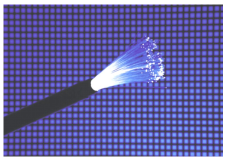


TP n°26 : synthèse d'un filtre numérique. Application aux signaux sonores.

➔ <u>But du TP</u> : le but de ce dernier TP de seconde année est l'étude de quelques solutions pour synthétiser un filtre numérique à partir d'un filtre analogique, c'est-à-dire comment trouver les coefficients du filtre numérique à partir des caractéristiques du filtre analogique. On se servira de l'étude par la simulation (tableur) effectuée au TP précédent et de Labview associé à la carte d'acquisition pour étudier l'influence du filtre sur les signaux. Enfin, on utilise le logiciel Labview associé à la carte d'acquisition pour une étude sur un fichier son.

A) Synthèse d'un filtre numérique du premier ordre par approximation de l'équation différentielle.

Pour réaliser un filtre numérique , on peut partir de l'équation différentielle correspondant au filtre analogique que l'on désire synthétiser. On remplace alors la dérivée par une approximation, ce qui revient à remplacer la variable de Laplace p par une équivalence en Z. La fonction de transfert en Z obtenue représentera un filtre numérique (s'il est stable) dont le comportement sera proche du filtre analogique **dans une certaine gamme de fréquence**.

<u>Réalisation d'un filtre passe-bas du premier ordre correspondant à la fonction intégration.</u>

On a vu dans le Tp précédent qu'un système du premier ordre d'équation : $\tau.\frac{ds(t)}{dt}+s(t)=\mathrm{e}(t)$

correspondait au filtre numérique d'équation : $S_n=\left(\frac{\alpha}{1+\alpha}\right).S_{n-1}+\left(\frac{1}{1+\alpha}\right).E_n$ où $\alpha=\tau\times\mathrm{fe}$

car on avait fait l'approximation : $\frac{ds(t)}{dt}=\frac{s_n-s_{n-1}}{Te}$.

En utilisant la même méthode, démontrer que la fonction intégration :

$$e(t) \longrightarrow \boxed{\text{intégration}} \longrightarrow s(t) \text{ tel que } e(t) = \frac{ds(t)}{dt}$$

correspond à l'algorithme : $\mathrm{s_n=s_{n-1}+Te.e_n}$ avec , fe étant la fréquence d'échantillonnage.

<u>Remarque</u> : l'intégration est alors approchée par la méthode des rectangles supérieurs.

Déterminer la fonction de transfert Ti(z) de l'intégrateur correspondant.

Rappeler la fonction de transfert de Laplace correspondant à la fonction intégration.
En déduire que l'on peut remplacer p par $\mathrm{p=fe\cdot(1-z^{-1})}$

Rappeler l'expression de la fonction de transfert de Laplace H(p) correspondant au filtre analogique passe-bas du premier ordre de fréquence de coupure fc et d'amplification statique égale à 1. Déterminer alors l'expression de la fonction de transfert H(z) du filtre numérique correspondant en remplaçant p par $\mathrm{p=fe\cdot(1-z^{-1})}$

En déduire que l'algorithme du filtre s'écrit $y_n = a_0 \cdot x_n + b_1 \cdot y_{n-1}$ avec $a_0 = \dfrac{2\pi \cdot \dfrac{fc}{fe}}{1 + 2\pi \cdot \dfrac{fc}{fe}}$ et

$$b_1 = \frac{1}{1 + 2\pi \cdot \dfrac{fc}{fe}}\ .$$

Application numérique: On désire un filtre de fréquence fc=100Hz avec une fréquence fe=11025Hz, calculer les coefficients a_0 et b_1.

Détermination expérimentale des caractéristiques du filtre.

Ouvrir le «v.i.» intitulé « filtrage numérique sorties carte .vi» et rentrer les valeurs de a_0 et b_1 précédentes.
Ce «v.i.» permet de tester par simulation le filtre en entrant les coefficients a_0, a_1, a_2, b_1 et b_2 correspondants à l'algorithme: $y_n = a_0 x_n + a_n x_{n-1} + a_2 x_{n-2} + b_1 y_{n-1} + b_2 y_{n-2}$.
Mais ce «v.i.» peut aussi piloter la carte d'acquisition pour donner les signaux échantillonnés et bloqués de l'entrée et de la sortie du filtre numérique.

Réponse harmonique
Par simulation avec le vi, déterminer:

- l'amplification statique,
- la fréquence de coupure
- Le déphasage entre us(t) et ue(t) pour f=100Hz et f=1000Hz

Retrouver ces résultats par des mesures sur les signaux délivrés par la carte d'acquisition.

