UNIVERSITÉ DE SHERBROOKE
Faculté de génie
Département de génie électrique et génie informatique

PLAN DE TEST

Traitement de signal sur µC
APP6

Présenté à
Paul Charette

Présenté par
Équipe numéro 4
Xavier Boudreau – BOUX2901
Anthony Royer – ROYA2019

Sherbrooke - 26 juillet 2022

PRÉFACE

(Caractère : 11 ou 12 | Arial, Time Roman, Calibri ou Cambria | Justifié | Interligne 1,5)

TABLE DES MATIÈRES

LISTE DES FIGURES

LISTE DES TABLEAUX

No table of figures entries found.

Python conception des filtres

-filtre fir passe bas

- Description : conception d'un filtre fir passe bas 500 Hz dans python
- Résultat attendu, conditions du test (fréquence du signal d'entrée, etc.): fc=500Hz, fe=20000Hz, N=256, fenêtrage blackman et fonction signal.firwin

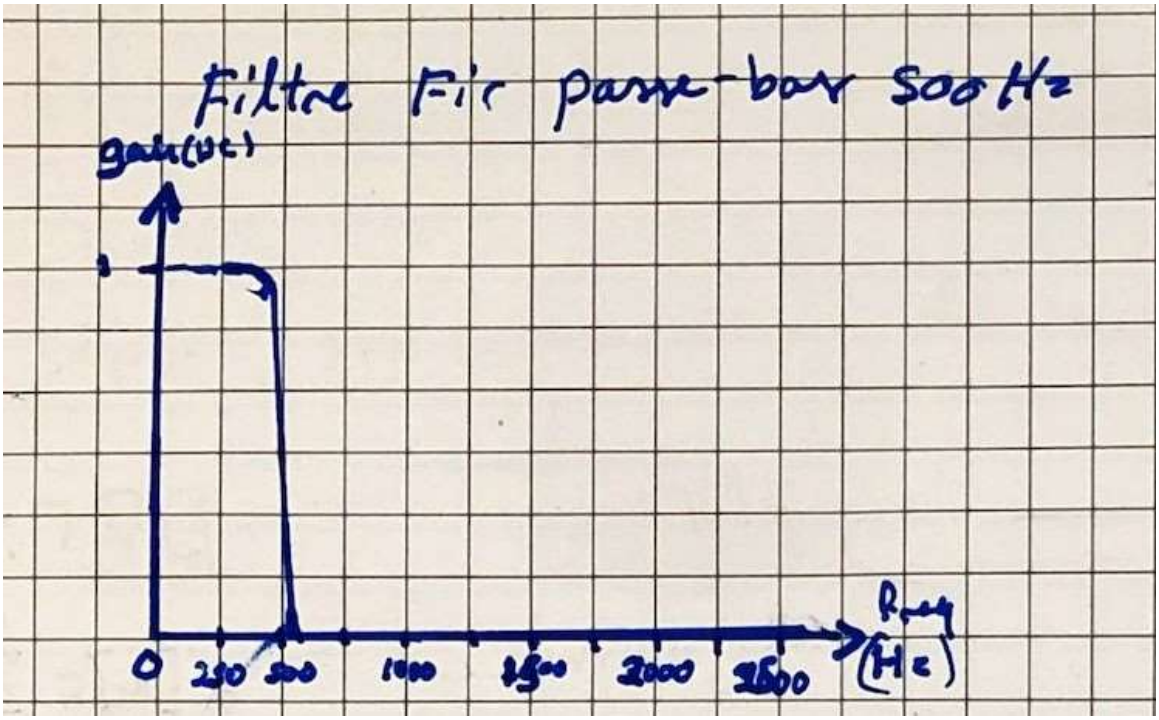


Figure 1:filtre passe-bas attendu

- Test unitaire à exécuter : effectuer un fft, un np.abs, un fftshift et puis l'afficher avec plt.plot

- Résultat obtenu :

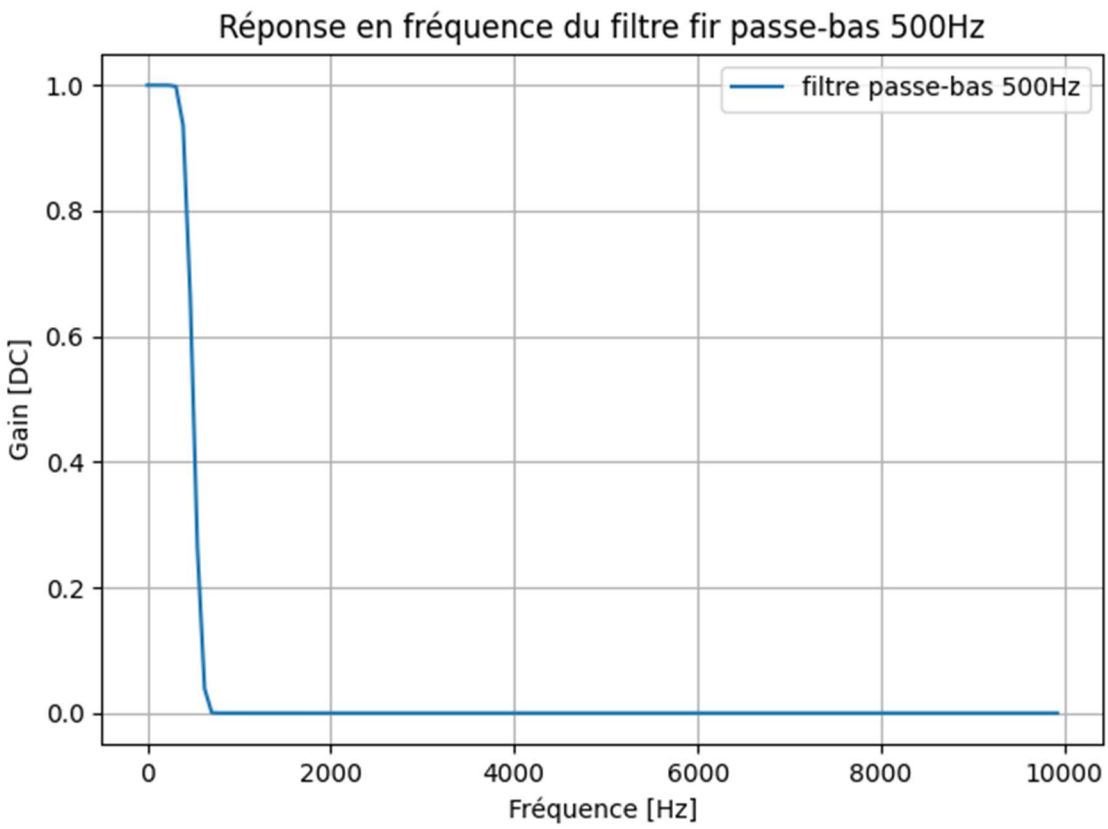


Figure 2 : filtre passe-bas obtenu, python

-filtre fir passe haut

- Description : conception d'un filtre fir passe-haut 4490 Hz dans python

- Résultat attendu, conditions du test (fréquence du signal d'entrée, etc.): fc=4490Hz, fe =20000Hz, N=256, fenêtrage blackman et fonction signal.firwin

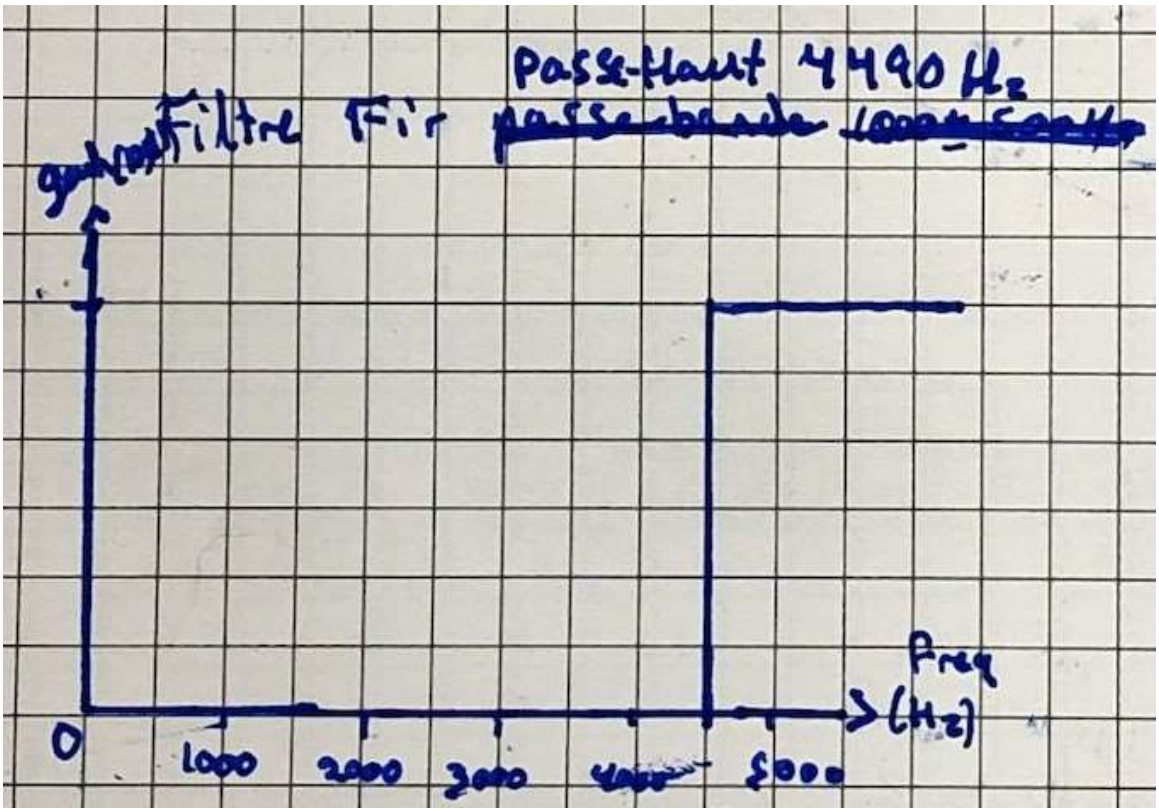


Figure 3 : filtre passe haut attendu

- Test unitaire à exécuter : effectuer un fft, un np.abs, un fftshift et puis l'afficher avec plt.plot

- Résultat obtenu :

