Michellod Yan | Mottiez Gilles | 10 janvier 2019

Filtre numérique

Electronique

Table des matières

[Introduction 2](#_Toc535326427)

[1.Transformation s->z 2](#_Toc535326428)

[1.1 Transformation exacte aux pôles et aux zéros 2](#_Toc535326429)

[1.2 Transformation bilinéaire aux pôles et aux zéros 3](#_Toc535326430)

[1.3 Comparaison des filtres numériques et analogique 4](#_Toc535326431)

[1.4 Simulation du comportement des filtres 0](#_Toc535326432)

[2. Synthèse d’un filtre IIR 3](#_Toc535326433)

[2.1 Approximation semi-manuelle 3](#_Toc535326434)

[2.2 Approximation Matlab 0](#_Toc535326435)

[2.3 Sections du 2ème ordre 0](#_Toc535326436)

[2.4 Mesure 0](#_Toc535326437)

Table des figures

[Figure 1 : diagramme des pôles et zéros de H(z) exacte 3](#_Toc535326493)

[Figure 2: diagramme des pôles et zéros de H(z) en approximation bilinéaire 4](#_Toc535326494)

[Figure 3: diagramme de Bode des 3 filtres 0](file:///C:\Users\Yan%20Michellod\Documents\GitHub\Eln_dsp\Rapport_filtre_numérique.docx#_Toc535326495)

[Figure 4: comportement lors d'un saut unité 0](file:///C:\Users\Yan%20Michellod\Documents\GitHub\Eln_dsp\Rapport_filtre_numérique.docx#_Toc535326496)

[Figure 5: injection d'un signal sinusoïdal 100 Hz dans le filtre analogique 0](#_Toc535326497)

[Figure 6:injection d'un signal sinusoïdal 100 Hz dans le filtre numérique exact 0](#_Toc535326498)

[Figure 7:injection d'un signal sinusoïdal 100 Hz dans le filtre numérique bilinéaire 1](#_Toc535326499)

[Figure 8: injection d'un signal sinusoïdal 2kHz dans le filtre analogique 1](#_Toc535326500)

[Figure 9: injection d'un signal sinusoïdal 2kHz dans le filtre numérique exact 2](#_Toc535326501)

[Figure 10:injection d'un signal sinusoïdal 2kHz dans le filtre numérique bilinéaire 2](#_Toc535326502)

[Figure 11: gabarit filtre passe-bas 3](#_Toc535326503)

[Figure 12: diagramme de bode des filtres analogique et numériques 0](file:///C:\Users\Yan%20Michellod\Documents\GitHub\Eln_dsp\Rapport_filtre_numérique.docx#_Toc535326504)

[Figure 13: réponse en fréquence normalisée du filtre numérique 0](#_Toc535326505)

# 

# **Introduction**

Les objectifs de ce laboratoire sont la compréhension de la discrétisation d’un filtre et la prise de conscience de ces effets, la compréhension de la conception de filtres numériques IIR et l’implémentation d’un filtre numérique en virgule fixe sur DSP

# 1.Transformation s -> z

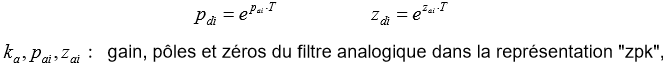
Soit la fonction de transfert d’un filtre analogique passe-bas de type Butterworth suivant :



La fréquence d’échantillonnage choisit pour la discrétisation du filtre est à 20 kHz, car en général la fréquence d’échantillonnage est 10 à 20 fois plus élevée que la fréquence de coupure du filtre.

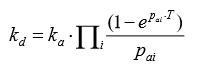
## 1.1 Transformation exacte aux pôles et aux zéros

En utilisant la fonction **tf2zpk** de matlab il est possible de déterminer sa fonction de transfert en z. Mais les coefficients z, p et k tirés de cette fonction viennent du filtre analogique. Pour les transformer en coefficients de fonction de transfert numérique, il faut utiliser les formules suivantes :





De plus, pour un filtre passe-bas, le gain vaut :



Matlab a fourni des résultats que nous avons affiché sur un graphe permettant de visualiser si les résultats sont cohérents avec la théorie.

