

# Identification et Filtrage (IDFIL)

## Travaux Pratiques n° 1

---

HOUSSEYNE NADOUR

Master ARIA, ASI

Sous direction de l'enseignant  
SAÏD MOUSSAOUI

---

31 décembre 2017

### Résumé

*L'objectif de ce TP est de mettre en pratique, les différents techniques de filtrage, et comparer entre eux, les filtres à synthétiser sont des filtre Analogiques, filtres RII, et des filtres RIF. Ce travail est composé de deux parties, dans chaque partie on analyse ces différent types de filtres.*

### TABLE DES MATIÈRES

#### I. LE SIGNAL À FILTRER :

Les filtres à synthétiser seront appliqués sur un signal de durée de 8s, composé successivement de 8 signaux sinusoïdales de fréquences différentes et de même durée (1s). La fréquence d'échantillonnage a été choisi de telle façon de respecter le théorème de Schannone qui consiste à choisir la fréquence d'échantillonnage deux fois plus grande que la plus grande fréquence que le signal comporte pour extraire toute les informations utiles dans ce signal, dans notre cas la plus grande fréquence est : 523 Hz, tant que la fréquence d'échantillonnage  $f_e = 8192$ , d'où le théorème de Shcannone est respecté.

#### II. FILTRAGE PAS-BAS ANALOGIQUE

##### i. L'ordre de chaque filtre

Le script du Matlab pour calculer l'ordre de chaque filtre :

**Listing 1** – Le script du Matlab pour calculer l'ordre de chaque filtre

```
1 %% L'ordre de chaque filtre:
2 %% Butterworth:
3 [nc_b, wnc_b]= buttord(2*pi*fp,2*pi*fa,
4   Ap,Aa,'s'); % en analogique
5 %% Chabyshev de type 1:
6 [nc_c1, wnc_c1]= cheblord(2*pi*fp,2*pi*
7   fa,Ap,Aa,'s'); % en analogique
8 %% Chabyshev de type 2:
9 [nc_c2, wnc_c2]= cheb2ord(2*pi*fp,2*pi*
10  fa,Ap,Aa,'s'); % en analogique
11 %% Chabyshev de ellipse:
12 [nc_el, wnc_el]= ellipord(2*pi*fp,2*pi*
13  fa,Ap,Aa,'s'); % en analogique
```

Type du filtre	L'ordre du filtre
Butterworth	20
Chabyshev de type 1	8
Chabyshev de type 2	8
Cauer	5

Remarque :

Il est clair que la méthode de Butterworth a le plus grand ordre, par contre le filtre de Cauer a le plus petit ordre, et les deux types de Chabyshev ont le même ordre entre les filtres de Butterworth et Cauer. On peut dire alors que le filtre de Cauer est le moins coûteux parmi les autres filtres analogiques ce qui signifie que ce filtre est simple à synthétiser, et n'exige pas beaucoup de calcul et beaucoup de temps pour filtrer le signal.

Le script suivant a pour objectif de calculer les coefficients de dénominateur et numérateur de chaque filtre :

**Listing 2** – Le script du Matlab pour calculer l'ordre de chaque filtre

```

1  %% B) Les coefficients de chaque filtre
2  :
3  % Butterworth:
4  [numc_b, denc_b] = butter(nc_b, wnc_b, '
5      low', 's'); % en analogique
6  % Chabyshev de type 1:
7  [numc_c1, denc_c1] = cheby1(nc_c1, Ap,
8      wnc_c1, 'low', 's'); % en analogique
9  % Chabyshev de type 2:
10 [numc_c2, denc_c2] = cheby2(nc_c2, Aa,
11     wnc_c2, 'low', 's'); % en analogique
12 % Chabyshev de ellipse:
13 [numc_el, denc_el] = ellip(nc_el, Ap, Aa,
14     wnc_el, 'low', 's'); % en analogique
15 H_ellip = tf(numc_el, denc_el);
    
```

Par exemple le filtre Cauer est de la forme suivante :

$$H_{\text{ellip}} = \frac{94.79s^4 + 6.36e-12s^3 + 1.978e09s^2 - 3.073e-06s + 9.42e15}{s^5 + 1319s^4 + 8.828e06s^3 + 8.501e09s^2 + 1.823e13s + 9.42e15}$$

la fonction de transfert de ce filtre est plus compacte que celles des autres filtres.