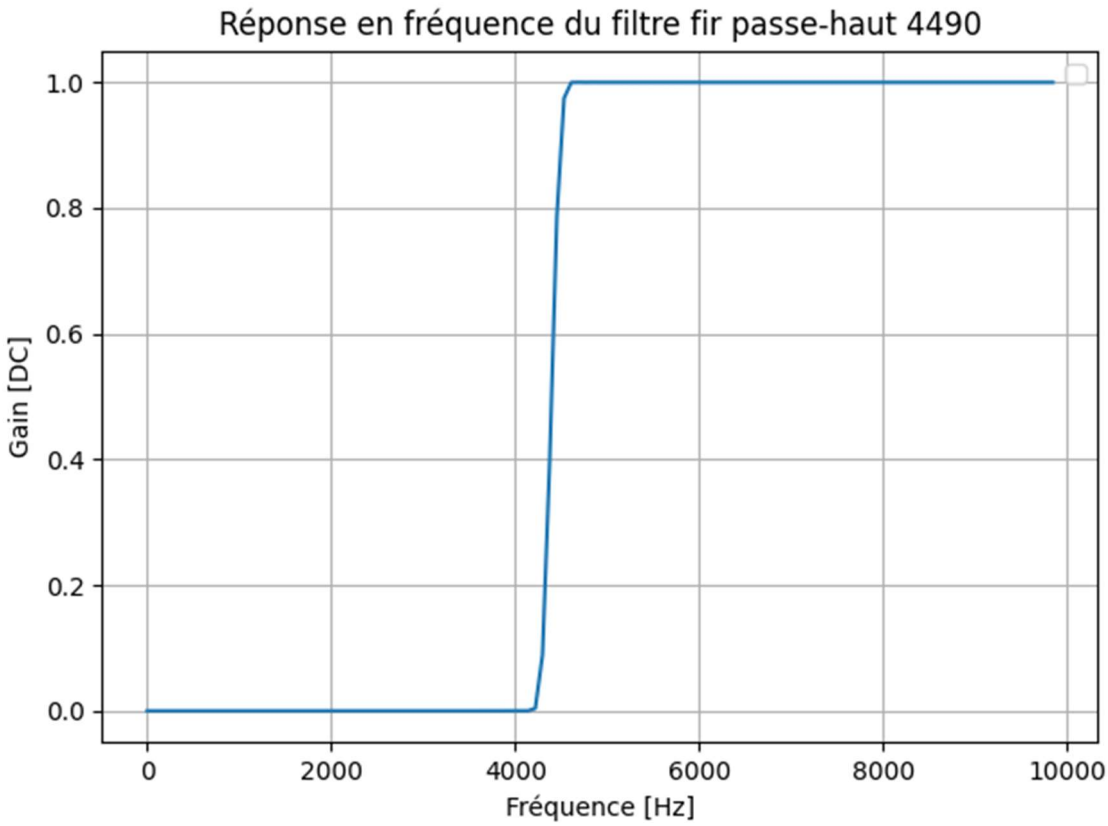


Figure 4: filtre passe-haut obtenu, python

-Filtre fir passbande1

- Description : conception d'un filtre fir passe-bande 1000±500 Hz dans python

- Résultat attendu, conditions du test (fréquence du signal d'entrée, etc.): fc_low=500Hz, fc_high=1500Hz fe =20000Hz, N=256, fenêtrage blackman et fonction signal.firwin

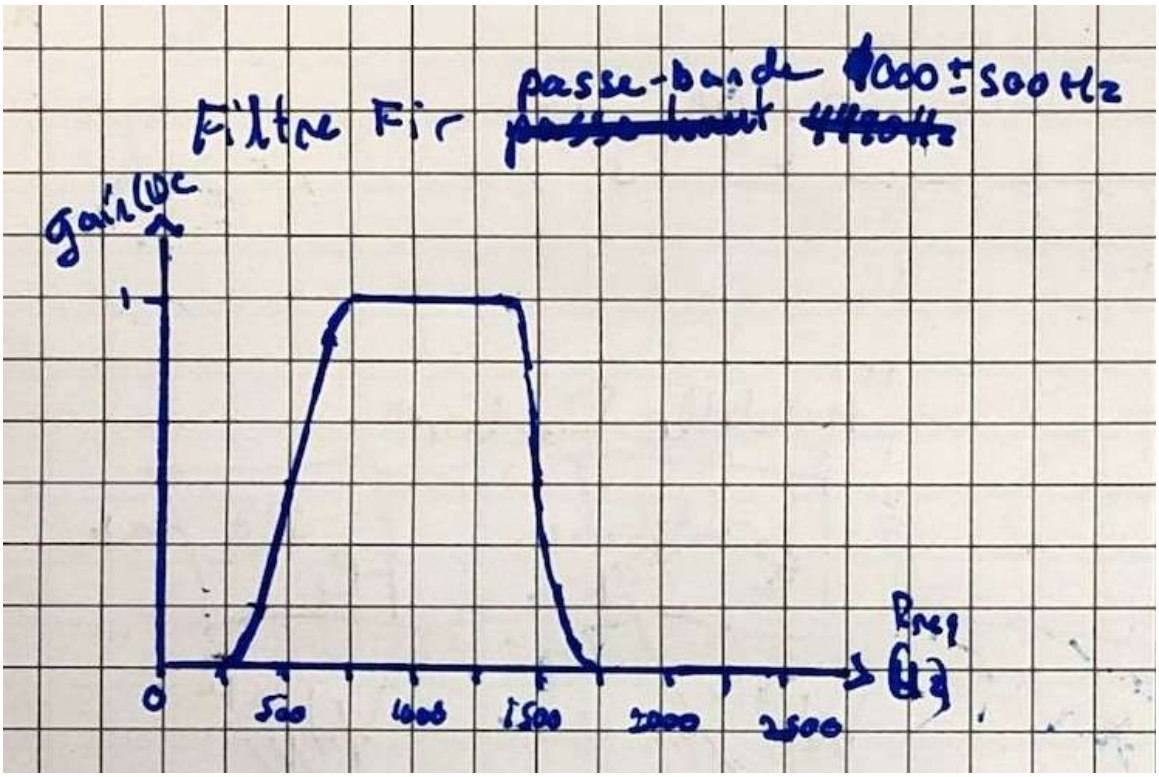


Figure 5:filtre passe bande attendue

- Test unitaire à exécuter : effectuer un fft, un np.abs, un fftshift et puis l'afficher avec plt.plot

- Résultat obtenu :

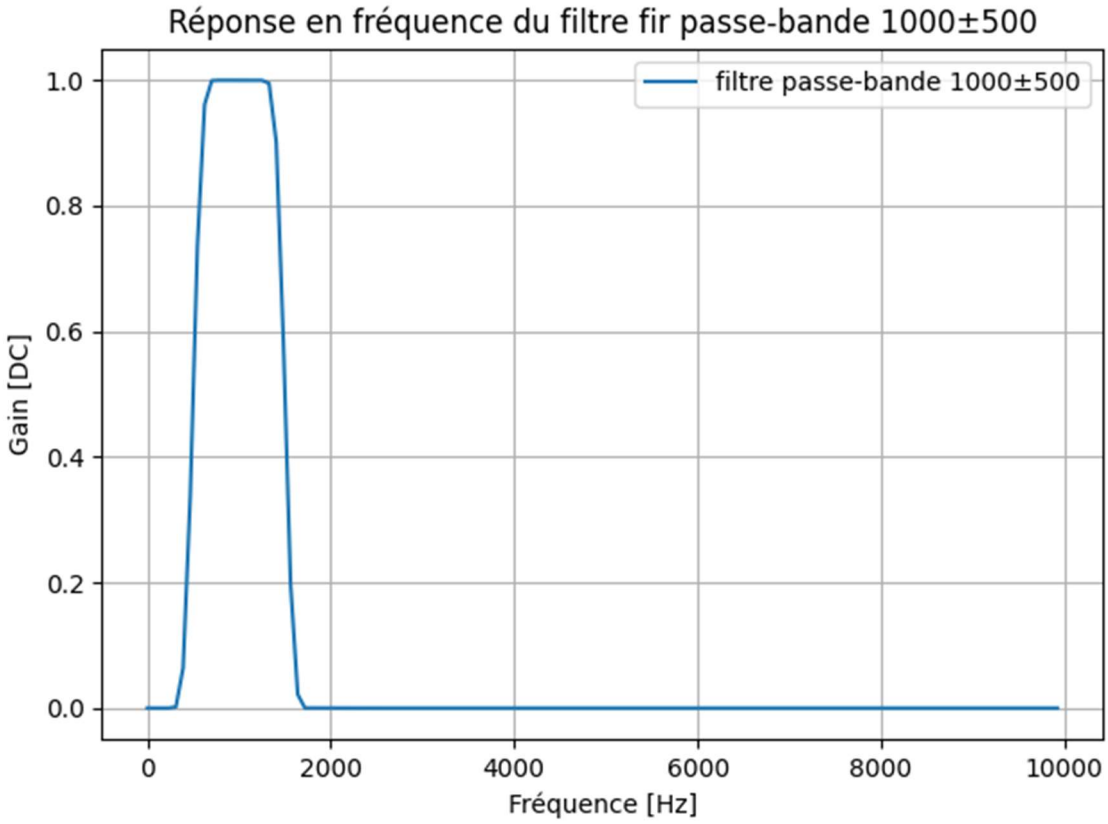


Figure 6: filtre passe-bande obtenu, python

-Filtre fir passbande2

- Description : conception d'un filtre fir passe bas 2000±500 Hz dans python

- Résultat attendu, conditions du test (fréquence du signal d'entrée, etc.): fc_low=1500Hz, fc_high=2500Hz fe =20000Hz, N=256, fenêtrage blackman et fonction signal.firwin