Les valeurs mesurées sont-elles cohérentes avec les valeurs attendues du filtre analogique ?
Dans quel domaine de fréquence le filtre numérique se comporte comme le filtre analogique?

Réponse indicielle

Valider le bon fonctionnement par simulation avec le vi ou avec le programme réalisé dans le TP précédent sur le tableur.

Évaluer la constante de temps τ, vérifier la cohérence avec les résultats précédents.

B) Synthèse d'un filtre numérique du premier ordre par transformation bilinéaire.

En essayant de se rapprocher de la relation entre p et Z, c'est-à-dire : $Z = e^{p.Te}$ (comme précisé dans l'annexe1), on obtient $p = 2fe \cdot \dfrac{1 - z^{-1}}{1 + z^{-1}}$, c'est la méthode appelée « transformation bilinéaire ».
Remarque : on peut retrouver cette relation dans le cas de l'intégration en prenant l'approximation par la méthode des trapèzes.

Le filtre obtenu admet comme équation aux différences : $y_n = a_0 \cdot x_n + a_1 \cdot x_{n-1} + b_1 \cdot y_{n-1}$ avec

$$a_0 = a_1 = \frac{\pi \cdot \frac{fc}{fe}}{1+\pi \cdot \frac{fc}{fe}} \quad \text{et} \quad b_1 = \frac{1-\pi \cdot \frac{fc}{fe}}{1+\pi \cdot \frac{fc}{fe}} \quad .$$

Reprenez les manipulations de la question précédente. En comparant les deux filtres numérisés, donnez la méthode qui donne les meilleur résultats. Pouvait-on le prévoir?

<u>Application du filtre sur un son enregistré dans un fichier « wav ».</u>

Pour cela on utilisera un autre « v.i » le fichier: « filtrage numérique-son .vi ». Il permet de traiter les fichiers sons avec des filtres numériques .

Les fichiers « wav » sont le résultat de la numérisation de son . Les sons peuvent être numérisés à différentes fréquences d'échantillonnage qui doivent être les mêmes que les fréquences de restitution du son à savoir : 8000, 11025, 22050 ou 44100 Hz. Il faut noter que la conversion analogique numérique peut être codée en 8 ou 16 bits et que les sons peuvent être enregistrés en mono ou en stéréo.

Tout filtre numérique appliqué aux données de ce genre de fichier devra donc tenir compte de la fréquence d'échantillonnage lors de la numérisation. Pour connaître les caractéristiques des fichiers wav il faut lire l'entête du fichier par un logiciel spécialisé (type audacity) ou un éditeur hexadécimal.

Le « v.i » construit un son en lisant le fichier « wav » auquel il rajoute le son filtré par le filtre numérique défini par ses coefficients sur le premier onglet.

Sur le deuxième onglet on peut observer les chronogrammes des deux signaux l'un derrière l'autre.Les deux derniers onglets donnent respectivement le module (ou le gain) et l'argument de la fonction de transfert.

Rentrer les coefficients a_1 et b_0 et charger le fichier « horse.wav ». Lancer le « v.i » et écouter l'influence du filtre sur le son.

Déterminer la fréquence de coupure du filtre ainsi que l'argument pour cette fréquence.

Imprimer les courbes de gains et d'argument.

Comparer ces courbes avec celles du filtre analogique attendu et conclure.

C) Étude de quelques effets des filtres numériques appliqués à un signal sonore.

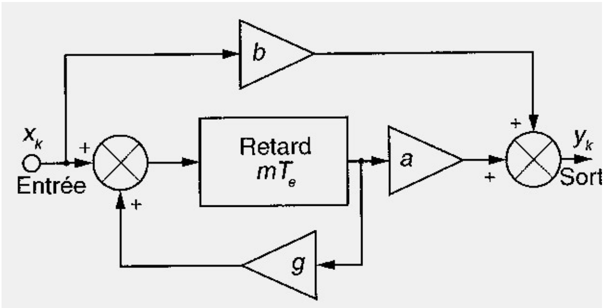
On veut étudier comment trouver les coeficients des filtres numériques qui réalisent les trois effets suivants sur des signaux sonores :

- ➔ écho.
- ➔ révérbération.
- ➔ égaliseur numérique.

L'écho est un phénomène acoustique résultant de la réflexion de l'onde sonore sur un obstacle, cette onde revanant vers la source sonore retardée et atténuée.
Dessiner le schéma symbolique de ce système et en déduire le schéma symbolique du système numérique associé (on utilisera les symboles classiques des retards et des atténuateurs).