## ii. les réponses fréquentielles en amplitude des quatre filtres

le script suivant permet d'obtenir les valeurs numériques de chaque fonction de transfert (chaque filtre) pour chaque valeur de fréquence  $f$  :

**Listing 3** – Le script du Matlab pour calculer la réponse de chaque filtre

```

1  %% La réponse fréquentielle de chaque
2  filtre analogique:
3  f = linspace(fmin, fmax, 1000);
4  Hc_b = freqs(numc_b, denc_b, 2*pi*f);
5  Hc_c1 = freqs(numc_c1, denc_c1, 2*pi*f);
6  Hc_c2 = freqs(numc_c2, denc_c2, 2*pi*f);
7  Hc_el = freqs(numc_el, denc_el, 2*pi*f);
    
```

On peut utiliser la fonction *margin* ou la fonction *bode* pour tracer les réponses en amplitude en phase :

**Listing 4** – réponse fréquentielle du filtre de Cauer

```

1  H_ellip = tf(numc_el, denc_el);
2  margin(H_ellip);
    
```

Mais avec ce script, les réponses sont dessinées en fonction de la pulsation au lieu de la fréquence ( $\omega = 2\pi.f$ ). pour obtenir les réponses en fréquence on utilise le script suivant :

**Listing 5** – Le script pour tracer les réponses fréquentielles

```

1  graph = plot(f, 20*log10(abs(Hc_b)), f,
2      20*log10(abs(Hc_c1)), f, 20*log10(
3      abs(Hc_c2)), f, 20*log10(abs(Hc_el)))
4      ;
5  legend(graph, 'Butterworth', 'Chabyshev
6      de type 1', 'Chabyshev de type 2', '
7      Cauer')
    
```

Les réponses sont illustrées sur la figure suivante :

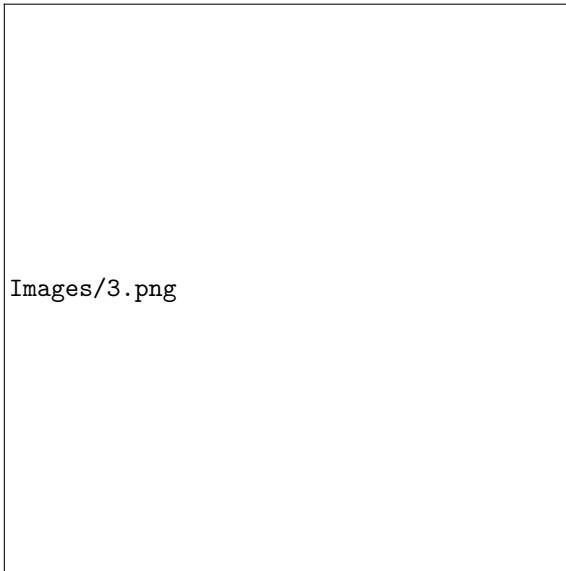


FIGURE 1 – la réponse fréquentielle de chaque filtre

#### Remarque :

Le graphe montre des caractéristiques commune et autre différentes entre les quatre filtres synthétisés :

- Tout les filtre synthétisé rejettent tous les signaux de fréquence supérieur à la fréquence de coupure désirée  $f_c = 420$
- La fréquence de coupure réelle n'est pas confondue à celle désirée (l'idéelle est 420 Hz), en effet les courbes montrent que la fréquence de coupure réelle est 360 Hz, dans notre ça ne pose pas le problem de rejeter des signaux utiles parce que le signal á filtrer comporte des fréquences espacées entre elles.
- La caractéristique en amplitude du filtre Butterworth est très plate dans la bande passante, et lisse dans bande coupée, et ça revient á l'ordre élevé du filtre. Par contre les autre filtres présentent de oscillations dans la bande passante á cause de l'ordre réduit de chaque filtre.
- Les deux filtres Chebychev type 1 et Cauer sont plus proches du filtre idéal que les filtres Butterworth et Chebychev de type 2, parce qu'ils ont une coupure extrêmement raide.