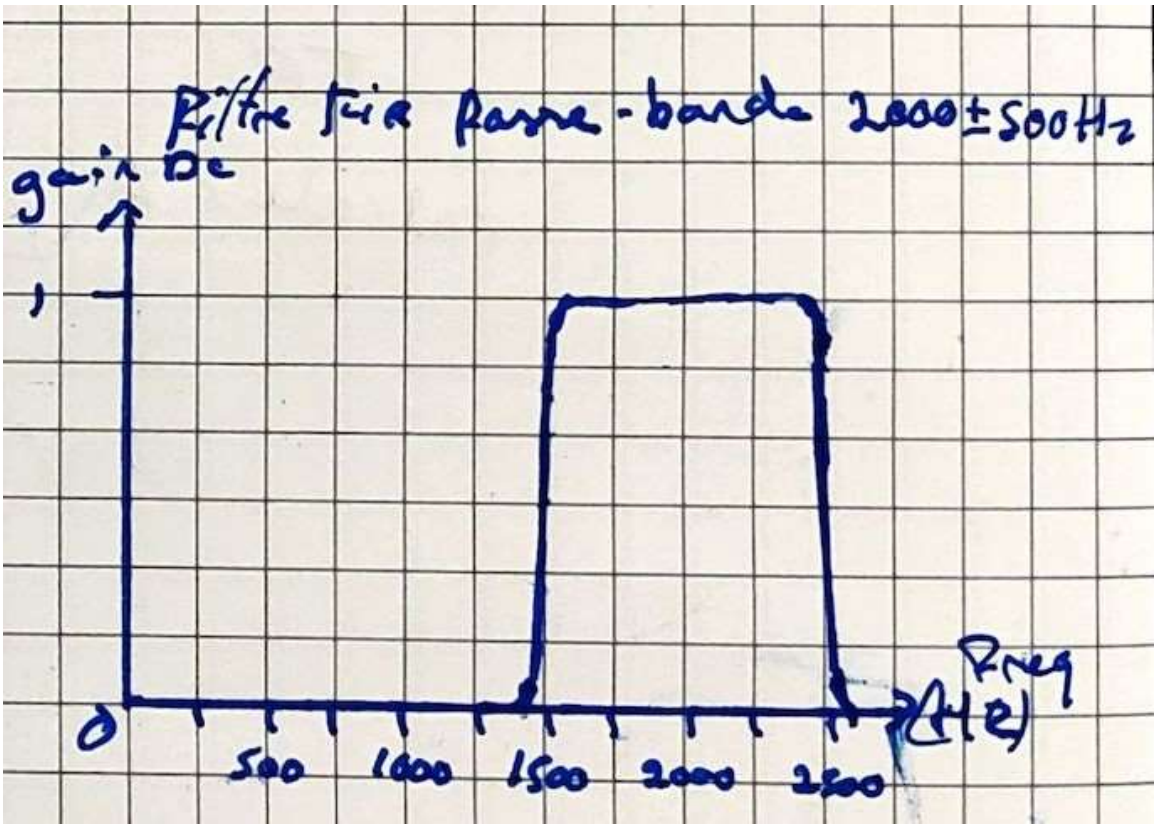


Figure 7:filtre passe bande attendu

- Test unitaire à exécuter : effectuer un fft, un np.abs, un fftshift et puis l'afficher avec plt.plot

- Résultat obtenu :

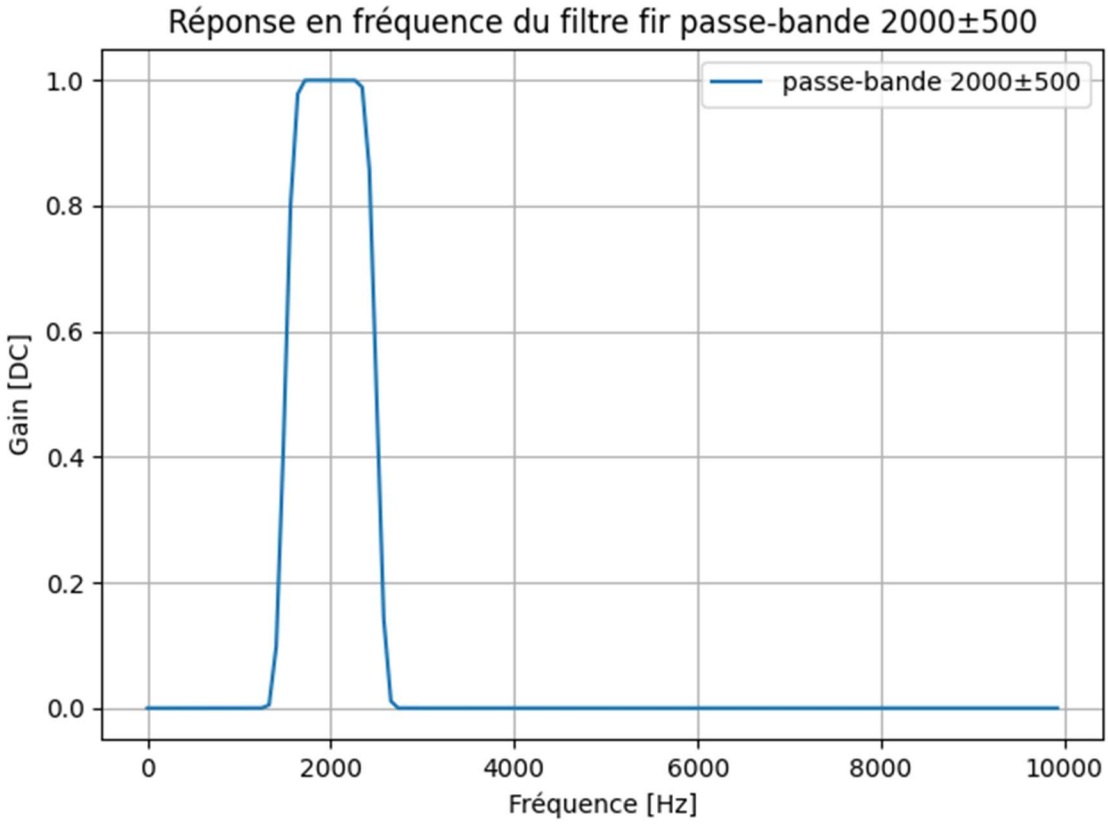


Figure 8 : filtre passe-bande obtenu, python

-Filtre fir passbande3

- Description : conception d'un filtre fir passe bas 3500±1000 Hz dans python

- Résultat attendu, conditions du test (fréquence du signal d'entrée, etc.): fc_low=2500Hz, fc_high=4500Hz fe =20000Hz, N=256, fenêtrage blackman et fonction signal.firwin

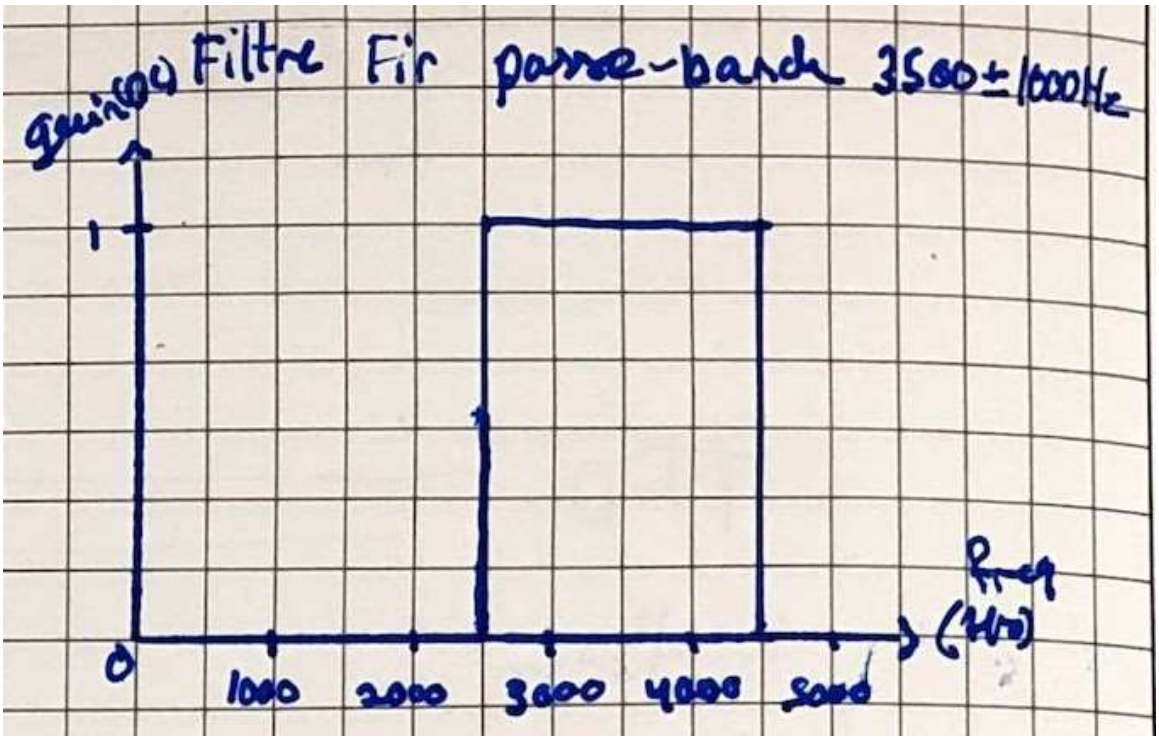


Figure 9:filtre passe bande attendu

- Test unitaire à exécuter : effectuer un fft, un np.abs, un fftshift et puis l'afficher avec plt.plot

- Résultat obtenu :

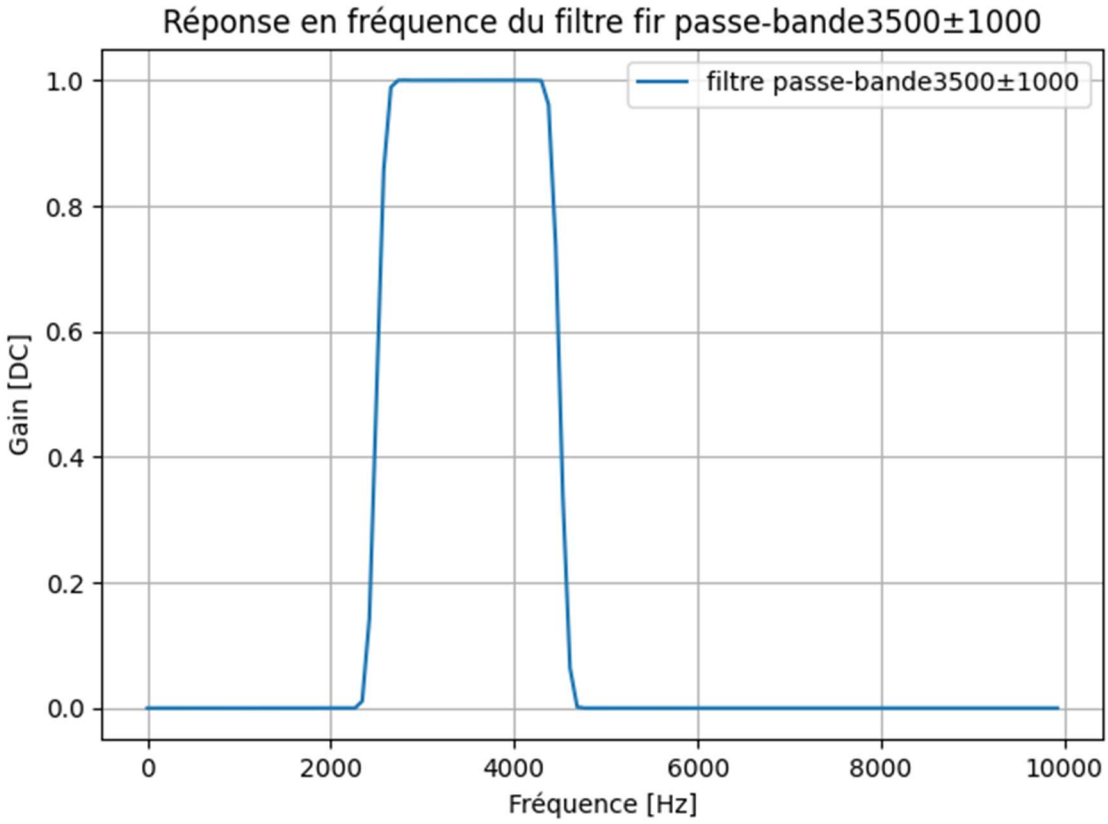


Figure 10: filtre passe-bande obtenu, python

-Filtre iir coupe bande

- Description : conception d'un filtre fir coupe bande 1000±50 Hz dans python

- Résultat attendu, conditions du test (fréquence du signal d'entrée, etc.): fc_low=950Hz, fc_high=1050Hz fe =20000Hz, N=256, fenêtrage blackman et fonction signal.ellip

