CAPITAN
Guillaume
3A
casier n° 35

Compte-rendu OASIS TP1

Traitement du son

Q2.3 la fonction valconv proposée ne convient pas car, par défaut, Python fera une convolution circulaire. En effet, lorsque idx est trop grand, on peut avoir $n - idx < 0$ et alors au lieu de valoir 0, $u[n-idx]$ prend à nouveau les valeurs des "derniers" indices non nuls de u au lieu de s'arrêter/prendre 0.
Il faut donc délimiter plus précisément l'intervalle de variation de idx.

On écrit idx = np.arange(?, ?) en remplaçant 0 et a par d'autres bornes afin de se débarasser de ces effets de bord.

On doit vérifier $\begin{cases} 0 \leq idx \leq a-1 \\ 0 \leq n - idx \leq b-1 \end{cases}$ ie $n \geq idx \geq n+1-b$

Comme arange exclut le dernier terme du vecteur ligne, on obtient :

$$\boxed{idx = np.arange\left(\max\left(0, n+1-b\right), \min\left(a-1, n\right)+1\right)}$$

Ainsi, $idx \geq n+1-b$ et $idx \geq 0$
et $idx \leq a-1$ et $idx \leq n$.

Q2.4 On applique alors cette nouvelle formule pour le calcul de chacun des termes de la suite résultante de la convolution.

Q2.5 play(x, Fe) écoute le fichier x à la fréquence d'échantillonnage Fe.
play(x, Fe/2) permet donc de jouer le fichier à la moitié de la fréquence d'échantillonnage Fe.

On constate pour le son du piano par exemple que l'on entend une octave plus bas avec $Fe/2$ qu'avec Fe.

Avec quelques tests, en choisissant correctement les valeurs ($2^{\frac{\pm k}{12}} Fe$) on peut jouer toute une gamme.
Le phénomène s'explique très bien par le théorème d'échantillonnage de Nyquist-Shannon que nous verrons par la suite.

• Création d'un écho :

On veut convoler notre signal sonore avec un vecteur d'écho h du type

$$\underbrace{h_0 = 1}_{\text{signal entrée}}, \quad \underbrace{h_{t_1} = 0.5}_{\text{premier écho à } t_1 \text{ échantillons plus tard, d'amplitude } 50\% \text{ de l'entrée}}, \quad \dots \text{ éventuellement } \underbrace{h_{t_2} = 0.1 \quad h_{t_3} =}_{\text{échos multiples comme on peut voir à la fin du colab.py.}}$$

Pour fabriquer un retard de 0.01 s, il faut donc prendre un vecteur avec une valeur non nulle vers 500 (puisque $\underbrace{500}_{\text{nb échantillons}} \times \underbrace{\frac{1}{44100}}_{\text{fréq. échant.}} = 0.01\,s$) qui correspondra à l'atténuation.

Ici, on veut par exemple un écho à 0.01 s d'amplitude 80% de l'entrée.

On aura alors $\boxed{h_0 = 1, \quad h_{500} = 0.8,}$ $h_i = 0$ ailleurs

⚠ Si la fréquence d'échantillonnage avait été plus basse, l'indice du $h_i = 0.8$ (i) aurait été plus bas également.

$h_0 = 1$ entrée, $h_i = q$ avec q l'amplitude relative de l'écho et i le retard en nombre d'échantillons.

Mon programme s'effectue en quelques secondes, entre 4 et une petite dizaine selon les fichiers.
L'utilisation de scipy.convolve réduit ce temps à une durée significativement inférieure à 1 seconde. Cette a dû être optimisée plus que la nôtre, comme on pouvait s'y attendre.

On passe désormais l'écho à 0.2 secondes de décalage de l'entrée, soit $h_{9000} = 0.8$ (car 9000 échantillons de $\frac{1}{44100}$ s. font environ 0.2 s).
Cette fois l'écho est clairement audible après l'entrée. Le son nous "revient" nettement après le premier son.
Cela donne l'impression qu'il revient de plus loin, que le mur qui a servi de réflecteur au son de départ était plus loin que précédemment.

Sachant que la vitesse du son dans l'air est de $340 \, m \, s^{-1}$, on a :

écho : 0.2 s $\Longleftrightarrow$ trajet de 68 m $\Longleftrightarrow$ mur à 34 m

écho : 0.01 s $\Longleftrightarrow$ trajet de 3.4 m $\Longleftrightarrow$ mur à 1,7 m

L'écho de 0.2 s correspond à un mur lointain, ou quand on crie face à une maison / colline. Celui de 0.01 s revient à crier dans sa chambre face à un mur : on n'entend pas tellement d'écho, il se confond avec l'entrée (sans compter les échos multiples, échos d'échos et diffraction...)