### iii. Filtrage du signal audio

Listing 6 – Le script du Matlab pour filtrer la gamme, et écouter les gammes filtrées

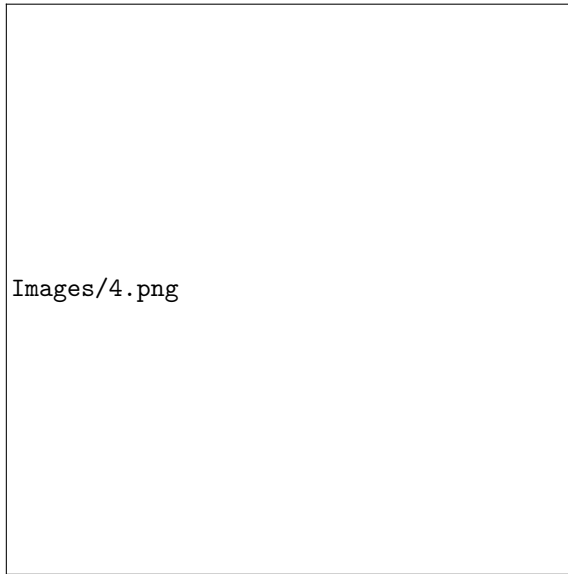
```
1 %% Filter and listen:
2 Filtred_Signal_ButterWorth = lsim(tf(
    numc_b,denc_b),sig,t);
3 soundsc(Filtred_Signal_ButterWorth, fe)
   ;
4 Filtred_Signal_Cheb1 = lsim(tf(numc_c1,
    denc_c1),sig,t);
5 soundsc(Filtred_Signal_Cheb1, fe);
6 Filtred_Signal_Cheb2 = lsim(tf(numc_c2,
    denc_c2),sig,t);
7 soundsc(Filtred_Signal_Cheb2, fe);
8 Filtred_Signal_Cauer = lsim(tf(numc_el,
    denc_el),sig,t);
9 soundsc(Filtred_Signal_Cauer, fe);
```

Les trois dernier notes ont été partialement rejetées, en plus la note *Sol* (qui est considérés comme une information utile dans la bande passante) a été influencée par le filtre, malgré qu'on ne peut jamais synthétiser un filtre idéal, mais pour notre signale ou les fréquence des notes sont espacées, on peut modifier les paramètres du filtre(la bande passantes et les amplitudes) pour rejeter parfaitement les trois dernières note sans influencer aucune note utile.

En faite, les paramètres suivants satisfont l'objective désiré :

$f_c = 425 \text{ Hz}$  au lieu  $f_c = 420 \text{ Hz}$   
 $\Delta f = 60 \text{ Hz}$  au lieu  $\Delta f = 100 \text{ Hz}$   
 $A_a = 50$  au lieu  $A_a = 40$

Les nouveaux réponses fréquentielles :



**FIGURE 2** – la réponse fréquentielle de chaque filtre après l'amélioration

Tous les filtres rejettent d'une manière presque parfaite les trois dernières notes et sans affecter la note *Sol*.

Mais cet avantage est payé par une augmentation dans les ordres des filtres :

Type du filtre	L'ordre du filtre
Butterworth	41
Chabyshev de type 1	12
Chabyshev de type 2	12
Cauer	6

Remarque :

La méthode de Cauer montre une caractéristique intéressante, son ordre n'a augmenté que de 20%, tant que celui de Butterworth a augmenté de plus que 100%, et l'ordre de Chabyshev de type 1 et 2 ont une augmentation de 50%.

Cet résultat montre que le filtre de Cauer est le meilleur choix.

#### iv. Le graphe de l'allure de la gamme filtrée dans le domaine temporel et fréquentiel

premièrement, on trace l'allure du signal filtré en utilisant les paramètres donnés dans la question. puis on trace l'allure en utilisant les paramètres proposés.