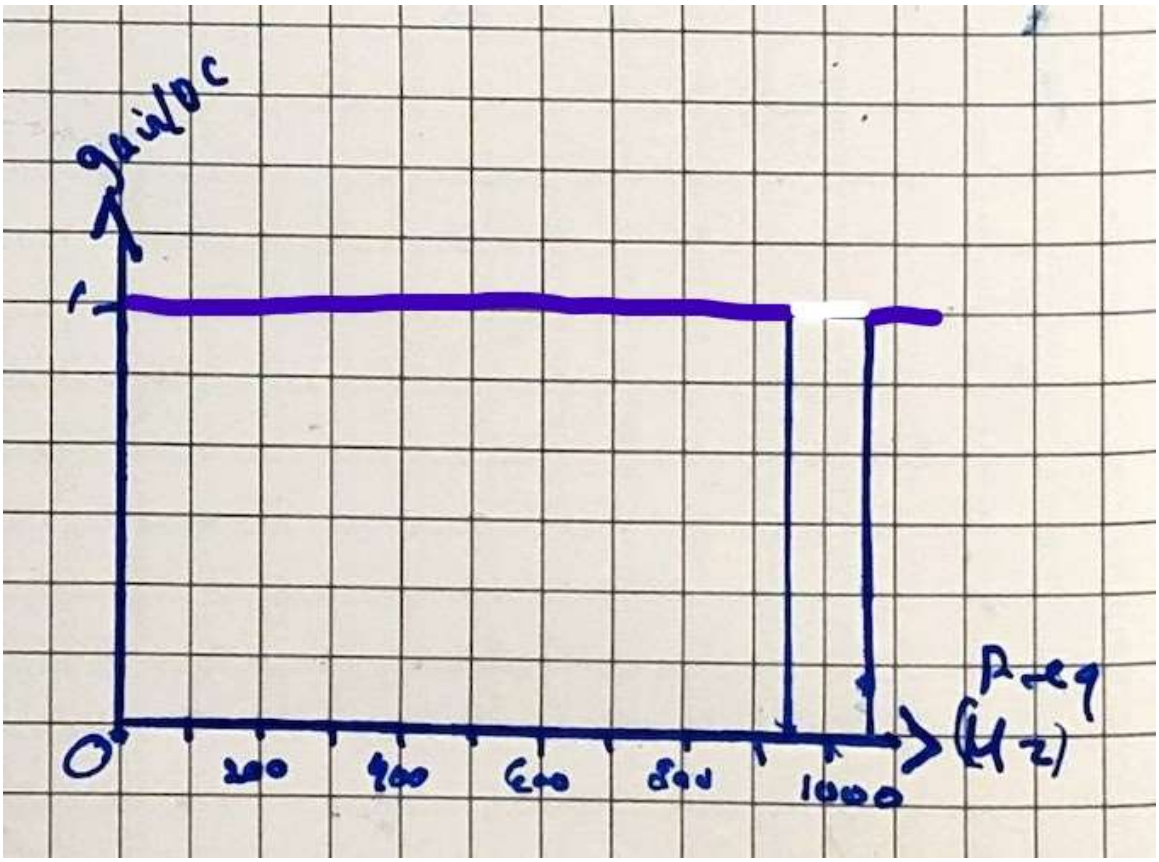


Figure 11:filtre coupe-bande attendu

- Test unitaire à exécuter : effectuer un plt.semilogx
- Résultat obtenu :

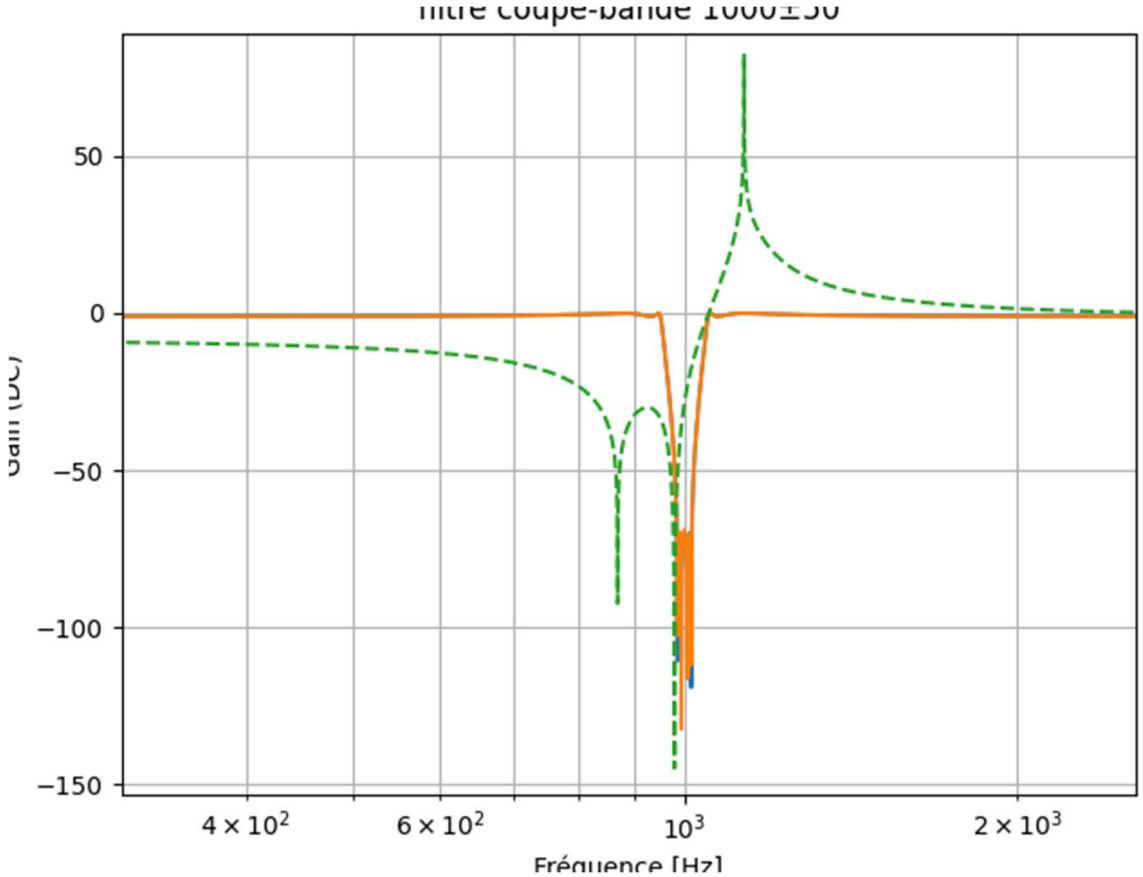


Figure 12:filtre coupe-bande obtenu

A1 - CALCUL DE LA RÉSOLUTION FRÉQUENTIELLE DU SPECTRE

DESCRIPTION

Calcul et affichage de de la valeur de la variable *spectralResolution (double)*

$$df = fe/N$$

RÉSULTAT ATTENDU

$\Delta f \approx 19$

TEST UNITAIRE À EXÉCUTER

Abaisser SW0 et regarder le résultat à l'écran du calcul

RÉSULTAT OBTENU

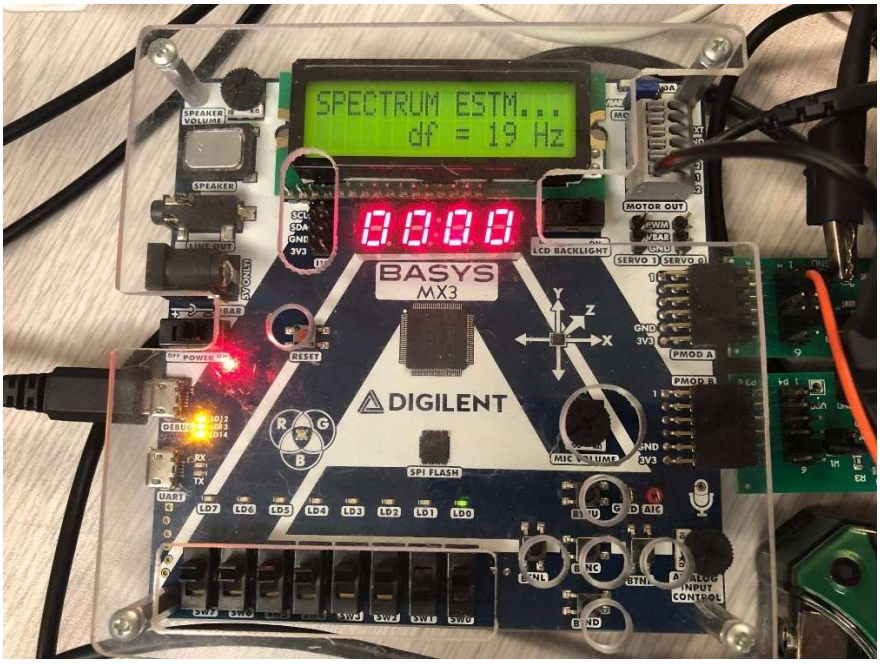


Figure 13: résultat démontrant la réussite du test unitaire A2

A2 & A3 – CALCUL DU SPECTRE D'ENTRÉE, EFFET DU FENÊTRAGE

DESCRIPTION

Calcul de la TF du signal d'entrée avec la fonction mips32_fft(), calcul du module carré du spectre |X[k]|2 en dB, stockage du résultat dans le tableau debugBuffer1, affichage avec DMCI.

