ou fichiers	Comment les utiliser	Comment vous dépanner
MATLAB	S'utilise directement à partir de l'icône MATLAB. Une fois dans Matlab, faire cd `j:\`*`votre_rep_tds`* Pour que Matlab trouve les fichiers dans votre répertoire de travail	Matlab se trouve dans le répertoire `Program\Matlab\bin\matlab.exe` sur la machine nt_gtr ou `j:\Matlab\bin\matlab.exe`
Kit de développement DSP (assembleur, débogueur, chargeur ("loader") A UTILISER DANS UNE FENETRE MS-DOS ou depuis l'interface graphique (cf. page suivante)	Depuis votre répertoire de travail (dans une fenêtre MS-DOS), l'assembleur (dsk3a ou dsk5a), le débogueur (dsk3d ou dsk5d) et le chargeur (dsk3l ou dsk5l) sont accessibles directement	Le kit de développement se trouve dans le répertoire `Program\DSP\DSKC30\` ou `Program\DSP\DSKC50\` sur la machine nt_gtr (ou depuis j:) Si le kit de développement est inaccessible, il faut ajouter les logiciels dans le chemin de

		DOS, pour cela vous pouvez exécuter `envc30` ou `envc50` (ces deux fichiers se trouvent dans `\TDS\DSP` dans le répertoire pub des étudiants).

INTERFACE GRAPHIQUE POUR LE DEVELOPPEMENT SUR DSP

Afin de simplifier l'utilisation des DSP, une interface graphique a été développée (cf. figure ci-dessous).



Cette interface permet d'une part d'appeler Matlab automatiquement mais surtout d'appeler les différents logiciels de développement d'un programme pour le DSP.

Cette interface se compose de deux parties :

- Sur la droite, trois champs à remplir ("nom du fichier", choix d'un couple "fréquence d'échantillonnage-fréquence du filtre d'anti-repliement", validation ou non validation du "filtre d'anti-repliement") qui devront être remplis avant d'appeler les trois programmes de développement pour DSP
- Sur la gauche trois boutons à cliquer pour exécuter chacune des trois phase de développement d'un programme pour DSP :
  - fnr3 (ou fnr5) qui permet de créer le source du programme assembleur en fonction des coefficients sauvegardés sous Matlab (voir TP 6), ainsi que des valeurs choisies dans la partie droite de l'interface (il faut impérativement avoir rempli la partie droite au préalable),
  - dsk3a (ou dsk5a) qui permet de transformer le source assembleur en un programme implantable sur le DSP
  - dsk3l (ou dsk5l) qui permet d'envoyer le programme sur le DSP et de lancer son exécution.

N.B. Pour pouvoir utiliser correctement l'interface graphique, il faut au préalable l'avoir configurée (cf. page précédente).