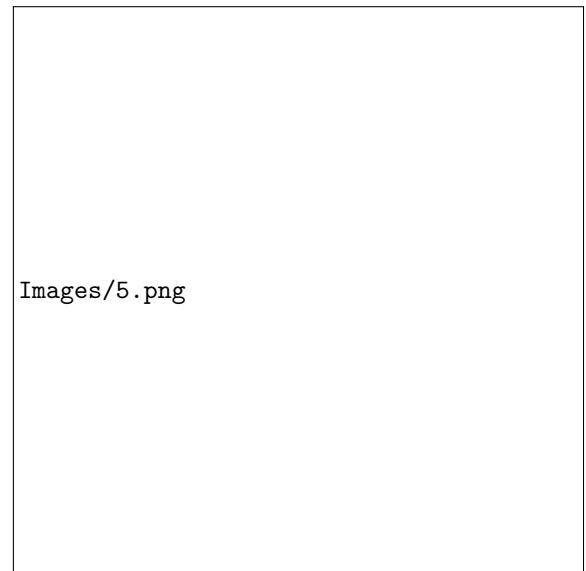
**Listing 7** – Le script du Matlab pour la gamme filtrée dans le domaine temporel

```

1 figure(2);
2 subplot(2,2,1); plot(t,
    Filtered_Signal_ButterWorth);title(
    'ButterWorth');
3 subplot(2,2,2); plot(t,
    Filtered_Signal_Cheb1);title('Cheb1
    ');
4 subplot(2,2,3); plot(t,
    Filtered_Signal_Cheb2);title('Cheb2
    ');
5 subplot(2,2,4); plot(t,
    Filtered_Signal_Cauer);title('Cauer
    ');
6
7 legend(graph,'Butterworth','Chabyshev
    de type 1','Chabyshev de type 2','
    Cauer')

```

#### iv.1 L'allure dans le domaine temporel



**FIGURE 3** – L'allure dans le domaine temporel de la gamme filtrée

Remarque : Ce graphe confirme les remarques qu'on a constaté précédemment :

- Les notes à filtrer (*La Si Do*) ont été rejetées mieux par le filtre Cauer et Chebychev 1 que Butterworth et Chebychev 2.

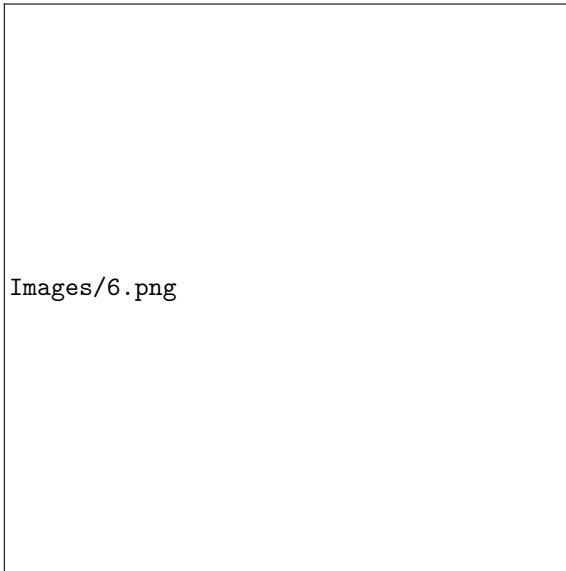
- La note *Sol* a été influencée par le filtrage, On ne va plus remarquer cette inconvénient après l'amélioration des paramètres du filtre (Voire le graphe suivant).
- On remarque des ondulation dans la bande passant pour le filtre de Chebychev type 1 et le filtre de Cauer, c'est le coût d'avoir une coupure raide (bande transitoire très courte), cette remarque est confirmée par la réponse fréquentielle qui montre des ondulation dans la bande passante. Par contre les deux filtres de Butterworth et Chebychev type 2, ne présentent pas cet inconvénient, surtout le filtre de Butterworth, c'est expliqué par la réponse fréquentielle qui montre une caractéristique en amplitude très plate dans la bande passante.
- Les altérations du signal filtré dans la bande passante (les ondulations) signifie que le volume du son change dans la bande passante.