RÉSULTATS ATTENDUS

Le graphique attendu dans DMCI est le même que celui obtenu dans python ci-dessous

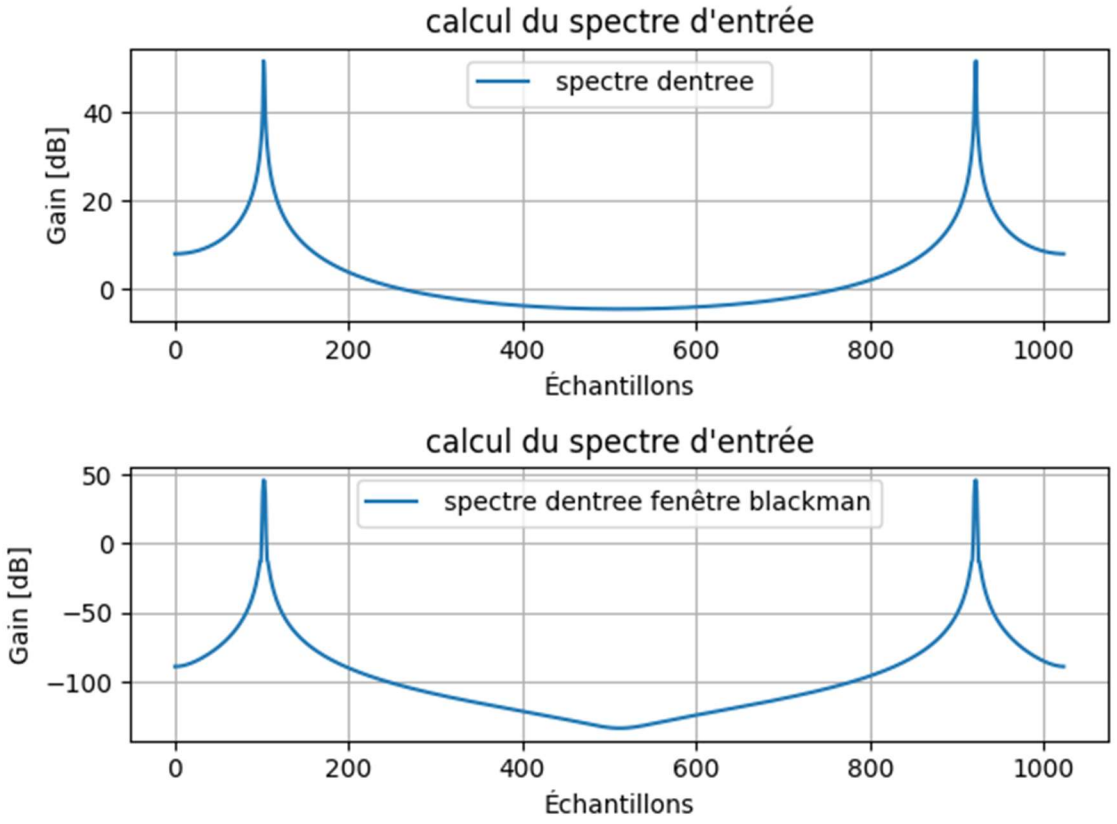


Figure 14:spectre du signal d'entrée

TEST UNITAIRE

Aller en DMCI et vérifier que le graphique de debugBuffer1 est le même que le graphique de python pour un signal de 2000Hz en entrée.

RÉSULTAT

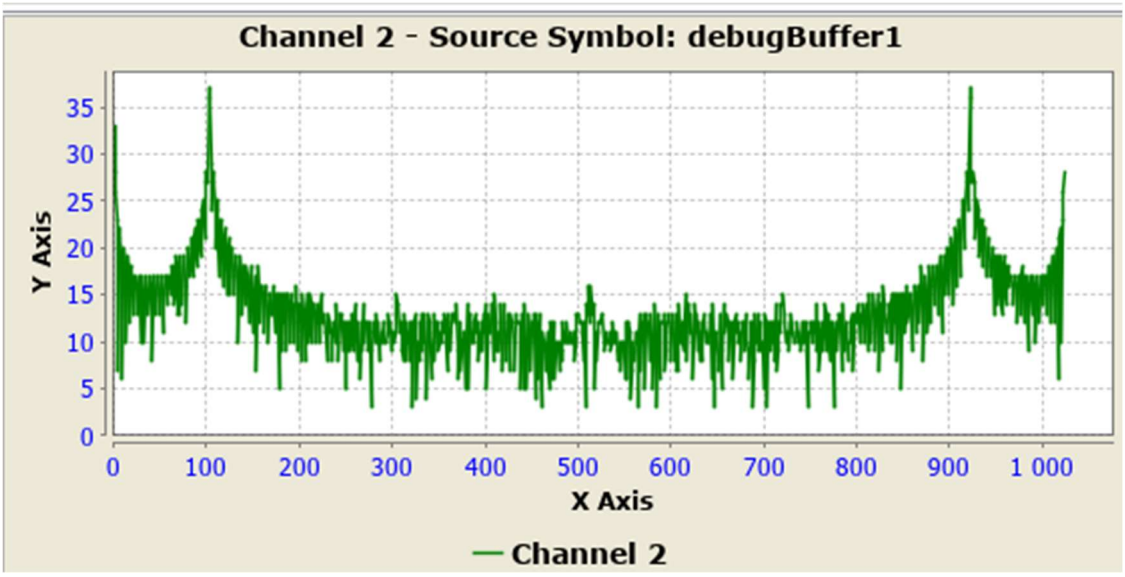


Figure 15:spectre du signal d'entrée DMCI

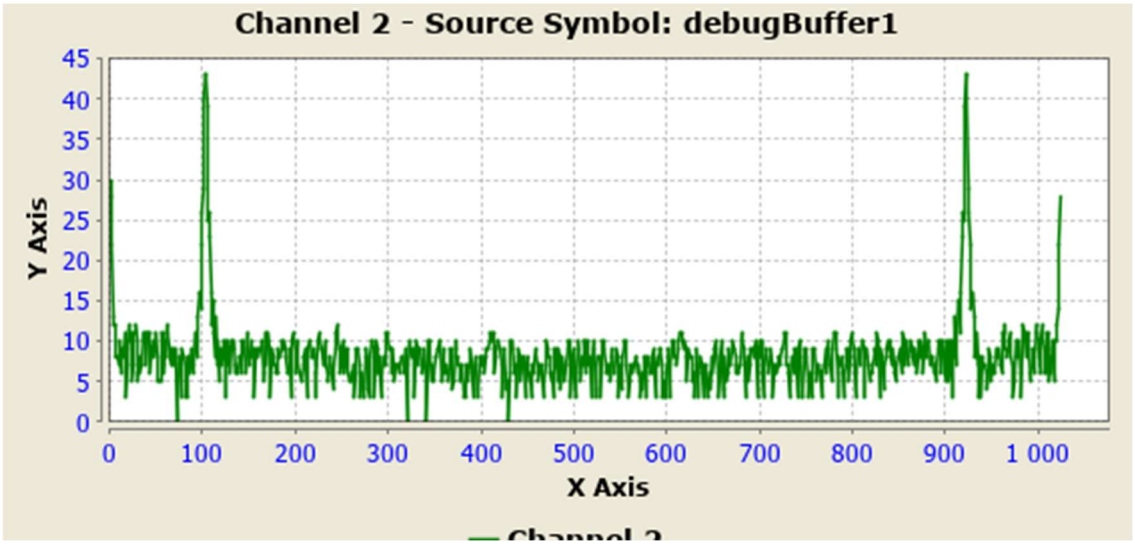


Figure 16:spectre du signal d'entrée fenêtrer DMCI

A4 – Calcul de l'indice de la fréquence à amplitude maximale

Description

Calcul de la valeur de la variable maxAmplFreq (type Int) afin d'obtenir l'indice de la fréquence avec l'amplitude maximal en Hz.

Résultat attendu, conditions du test (fréquence du signal d'entrée, etc.)

Le résultat affiché à l'écran est de 1992. Avec un signal d'entrée de 2000Hz

Test unitaire à exécuter

Envoyer un signal de 200Hz en entrée et, en mode DMCI, aller chercher la valeur de l'indice le plus haut (sinon, pour être plus précis, on peut exporter le buffer en csv aller chercher sa valeur de cette manière).

Résultat obtenu

L'écran affiche 1992

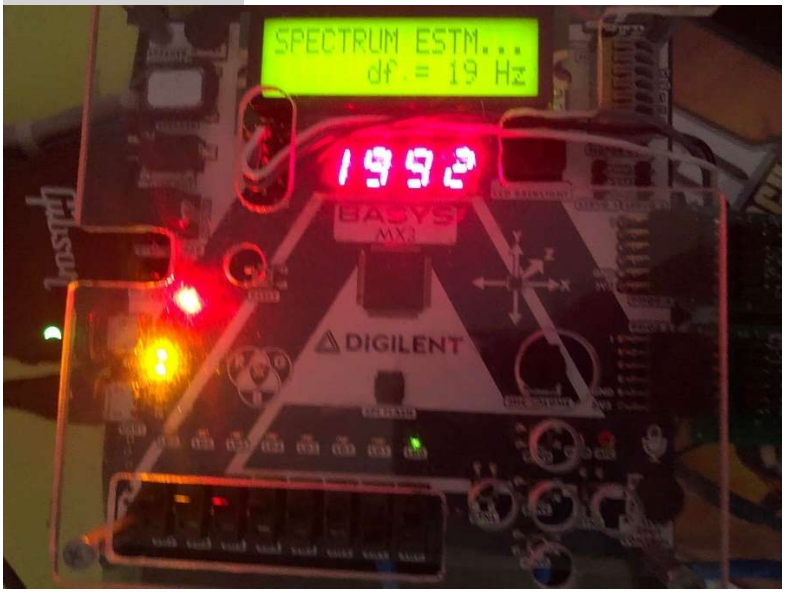


Figure 17:affichage de la bonne valeur sur la MX3

B0 B1 – Calcul de la FFT du signal d'entrée rallongé à 4N (méthode overlap & Save)

Description

Signal d'entrée rallongé à 4N dans le tableau inFFT(type : int32c, longueur : 4N)

Calcul de la FFT avec la fonction mips32_fft() dans le tableau outFFT(type : int32c, longueur : 4N)

Résultats attendus, conditions du test

En DMCI, le graphique pour debugBuffer1 (A2 et A3) et le debugBuffer2 seront identiques.

Test unitaire à exécuter

Afficher en mode DMCI le infft avec un signal d'entrée de 2000Hz.