1 Q3

Le théorème de Nyquist-Shannon nous dit que l'ensemble des sons que l'on peut décrire avec Fe est l'ensemble des sons de fréquences inférieures à $\frac{Fe}{2}$. Donc en divisant Fe par 2 comme dans cette question, on rejette la limite plus bas et on ne peut plus entendre que les sons de fréquence inférieure à $\frac{Fe}{4}$.
Ceux entre $\frac{Fe}{4}$ et $\frac{Fe}{2}$ ont virtuellement disparu, ils ne sont pas visibles. ()

Dans le premier cas, avec 44100 échantillons par seconde, on entend le son joué à 440 Hz sans problème.
En passant à 22050 Hz, on entend le même fichier audio produire un son à 220 Hz. On passe de La_3 à La_2.

Pour le calculer : on dit que $f/Fe = \nu = 0.01$, c'est la fréquence réduite de l'onde, entre -0.5 et 0.5.
Mais ce qu'on entend dépend de la fréquence d'échantillonnage Fe et est obtenu par la formule $f = Fe * \nu$. (ν se lisait à 0.01 sur le TP, et f se calcule ainsi)

on obtient alors $f_1 = Fe_1 * \nu = 44100 * 0{,}01 = 441$ Hz

et $f_2 = Fe_2 * \nu = 22050 * 0{,}01 = 220$ Hz

Ce sont les fréquences perçues.

Q4 Filtres rejecteurs

le fonctionnement de la fonction lfilter est le suivant :
Supposons que l'on appelle lfilter(L_1, L_2, L_3) (il y a deux autres arguments facultatifs en rapport avec des applications multiples du filtre)
dans ce cas lfilter renverra L_3 filtrée
par le filtre $\frac{L_1}{L_2}$ (L_1 liste des coefficients au numérateur,
L_2 ——— au dénominateur)

On appelle souvent lfilter sur la sortie que l'on vient de filtrer pour appliquer justement plusieurs filtres consécutifs.

(A la fin, cela donne un seul gros filtre du style $\frac{L_{11} L_{12} L_{13} \cdots}{L_{21} L_{22} L_{23} \cdots}$)

la RIF est $\frac{L_1}{L_2}$, L_3 est le signal à convoluer contre.

la fréquence de l'onde parasite est (lecture TP) $\boxed{f = 1261 \text{ Hz}}$ et comme $Fe = 44100$ Hz
on en déduit la fréquence réduite $f/Fe = \boxed{\nu = 0.0286} \in [-\frac{1}{2}; \frac{1}{2}[$

rejette 1 : filtre une première fois avec une convolution avec $[1, -z_0]$
puis une seconde fois ———— $[1, -\overline{z_0}]$

On a donc $H_1(z) = 1 - z_0 z^{-1}$

et $H_2(z) = 1 - \overline{z_0} z^{-1}$

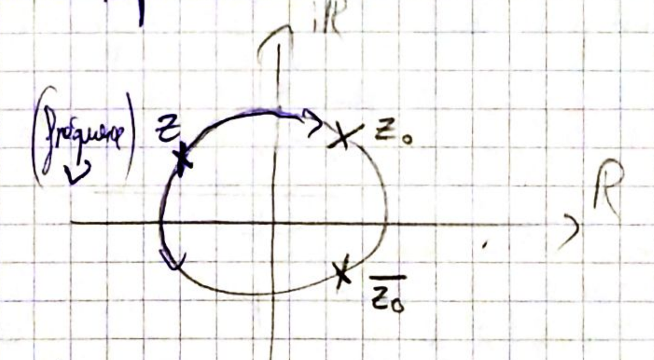
soit $\boxed{H_{rejette_1}(z) = 1 - 2\,\mathrm{Re}(z_0) z^{-1} + |z_0|^2 z^{-2}}$

Les zéros sont clairement z_0 et $\overline{z_0}$, et pas de dénominateur
donc pas de pôles ⟶ on le voit d'ailleurs au 1 dans l'appel à lfilter.
(lfilter([1, -z0], 1, entree)).