Si on utiliser les paramètres suivant :

$$f_c = 425 \text{ Hz} \quad \text{au lieu} \quad f_c = 420 \text{ Hz}$$

$$\Delta f = 60 \text{ Hz} \quad \text{au lieu} \quad \Delta f = 100 \text{ Hz}$$

$$A_a = 50 \quad \text{au lieu} \quad A_a = 40 \text{ On trouve les résultats désirés.}$$

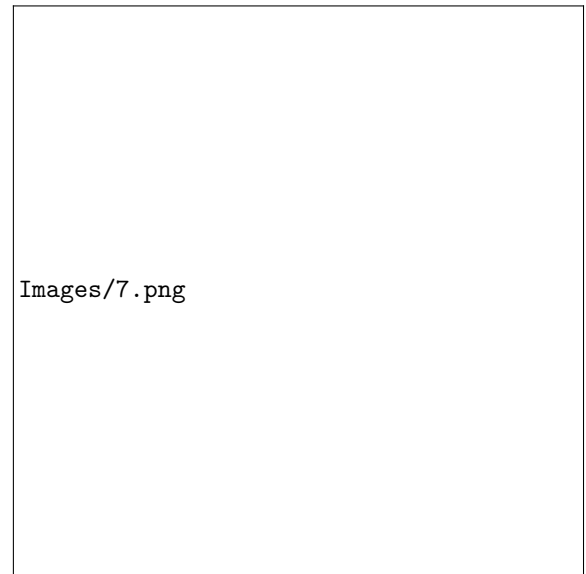


**FIGURE 4** – *L'allure dans le domaine temporel de la gamme filtrée après l'amélioration des paramètres*

Remarque : Le filtre de Butterworth montre des

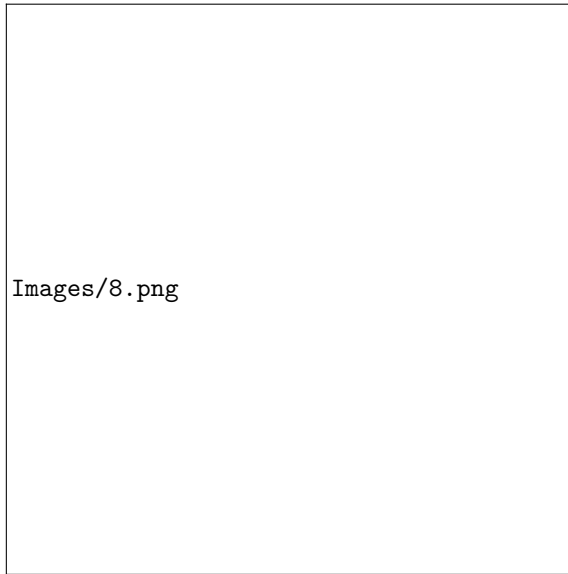
bonnes caractéristiques dans la bande passante et dans la bande coupée par rapport à les autres filtres, mais c'est payé par le coût de l'ordre de ce filtre ( $n = 40$ ).

#### iv.2 Le spectre d'amplitude de la gamme filtrée



**FIGURE 5** – *Le spectre de la gamme filtrée*

Ce graphe confirme les points remarqués sur l'allure de la gamme filtrée dans le domaine temporel. En utilisant les paramètres proposés, on trouve des bons résultat dans les deux bandes, passantes et coupée.



**FIGURE 6** – *Le spectre de la gamme filtrée après l'amélioration des paramètres*

## v. Modification de la bande de transition

le paramètre qui va se changer c'est :

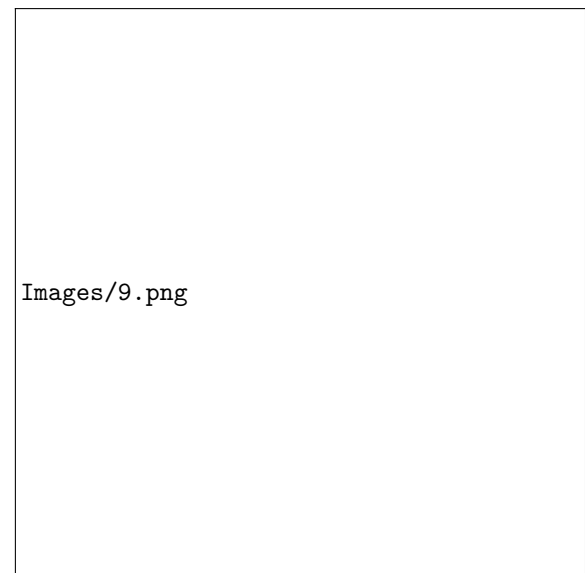
$$\Delta f = 20 \text{ Hz}$$

On résume les résultats trouvés pour chaque filtre dans les graphes suivants :