L'écho n'est perçu par l'oreille humaine que si le retard entre l'onde émise et l'onde réfléchie est de l'ordre de 50 ms.
On a construit un vi qui permet de lire un fichier son, puis le fichier son travaillé par un écho : ouvrir le fichier « echo.vi ». Visualiser son diagramme pour bien montrer que l'équation de récurrence : yn = (coeff_signal).xn + (coeff_echo).xn-m avec m l'indice qui correspond au retard voulu et essayer ce vi sur le signal « decompte_5_16bits.wav » ou bien sur le signal « hautbois-ancien.wav » avec un retard de 50 ms. Mettre un retard de 10 ms et conclure.

Lorsque le signal initial subit un écho et que cet écho est prolongé par une suite d'obstacles, on parle de révérbération. Le schéma symbolique de ce phénomène peut être décrit par :



Les coefficients :

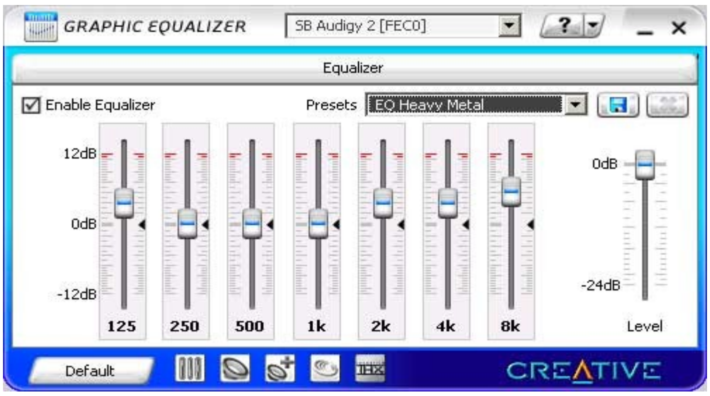
- b règle le gain des sons directs et a celui des sons retardés.
- m règle la durée du retard (comme pour l'écho).
- g règle la persistance de l'écho (il simule le taux de réflexion sur des murs, des obstacles).

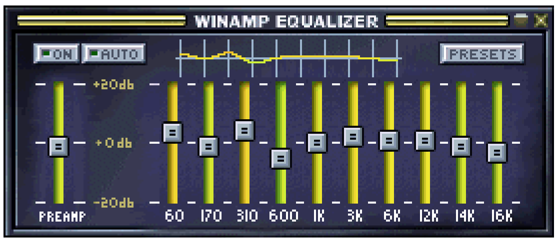
Montrer que l'équation de récurrence du filtre numérique s'écrit :

$$y_n = b.\ x_n + (a- b.g).\ x_{n-m} + g.\ y_{n-m}.$$

Ouvrir le vi « reverberation.vi » et appliquer ce programme au signal sonore « hautbois-ancien.wav » avec un retard de 50 ms et des coefficients a = 0,5 b = 1,5 g = 0,8.
En déduire que cet effet donne de la profondeur à un signal.
Essayer ce même vi sur le fichier « david.wav » qui est un son qui déjà été travaillé et conclure.

On veut maintenant effectuer un égaliseur numérique tel que ceux que l'on trouve sur des logiciels de lecture de fichiers sons numérisés :





L'égaliseur numérique de winamp par exemple coupe la bande sonore (20 Hz – 20 kHz) en dix bandes égales et permet d'amplifier telle ou telle bande dans le signal grâce à des curseurs.

On va donc créer 10 filtres passe-bande en octaves, tels que les fréquences centrales des filtres et leur bande passante vérifient : $f_{0i+1} = 2 \,.\, f_{0i}$ et $\Delta f_{0i+1} = 2 \,.\, \Delta f_{0i}$

TABLEAU 8.3 FRÉQUENCES CENTRALES DES BANDES D'OCTAVE.

f_{01}	f_{02}	f_{03}	f_{04}	f_{05}	f_{06}	f_{07}	f_{08}	f_{09}	f_{010}
31,5 Hz	63 Hz	125 Hz	250 Hz	500 Hz	1 kHz	2 kHz	4 kHz	8 kHz	16 kHz

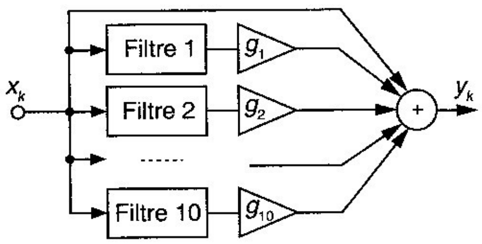


Figure 8.7 Égaliseur spectral en bandes d'octave.