Résultat obtenu

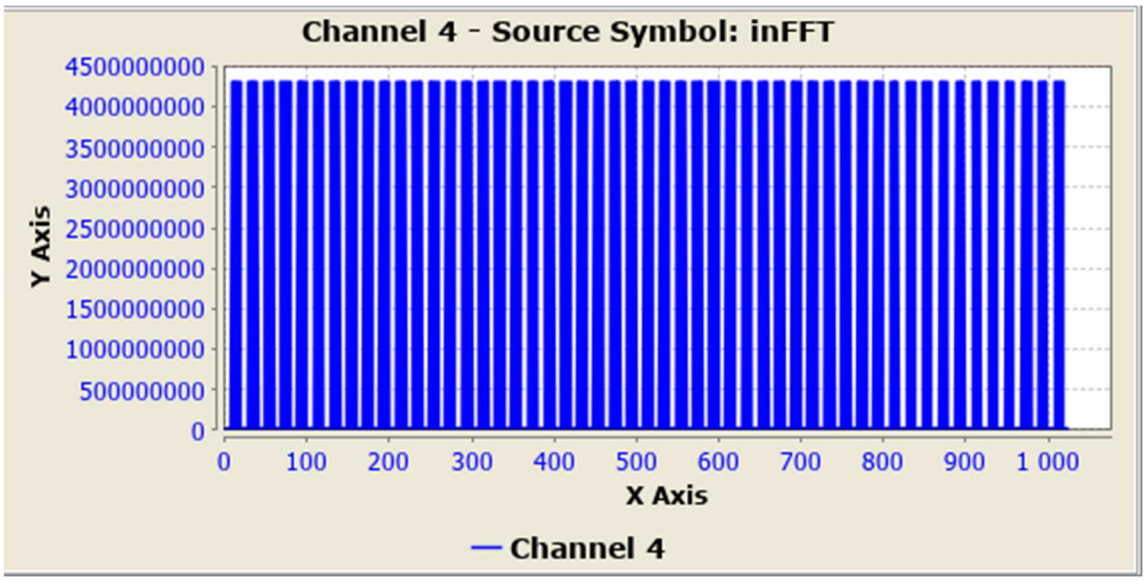


Figure 18:infft obtenu dans DMCI

B2 – FILTRAGE DANS LE DOMAINE FRÉQUENTIEL

DESCRIPTION

Filtrage dans le domaine fréquentiel par la multiplication des transformées de Fourier

RÉSULTATS ATTENDUS, CONDITIONS DU TEST

Filtre passe bas 500Hz: faire passer des signaux sinusoïdaux de 300Hz et 1000Hz séparément. Seul le signal a 300Hz devrait restera intact

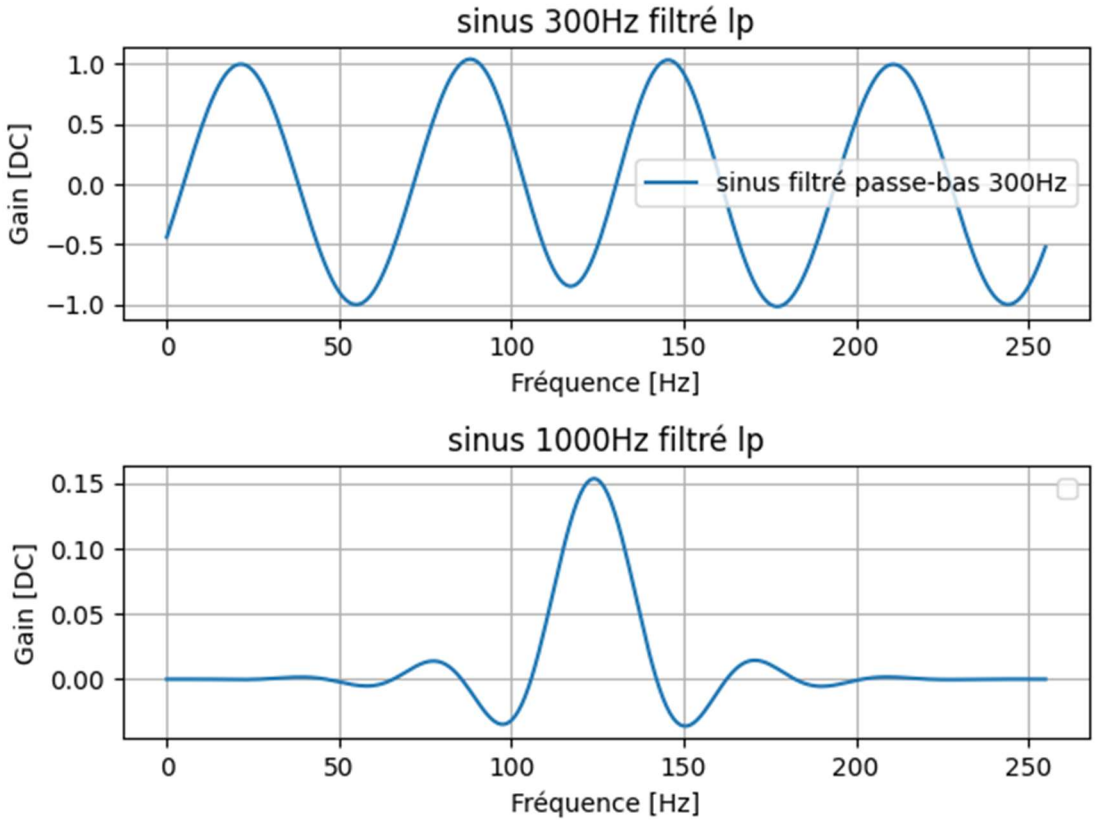


Figure 19:filtrage de deux sinus par le filtre passe bas fait dans python

Filtre passe haut 4490Hz: faire passer des signaux sinusoïdaux de 4000Hz et 5000Hz séparément. Seul le signal a 5000Hz devrait restera intact

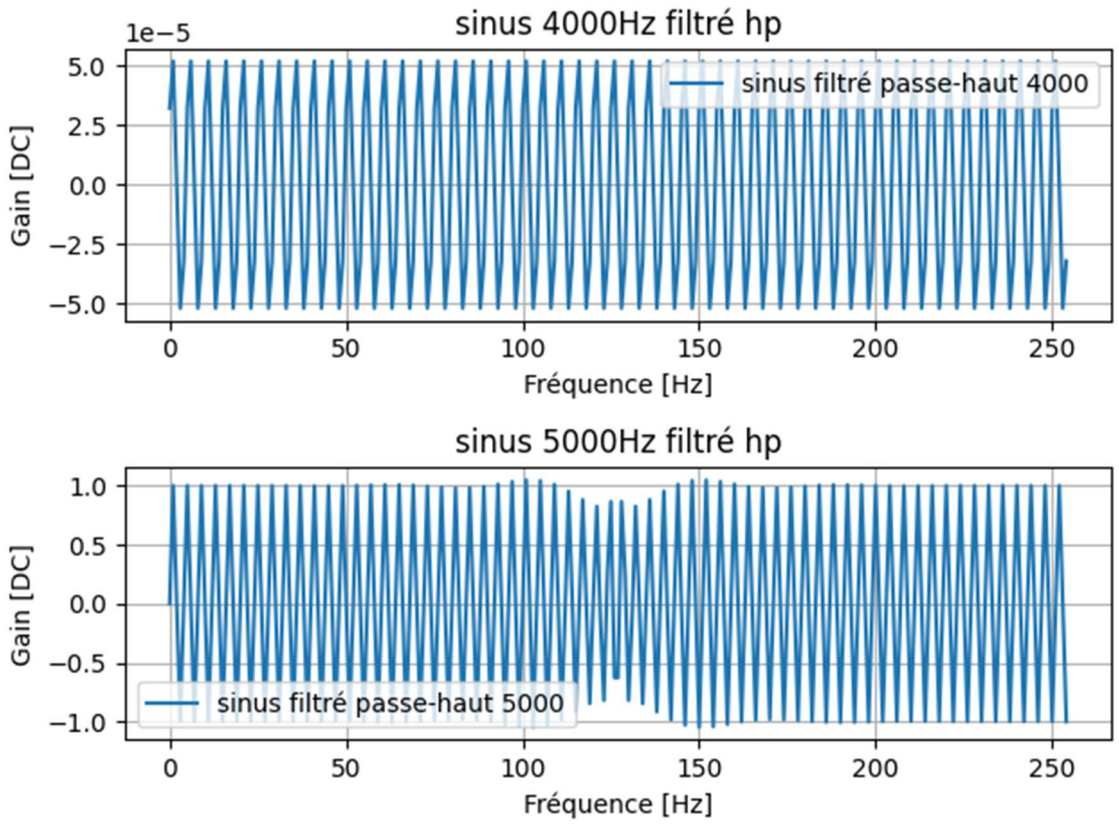


Figure 20filtrage de deux sinus par le filtre passe haut fait dans python

Filtre passe bande 1000+-500Hz: faire passer des signaux sinusoïdaux de 200Hz, 1000Hz et 2000Hz séparément. Seul le signal a 1000Hz devrait restera intact

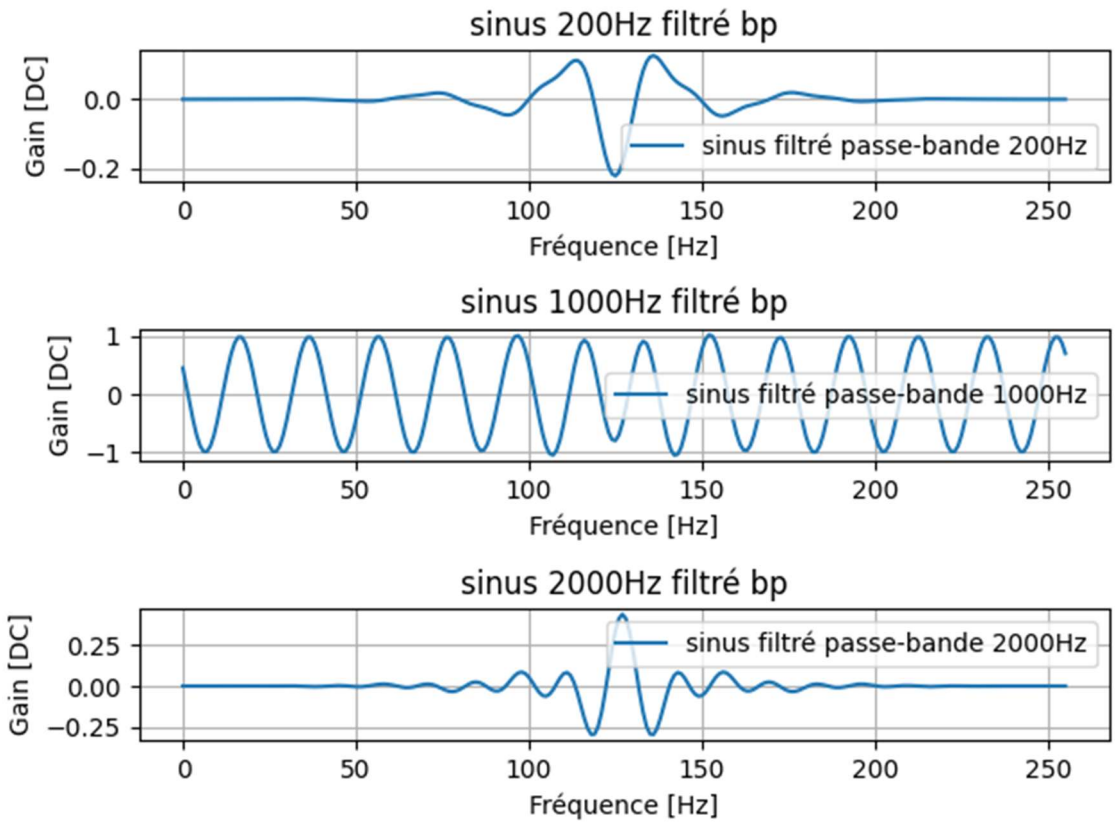


Figure 21 :filtrage de trois sinus par le filtre passe bande 1000+-500 fait dans python