### v.1 L'ordre de chaque filtre

Type du filtre	L'ordre du filtre
Butterworth	111
Chabyshev de type 1	20
Chabyshev de type 2	20
Cauer	7

### v.2 les réponses fréquentielles en amplitude des quatre filtres



**FIGURE 7** – *les réponses fréquentielles en amplitude des quatre filtres*

### v.3 L'allure de la gamme filtrée dans le domaine temporel

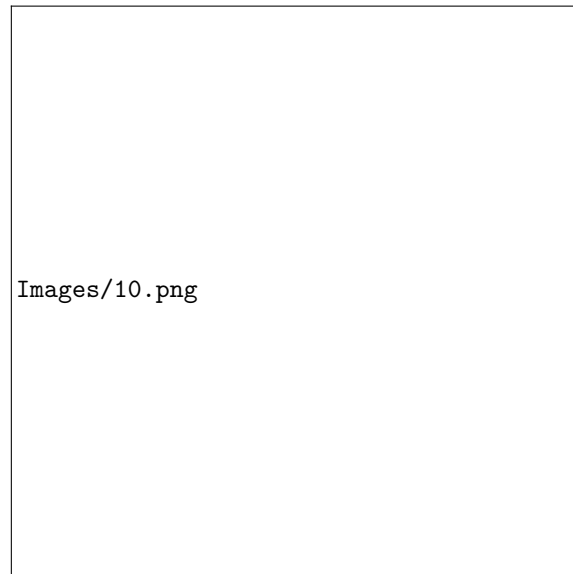


FIGURE 8 – L'allure de la gamme filtrée dans le domaine temporel

Remarque : On remarque des avantages et des inconvénients en modifiant la bande de transitoire :

- L'ordre de chaque filtre a augmenté, surtout le filtre de Butterworth, qui a exigé un ordre de 111, et ça rend le filtre non applicable, et on peut voir que le matlab n'a pas pu dessiner les allures pour le filtre de Butterworth due au chut de calcul. par contre le filtre de Cauer reste toujours dans le cadre pratique due au son ordre qui est 7.
- Un bon filtrage de notes à rejeter dans la bande à coupée.
- Les ondulations dans la bandes passantes ont disparu, que signifie que le volume du son ne change pas dans la bande passante.
- Les filtres ont des coupure extrêmement raides (bande transitoire très courte), et d'où l'avantage suivant :
- Les signales utiles (comme la note Sol) ne ont pas été influencée par le filtrage.

### v.4 L'allure de la gamme filtrée dans le domaine fréquentiel



FIGURE 9 – L'allure de la gamme filtrée dans le domaine fréquentiel

### III. ELIMINATION D'UNE NOTE PAR FILTRAGE COUPE-BANDE

#### i. Filtre numérique coupe-bande de type Chebyshev 2

Le script pour calculer le filtre :