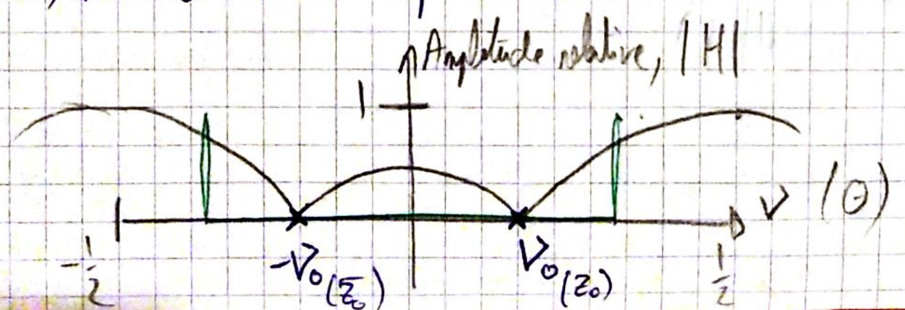
Ce rejecteur-ci est, à l'oreille, plutôt mou et n'accomplit pas tellement ce que l'on veut. Certes, il coupe le bruit parasite à 1261 Hz mais il coupe aussi une très large bande de fréquences autour de la parasite.
Bien sûr, $z_0 = e^{2\pi \frac{f_0}{F_e}}$ avec le f_0 proche de la fréquence à éliminer

sur le cercle unité, donc le filtre fonctionne, mais il rejette trop le reste, et le son est finalement trop atténué.

Plan complexe :



quand z se rapproche de l'un ou l'autre des zéros, il y a atténuation du signal, mais elle arrive trop vite…



Toute la plage de fréquences en vert est atténuée (grossièrement) rejette 2 filtre une première fois en convoluant avec $\dfrac{[1, -z_0]}{[1, -\rho z_0]}$

puis une seconde fois avec $\dfrac{[1, -\overline{z_0}]}{[1, -\rho \overline{z_0}]}$

Ce qui donne comme filtres intermédiaires

$$H_1(z) = \frac{1 - z_0 z^{-1}}{1 - \rho z_0 z^{-1}}$$ zéro z_0, pôle ρz_0

et $$H_2(z) = \frac{1 - \overline{z_0} z^{-1}}{1 - \rho \overline{z_0} z^{-1}}$$ zéro $\overline{z_0}$, pôle $\rho \overline{z_0}$

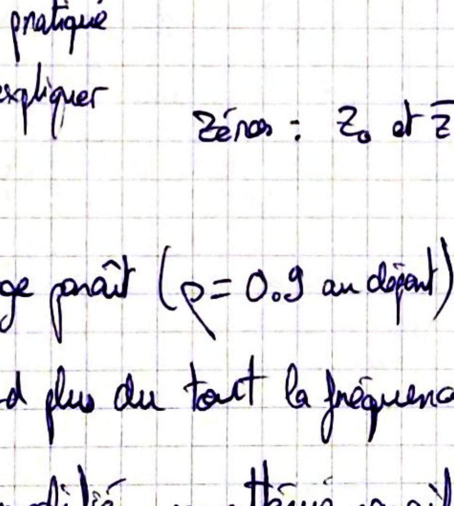
soit $$\boxed{H_{rejette2}(z) = \frac{(1 - z_0 z^{-1})(1 - \overline{z_0} z^{-1})}{(1 - \rho z_0 z^{-1})(1 - \rho \overline{z_0} z^{-1})}} = \frac{1 - 2\mathrm{Re}(z_0) z^{-1} + |z_0|^2 z^{-2}}{1 - 2\rho \mathrm{Re}(z_0) z^{-1} + \rho^2 |z_0|^2 z^{-2}}$$

forme pratique pour expliquer

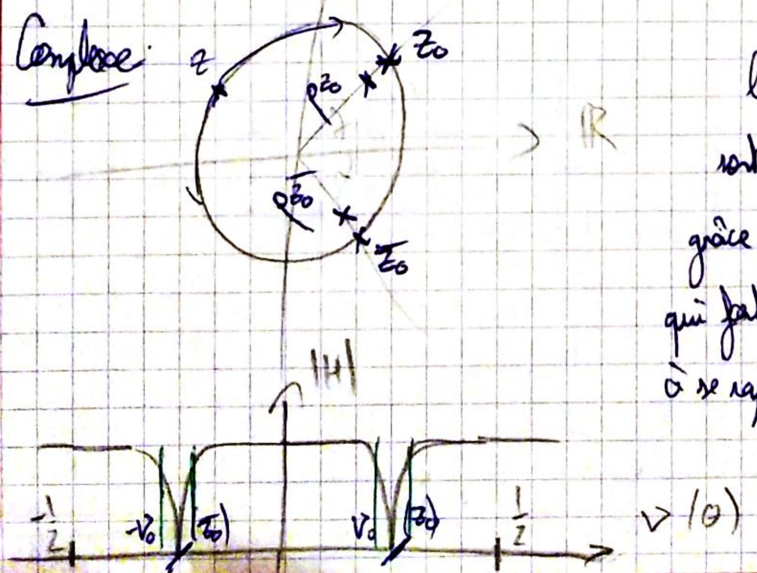
zéros : z_0 et $\overline{z_0}$, pôles : ρz_0 et $\rho \overline{z_0}$.