Filtre passe bande 2000+-500Hz: faire passer des signaux sinusoïdaux de 1000Hz, 2000Hz et 3000Hz séparément. Seul le signal a 2000Hz devrait restera intact

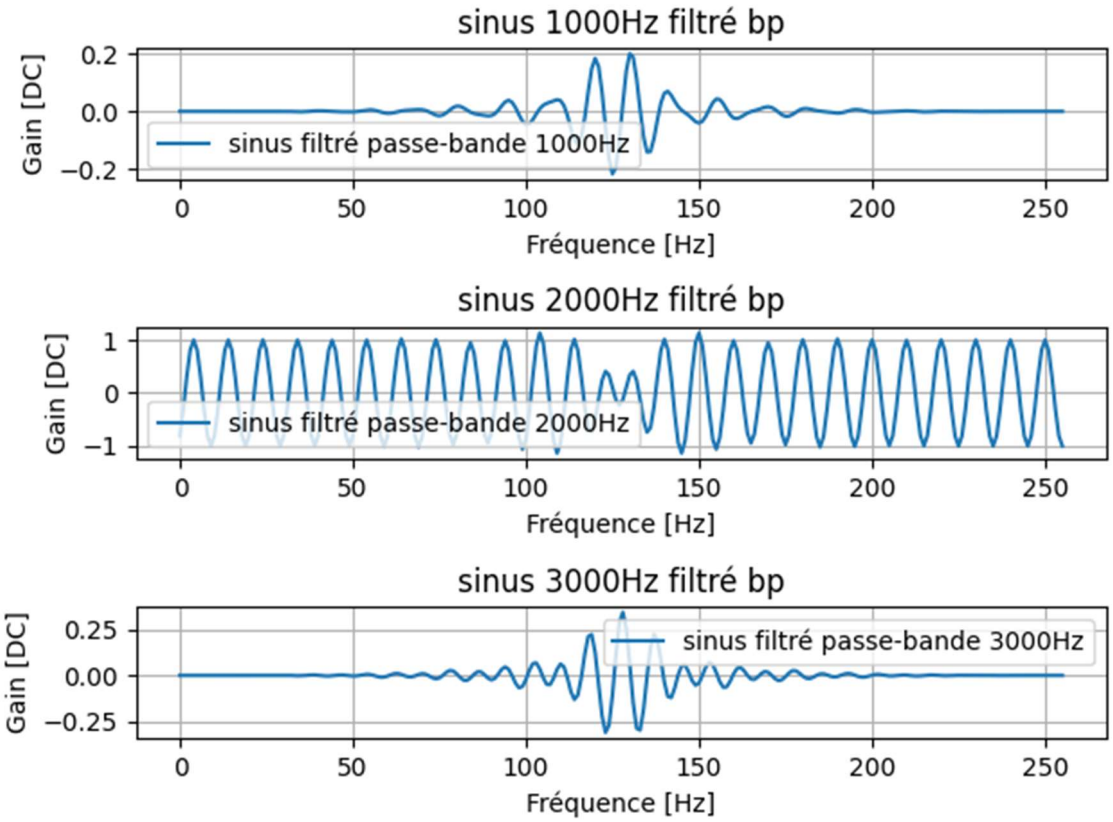


Figure 22:filtrage de trois sinus par le filtre passe bande 2000+-500 fait dans python

Filtre passe bande 3500+-5500Hz: faire passer des signaux sinusoïdaux de 3500Hz, 5500Hz et 2000Hz séparément. Seul le signal a 3500Hz devrait restera intact

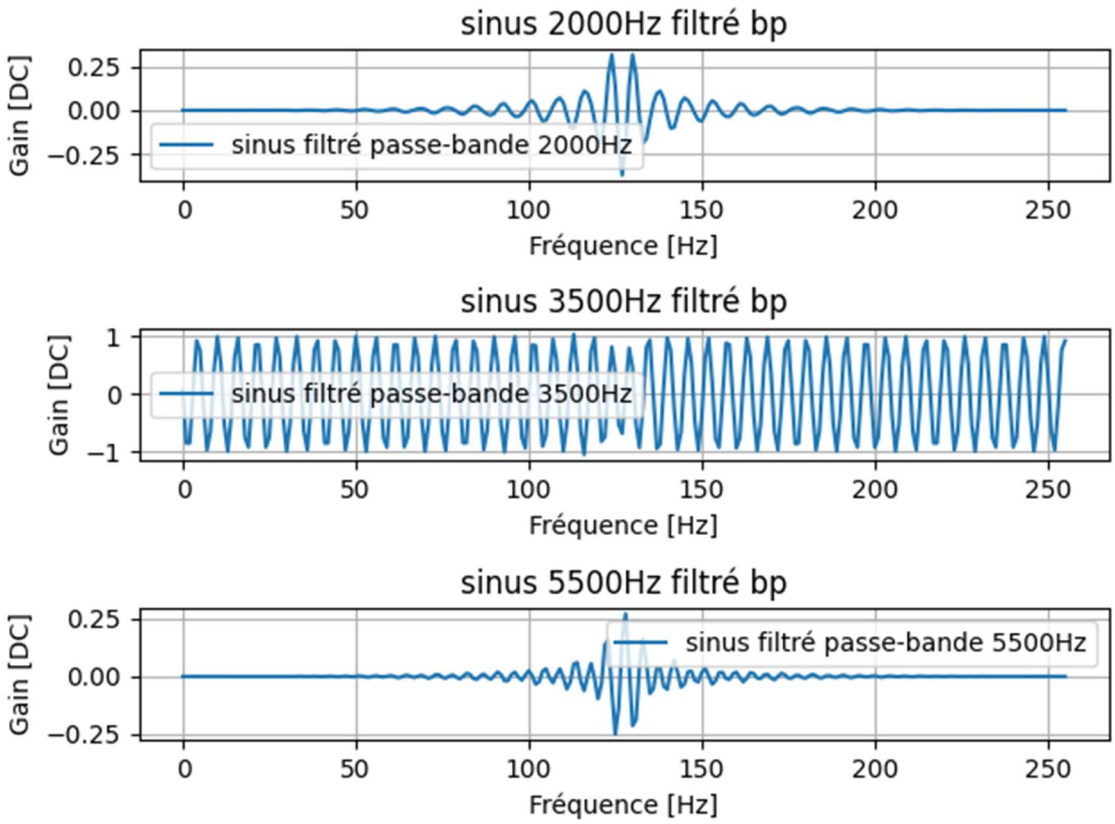


Figure 23:filtrage de trois sinus par le filtre passe bande 3500+-1000 fait dans python

TEST UNITAIRE À EXÉCUTER

Créer un graphique de réponse impulsionnelle pour chaque filtre sur python et générer un graphique en MPlab pour comparaison (voir graphiques en python pour référence).

RÉSULTAT OBTENU

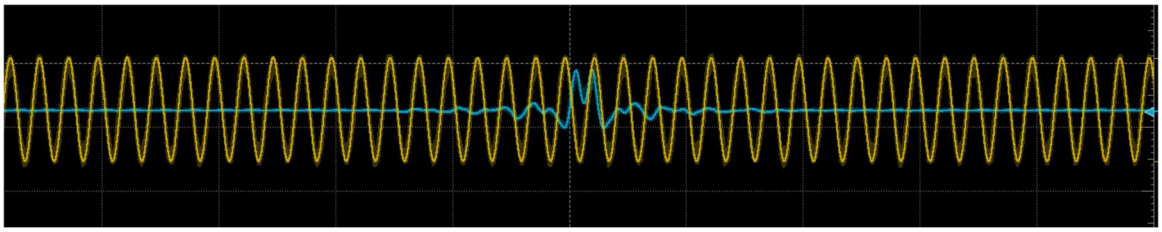


Figure 24signal 1000Hz

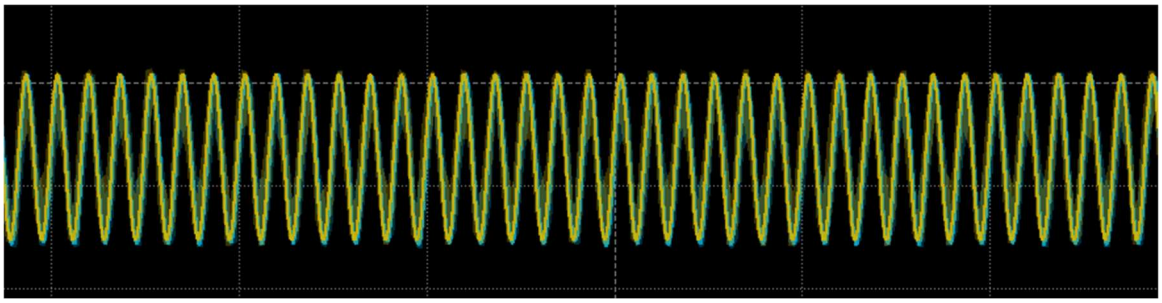


Figure 25signal 2000Hz

B3 & B4 – FFT INVERSE, EXTRACTION DE LA TRAME FILTRÉE

DESCRIPTION

Faire une iFFT avec la fonction de FFT (mips32_fft) et en extraire les paramètres complexes et réels (garder seulement réel en Q15) et l'exporter dans notre buffer de sortie précédent.