**Listing 8** – Le script pur calculer le Filtre numérique coupe-bande de type Chebyshev 2, et pour tracer la réponse fréquentielle

```

1  %% Chabyshev de type 2:
2  Df = 10;
3  fc_b = 340;
4  fc_h=360;
5  % Df = 5;
6  % fc_b = 345;
7  % fc_h=355;
8
9  fp_b = fc_b - Df/2;
10 fp_h = fc_h - Df/2;
11 fa_b = fc_b + Df/2;
12 fa_h = fc_h + Df/2;
13 Wp = 2*[fa_b fp_h]/fe; % IN
14 Ws = 2*[fp_b fa_h]/fe; % EX
15 Ap=1; Aa=40;
16 %% A) Filtrage Coup-Bande analogique
17 %% L'ordre de chaque filtre:
18 [nd_c2, wnd_c2]= cheb2ord(Wp,Ws,Ap,Aa)
19      ;
20 %% B) Les coefficient de chaque filtre:
21 [numd_c2, dend_c2] = cheby2(nd_c2,Aa,
22      wnd_c2,'stop');
23 %
24 %% C) Les reponses frequentielles;
25 figure(2)
26 f = linspace(fmin,fmax,1000);
27 Hd_c2 = freqz(numd_c2,dend_c2,f,fe);
28 plot(f,20*log10(abs(Hd_c2)));
    
```

Images/12.png

**FIGURE 10** – La réponse fréquentielle du filtre RII de type Chebychev 2

- La graphe est très plat dans les bandes passantes (il n'y a pas des ondulations dans cette bande), que signifie que le volume du son ne est pas altéré.
- La note  $Fa$  qui correspond à la fréquence  $349Hz$  est rejetée.
- Malgré que la coupure est raide (bande transitoire très courte), le filtre touche la bande passant (la bande coupée réelle est plus large que celle désirée), que signifie que la note la plus proche  $Mi$  est altéré par ce filtre. on peut régler ce problem en prenant les paramètres suivants :

$$\Delta f = 5$$

$$f_c^b = 345$$

$$f_c^h = 345$$

Et on trouve le résultat désiré :



Images/14.png

**FIGURE 11** – La réponse fréquentielle du filtre RII de type Chebychev 2 après l'amélioration des paramètres

Images/15.png

**FIGURE 12** – L'allure temporelle de la gamme filtrée

## ii. L'allure temporelle et le spectre d'amplitude de la gamme filtrée

**Listing 9** – Le script pur Tracer l'allure temporelle et le spectre d'amplitude de la gamme filtrée

```

1  %% D1) Tracer sur la meme figure 1
   allure temporelle et le spectre d
   amplitude de lagamme filtree
2  figure(3)
3  Cheb2 = filter(numd_c2,dend_c2,sig);
4  plot(t,Cheb2);title('Cheb2');
5  %% D2) Le spectre d'amplitude de la
   gamme filtree
6  figure(4)
7  ffc = [0:length(Cheb2)-1]*(fe/length(
   Cheb2));
8  fmin = 100; fmax = 600; ymax = 0.6;
9  Sfc = fft(Cheb2)/fe; plot(ffc, abs(Sfc
   )); axis([fmin fmax 0 ymax]); hold
   on;
10 soundsc(Cheb2, fe)

```

Images/13.png

**FIGURE 13** – Le spectre d'amplitude la gamme filtrée

Cet résultat confirme les points remarqués sur le graphe de la réponse fréquentielle du filtre :

- Le volume du son n'est pas altéré (sauf la note *Mi*), parce que le filtre ne présente pas des ondulations dans cette bande
- La note *Fa* qui correspond à la fréquence 349 est rejetée.
- La note la plus proche *Mi* est altéré par ce filtre. on peut régler ce problem en prenant

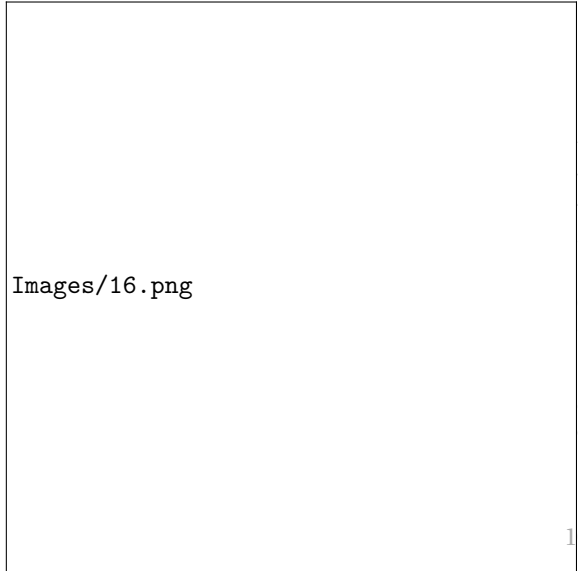
les paramètres suivants :

$$\Delta f = 5$$

$$f_c^b = 345$$

$$f_c^h = 345$$

On trouve le résultat désiré :