<u>Etude d'un filtre passe-bande</u>

On rappelle que la fonction de transfert de Laplace d'un filtre passe-bande peut s'écrire :

$$H(p) = \frac{2m.\dfrac{p}{\omega_0}}{1+2m.\dfrac{p}{\omega_0}+\left(\dfrac{p}{\omega_0}\right)^2} = \frac{2m.s}{1+2m.s+s^2} \text{ si on note } s=\frac{p}{\omega_0}$$

ω_0 est la pulsation propre et m est le facteur d'amortissement.

Rappeler les relations entre ω_0 et m d'un côté, et f_0 et Δf_0 de l'autre.

En utilisant la transformation bilinéaire , on peut alors montrer que la fonction de transfert en Z peut s'écrire : $H(Z) = \dfrac{a.(1-Z^{-2})}{(1+b.Z^{-1}+c.Z^{-2})}$ avec a, b et c les coefficients donnés par les tableaux suivants :

TABLEAU 8.4 VALEURS DES COEFFICIENTS DU FILTRE POUR LES HUIT PREMIÈRES OCTAVES.

f_0	Ω_0	a	b	c
31	$4{,}4167\ 10^{-3}$	0,00157493	–1,99683067	0,99685015
62	$8{,}8335\ 10^{-3}$	0,0031449	–1,99363241	0,9937102
125	$17{,}809\ 10^{-3}$	0,00632033	–1,98704418	0,98735934
250	0,03562	0,01256127	–1,97362483	0,97487747
500	0,07124	0,02481088	–1,94543141	0,95037825
1 000	0,14247	0,0484204	–1,88387539	0,9031592
2 000	0,28495	0,09236829	–1,74206334	0,81526342
4 000	0,5699	0,16911565	–1,39913128	0,66176869

On construit alors le vi « egaliseur.vi » de cette façon et vous demande de l'utiliser sur les signaux sonores proposés.

. Format des fichiers RIFF_WAV

Analysons l'entête d'un fichier "WAV" en prenant l'exemple d'un fichier mesurant au total 65592 octets et contenant 65536 échantillons :

```
adr:        HEXA                                        ASCII
0000  52 49 46 46  30 00 01 00  57 41 56 45  66 6D 74 20  |RIFF0...WAVEfmt |
0010  10 00 00 00  01 00 01 00  11 2B 00 00  11 2B 00 00  |.........+...+..|
0020  01 00 08 00  66 61 63 74  04 00 00 00  00 00 01 00  |....fact........|
0030  64 61 74 61  00 00 01 00  80 80 80 80  80 80 82 84  |data............|
0040  88 8E A6 88  5E 50 5A 4A  64 76 84 90  A2 9C 9A 9E  |....^PZJdv......|
0050  8C 7C 78 72  5E 5A 62 60  60 72 7E 82  92 9E 9A 96  |.|xr^Zb``r~.....|
0060  9A 90 7C 76  70 62 60 66  68 6C 7E 86  8A 92 9C 9A  |..|vpb`fhl~.....|
etc.
```

L'entête mesure 56 octets, et s'étend donc de l'adresse 00h à l'adresse 38h.

Les fichiers WAV utilisent le format RIFF ; définition du format RIFF (Resource Interchange File Format) :

Octets 1 à 4 : Caractères 'RIFF' [52h 49h 46h 46h] identifiant le format.
Octets 5 à 8 : Longueur du groupe de données au format WAV = [30h.00h.01h.00h] = 00010030h = 65584 octets, (le reste du fichier, car 65584+8 = 65592 octets).
Octets 9 à 16 : Caractères 'WAVEfmt ' identifiant le format WAV.

Les octets 17 à 56 définissent les paramètres du format 'WAV' :