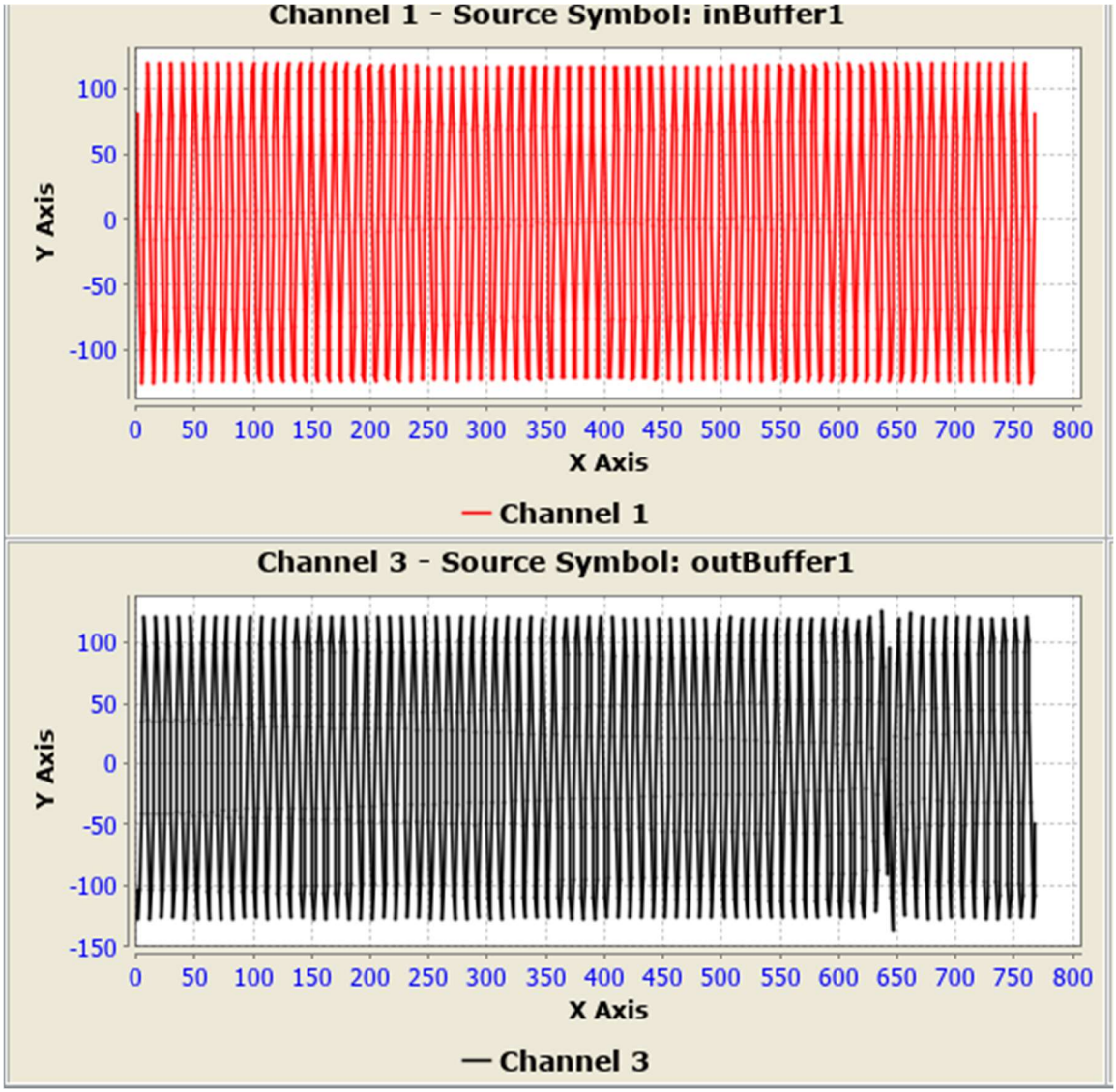
RÉSULTATS ATTENDUS, CONDITIONS DU TEST

Envoyer un signal de 2000Hz avec SW5 activé.

TEST UNITAIRE À EXÉCUTER

Envoyer un signal de 2000Hz et afficher le buffer dans le DMCI et comparer les résultats.

RÉSULTAT OBTENU



C1 – FILTRES IIR

DESCRIPTION

Fabriquer le filtre avec les équations des filtres IIR et ensuite convertir la sortie en Q0.15

RÉSULTATS ATTENDUS, CONDITIONS DU TEST

Lorsqu'on passe un signal de fréquence inférieur à la bande de coupure (ex : 400Hz), ou supérieur à la plage (ex : 2000Hz), celui-ci laisserai passer les signaux. Si le signal d'entrée sera atténué si celui-ci se trouve dans la plage de coupure (ex : 1000Hz).

-voire figure 11

TEST UNITAIRE À EXÉCUTER

Envoyer 3 signaux d'entré aux fréquences indiqués plus haut.

RÉSULTAT OBTENU

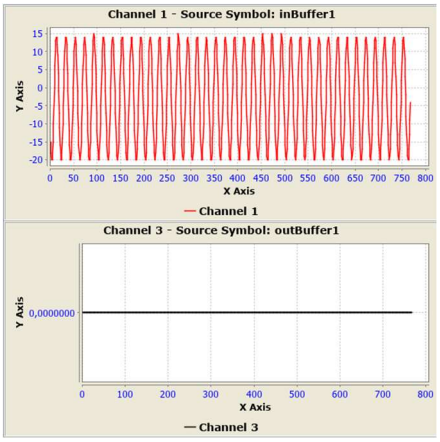
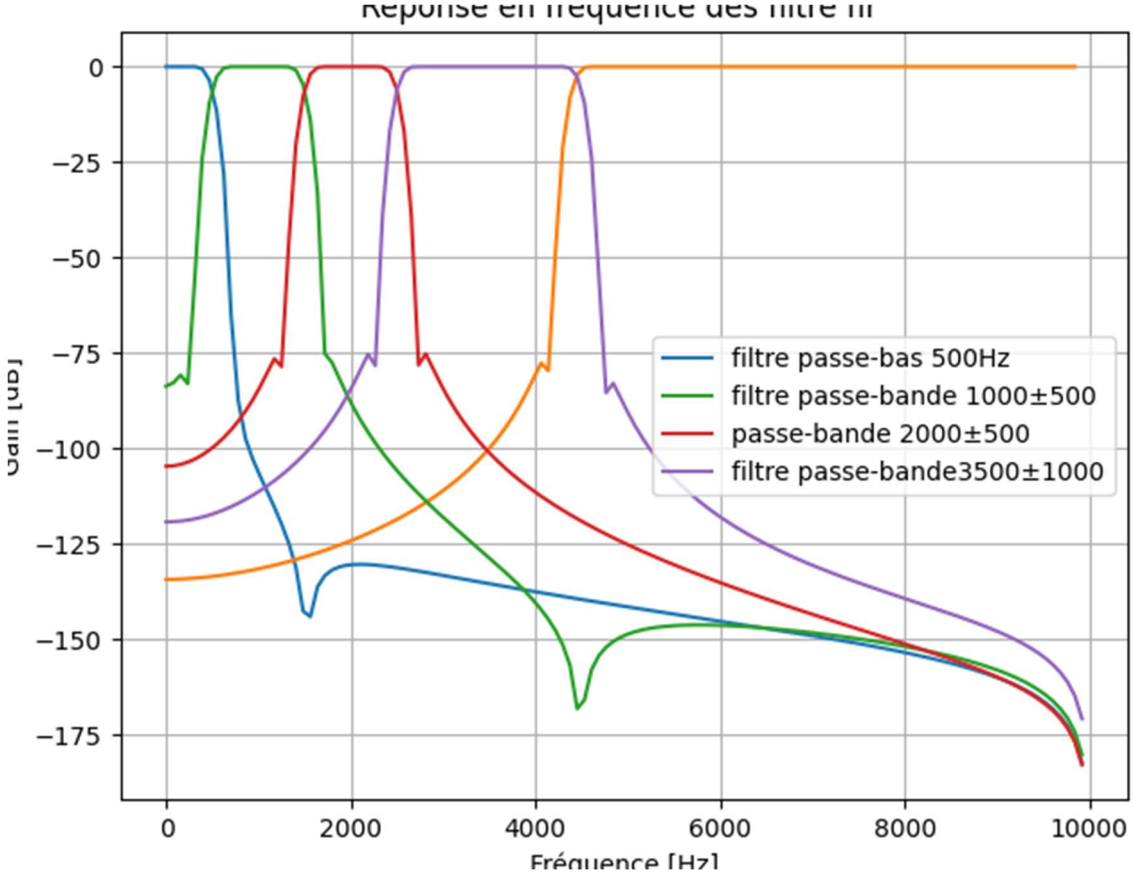


Figure 26 Filtre IIR et un signal de 1kHz

ANNEXE


Réponse en fréquence des filtre IIR
0
−25
−50
−75
−100
−125
−150
−175
Gain [dB]
0
2000
4000
6000
8000
10000
Fréquence [Hz]
filtre passe-bas 500Hz
filtre passe-bande 1000±500
passe-bande 2000±500
filtre passe-bande3500±1000

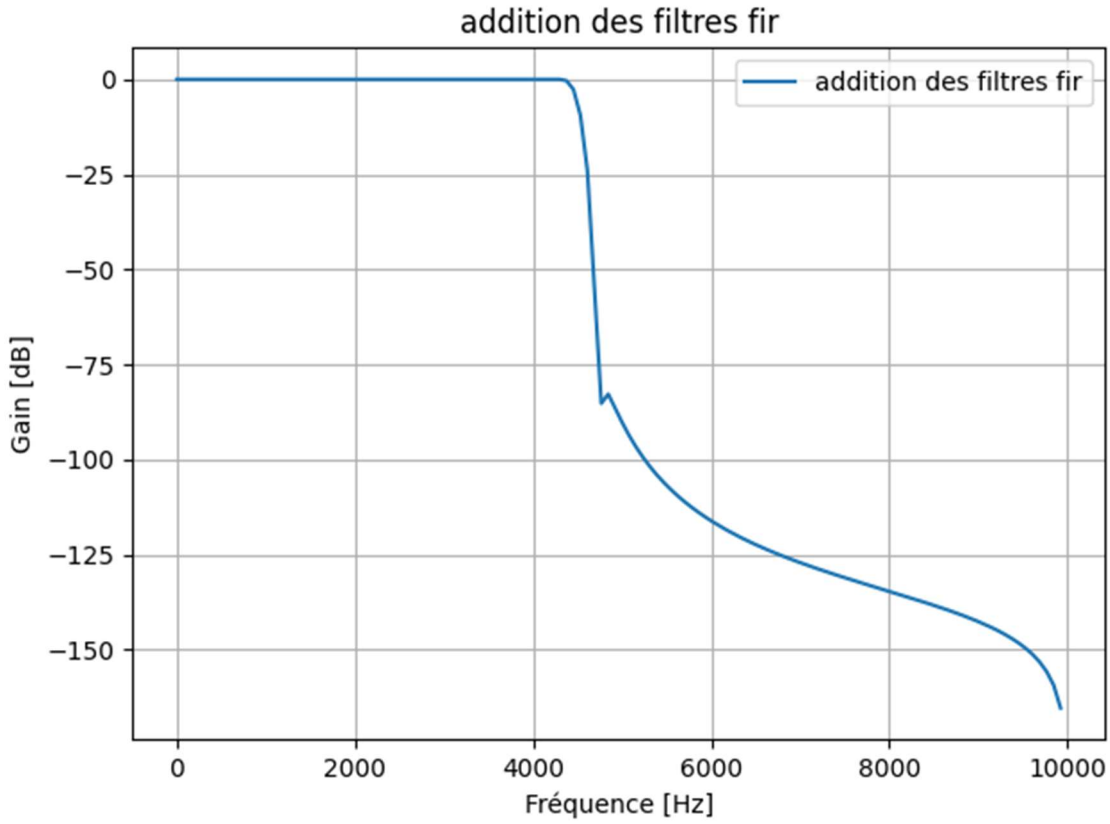


Figure 27: addition des filtres