**FIGURE 14** – L'allure temporelle de la gamme filtrée après l'amélioration des paramètres



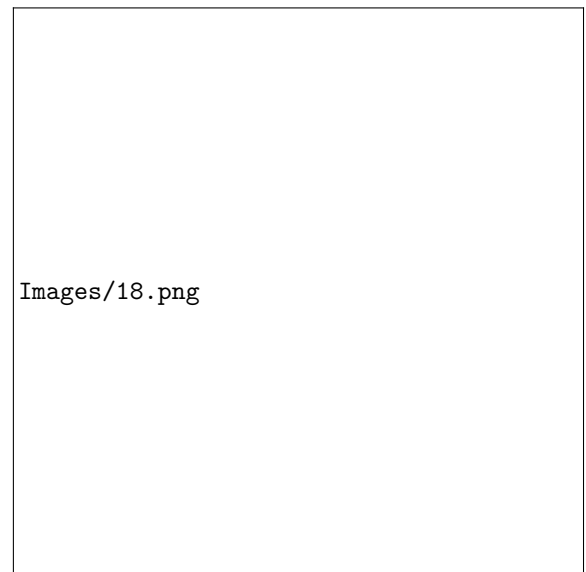
**FIGURE 15** – Le spectre d'amplitude de la gamme filtrée après l'amélioration des paramètres

iii. Le filtre RIF de longueur minimal (fenêtre de Kaiser)

**Listing 10** – Le script pur calculer Le filtre RIF de longueur minimal (fenêtre de Kaiser) et pour tracer la réponse fréquentielle et tracer la gamme filtrée

```
1 %% D1) Tracer sur la meme figure l
2 allure temporelle et le spectre d
3 amplitude de lagamme filtree
4 figure(3)
5 Cheb2 = filter(numd_c2,dend_c2,sig);
6 plot(t,Cheb2);title('Cheb2');
7 %% D2) Le spectre d'amplitude de la
8 gamme filtree
9 figure(4)
10 ffc = [0:length(Cheb2)-1]*(fe/length(
    Cheb2));
    fmin = 100; fmax = 600; ymax = 0.6;
    Sfc = fft(Cheb2)/fe; plot(ffc, abs(Sfc
    )); axis([fmin fmax 0 ymax]); hold
    on;
    soundsc(Cheb2, fe)
```

Les résultats de la simulation sont obtenus en utilisant les paramètres suivants :  $\Delta f = 5$ ,  $f_c^b = 345$ , et  $f_c^h = 345$



**FIGURE 16** – La réponse fréquentielle du filtre RIF ( avec fenêtre de Kaiser)



**FIGURE 17** – *L'allure temporelle de la gamme filtrée*



**FIGURE 19** – *La réponse fréquentielle du filtre de Chebychev 2 et celui de la RIF (avec fenêtre de Kaiser)*



**FIGURE 18** – *Le spectre d'amplitude de la gamme filtrée*

Remarque Ce filtre a réalisé l'objectif de la manière désirée.

#### iv. Comparaison entre le filtre de Chebychev de type 2 et celui de la RIF avec la fenêtre de Kaiser

On trace les deux réponses fréquentielles dans le même graphe.

##### Remarque

Le graphe montre clairement que le filtre de RIF avec la fenêtre de Kaiser est plus performant que celui de Chebychev 2, en effet :

- la bande à couper est plus courte que celle de Chebychev 2, que nous permet de remarquer le point suivant :
- Les informations utiles (les notes à conserver dans la bande passante) sont bien conservés, sans être influencée (le volume ne change pas).
- La coupure du filtre de RIF avec la fenêtre de Kaiser est plus raide que celle de Chebychev 2 (la bande de transition est très courte).

#### iv.1 Conclusion

Le filtre de RIF avec la fenêtre de Kaiser est plus performant que celui de Chebychev 2.