Octets 17 à 20 : [10.00.00.00] = 00000010h = 16 = nombre d'octets utilisés après pour définir le format.
Octets 21 à 22 : [01.00] = 0001h = 1 = numéro de format du fichier (pas de compression, format PCM classique).
Octets 23 à 24 : [01.00] = 0001h = 1 : nombre de canaux : ici, mono.
Octets 25 à 28 : [11.2B.00.00] = 00002B11h = 11025, fréquence d'échantillonnage (en Hz), c'est à dire le nombre d'échantillons pas seconde.
Octets 29 à 32 : [11.2B.00.00] = 00002B11h = 11025, nombres d'octets par seconde, ce qui revient au même car un échantillon mesure un octet et l'on est en mono.
Octets 33 à 34 : [01.00] = 0001h = 1 = Produit du nombre de canaux par le nombre d'octets par échantillon (ici 1x1=1).
Octets 35 à 36 : [08.00] = 0008h = 8 bits par échantillon (valeurs possibles : 8, 12 ou 16).
Octets 37 à 48 : [66.61.63.74][04.00.00.00][00.00.01.00] = 'fact', 4, 65536. Champ sur lequel je n'ai pas de précision ; je pense que 'fact' annonce des informations, que les 4 octets suivants indiquent la taille de ces infos (4 octets), et que dans ce cas, l'info est le nombre d'échantillons, lequel est repris après par 'data', soit 65536.
Octets 49 à 52 : 'data' : annonce l'arrivée des données.
Octets 53 à 56 : [00.00.01.00] = 00010000h = 65536 : taille des données.

Les 65536 octets suivants : valeurs numériques successives des échantillons, oscillant autour de la valeur moyenne 80h, selon l'amplitude et le signe du signal d'entrée...
N.B. : Selon l'éditeur WAV utilisé, on peut remarquer des différences minimes dans la taille des entêtes, mais l'essentiel des informations données ici restent valables.

Annexe 1 :

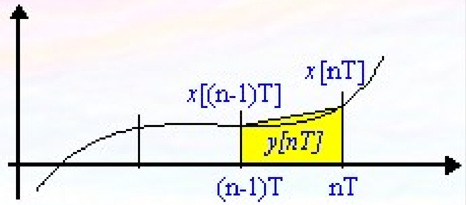
11-3-9 Méthode de la transformation bilinéaire (1/3 - principe) (synthèse RII)

La transformation bilinéaire est la méthode la plus couramment utilisée pour réaliser des filtres discrets à partir de filtres continus.

- z étant égal à e^{Tp}, $p = \dfrac{Log(z)}{T}$ se développe en série sous la forme $p = \dfrac{2}{T}\left[\dfrac{z-1}{z+1} + \dfrac{1}{3}\left(\dfrac{z-1}{z+1}\right)^3 + \dfrac{1}{5}\left(\dfrac{z-1}{z+1}\right)^5 + \ldots\right]$ (démonstration)

  La transformation bilinéaire en est le développement limité au 1er ordre : $\boxed{p = \dfrac{2}{T}\left[\dfrac{z-1}{z+1}\right]}$ ou encore $\boxed{p = \dfrac{2}{T}\dfrac{1-z^{-1}}{1+z^{-1}}}$

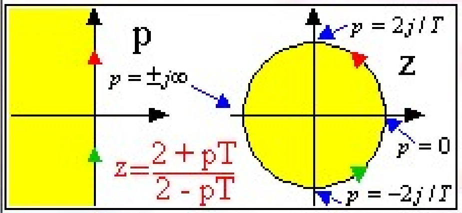
- Une autre façon de justifier la transformation bilinéaire est de la fabriquer à partir d'un système discret réalisant l'**approximation d'une intégrale par la méthode du trapèze** :



  $y[n)] = y[(n-1)T] + T\dfrac{x[nT] + x[(n-1)T]}{2} \Leftrightarrow Y(z) = z^{-1}Y(z) + T\dfrac{X(z) + z^{-1}X(z)}{2}$, qui donne

  $\dfrac{Y(z)}{X(z)} = \dfrac{T}{2}\dfrac{1+z^{-1}}{1-z^{-1}}$; l'équivalent numérique de l'intégrateur analogique $\dfrac{1}{p}$ est donc $\dfrac{T}{2}\dfrac{1+z^{-1}}{1-z^{-1}}$

  En inversant maintenant cette équation, on trouve l'équivalent numérique du dérivateur $\boxed{p = \dfrac{2}{T}\dfrac{1-z^{-1}}{1+z^{-1}}}$, point de départ de la transformation bilinéaire.



La **correspondance graphique entre plan p et plan z** est donnée par $\boxed{z = \dfrac{2+pT}{2-pT}}$

Le point p= 0 se transforme en z = 1, tandis que p= ± ∞ se transforme en z = -1.

La partie stable et négative du plan p a pour image l'intérieur du cercle unité du plan z.

A partir de systèmes continus stables, on obtient donc des systèmes discrets stables .

L'axe imaginaire du plan p se trouvant projeté sur la totalité du cercle unité, cette méthode conduit à une **compression de la bande passante du filtre analogique H(p) vers les basses fréquences** et à un gain nul à la fréquence de Shannon (qui sera matérialisé par un ou plusieurs zéros en -1).