Ce filtrage paraît ($\rho = 0.9$ au départ) bien plus satisfaisant que l'autre : on n'entend plus du tout la fréquence parasite et le reste du son semble très peu modifié, pas atténué par ailleurs. Le rejeteur semble précis.

Complexe :



les pentes des zéros sont beaucoup plus abruptes grâce à la présence des pôles, qui font diverger quand on commence à se rapprocher de z_0.



Plages atténuées bien plus restreintes, précises

Pour justifier la construction du module de la TFED du filtre 2 :

A la différence du filtre 1 dont le module fait des vagues douces vers 0 puisqu'il n'y a qu'un numérateur, le filtre 2 a des pôles.

Lorsque la fréquence $\nu(z)$ est loin de la fréquence qu'on cherche à couper ν_0 (z_0) comme ρ est proche de 1, les distances entre z et z_0 et z et ρz_0 se valent. Quand z commence à approcher z_0, le numérateur se rapproche de 0 à la même vitesse que le dénominateur se rapproche de $+\infty$ (module) à cause du pôle en $\rho z_0 \simeq z_0$. Cependant quand $z \longrightarrow z_0$, très proche, alors la distance entre z et z_0 devient négligeable devant la distance entre z et ρz_0 et alors $|H| \longrightarrow 0$.

Autrement dit, $|H| \xrightarrow[z \to z_0]{} 0$ car alors $d(z, z_0) \ll d(z, \rho z_0)$

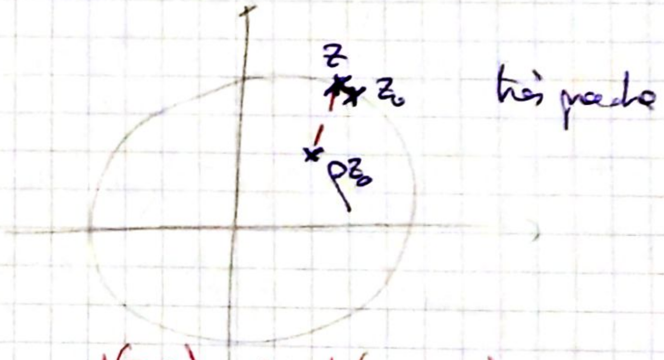
et sinon $|H| \simeq 1$ car $d(z, z_0) \simeq d(z, \rho z_0)$

même distances environ

↳ pas d'atténuation : OK

(mieux que filtre 1)

$|H| \simeq 1$



$d(z, z_0) \ll d(z, \rho z_0)$

même si $\rho \simeq 1$ car on peut toujours se rapprocher plus de la limite.

Alors $|H| \simeq 0$

Cette idée d'avoir $d(z, z_0) \simeq d(z, \rho z_0)$ quand z n'est pas trop proche de z_0 demande d'avoir ρ proche de 1, vraiment proche.

(et toujours < 1)

Bref, le filtre 2 est bien meilleur que le filtre 1.

si $\rho > 1$, on va violer la condition selon laquelle le filtre doit être causal (ie pas de dépendance du passé)

et alors il y aura une explosion numérique dans les valeurs du signal après filtrage. On explose l'atrée.

Quel compromis alors pour ρ ? On sait que $\rho \lneq 1$.

- Plus on rapproche ρ de 1, plus on gagne en précision sur le rejet.
(les pics deviennent de plus en plus fins dans le graphe de $|H|$)

cependant en le rapprochant trop, bien qu'on élimine alors moins de fréquences non parasites, on devient tellement précis qu'on rejette <u>à côté</u> de la fréquence voulue.
Le filtre est en effet à 1267 Hz pour un parasite à 1261 Hz.

Pour $\rho = 0,9$, c'est parfait, le filtre est bon.

Mais pour $\rho = 0,99$, on commence à entendre un léger bruit gênant,

et pour $\rho = 0,999$ on ne l'atténue plus du tout. On rejette trop précis.

Pour $\rho = 0,7$ le filtre ressemble au filtre 1, à atténuer trop de valeurs.

Bref, entre $\rho = 0,95$ et $\rho = 0,92$, cela semble bien.

C'est ça le problème d'être trop sélectif : si on ne se place pas parfaitement sur le bruit, on peut finir par passer à côté

- Autre raison de pas être trop proche de 1 : la rapidité de la décroissance exponentielle (qui est en $(\rho z_0)^{-n}$ donc plus ρz_0 est proche de 1, plus cette décroissance sera lente !)