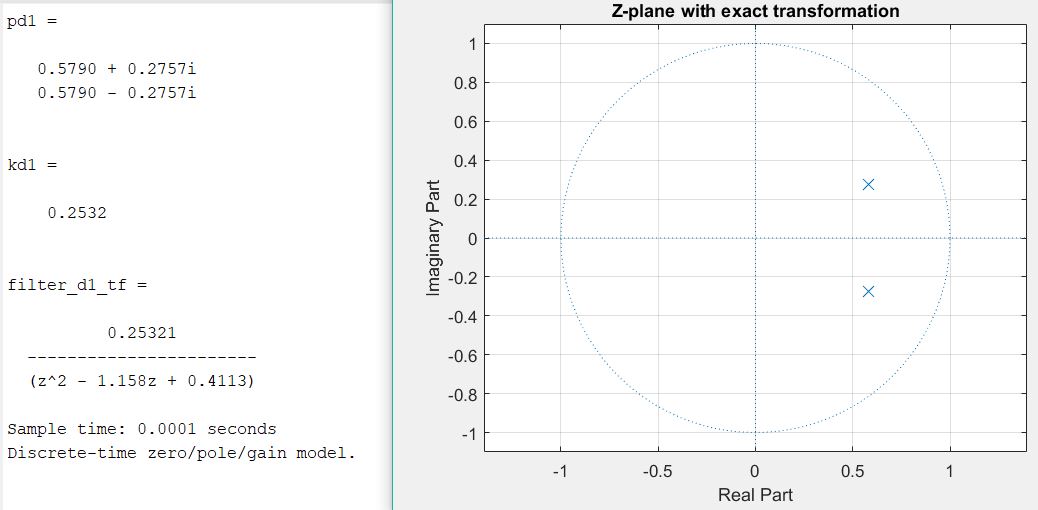


Figure 1 : diagramme des pôles et zéros de H(z) exacte

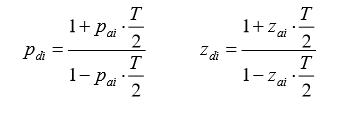
La transformation en z ne crée pas de zéro s’il n’y en a pas dans la fonction en s.

On peut affirmer que les résultats sont cohérents avec la théorie.

## 1.2 Transformation bilinéaire aux pôles et aux zéros

La transformation bilinéaire est une approximation de la transformation exacte z = esT.

Cette approximation se définit de cette manière :



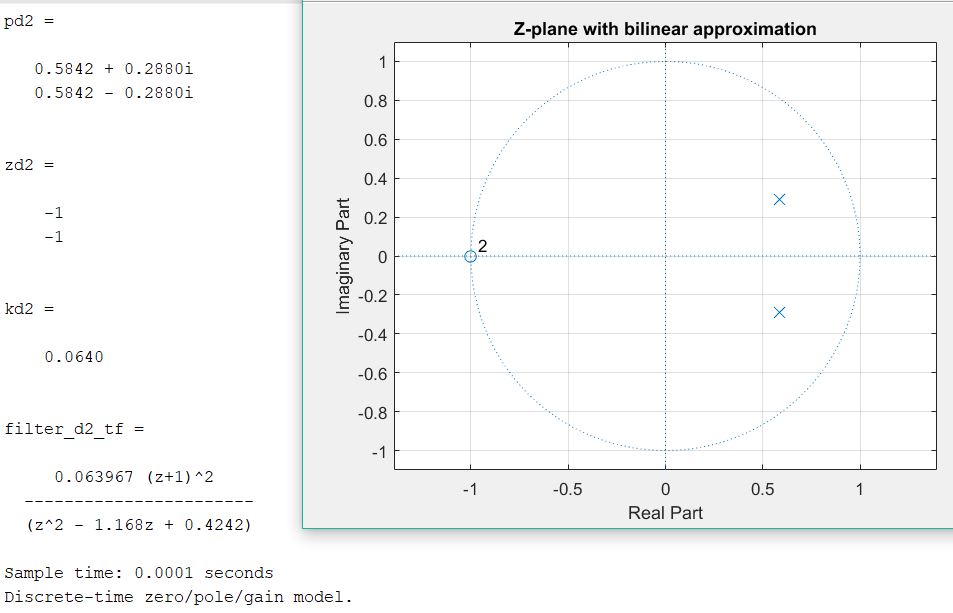


Figure 2: diagramme des pôles et zéros de H(z) en approximation bilinéaire

Les zéros sont apparus à -1 car il n’y a pas de zéro dans la fonction de transfert en s et les pôles sont à l’intérieur du cercle unité dans la partie positive des nombres réels.

## 1.3 Comparaison des filtres numériques et analogique

Les pôles des deux filtres numériques sont proches malgré l’approximation. Les filtres numériques ont un comportement identique que le filtre analogique à basse fréquence. Mais lorsque la fréquence d’entrée du signal dépasse la fréquence de coupure du filtre, le comportement des filtres numériques s’éloigne de celui du filtre analogique.

Le comportement du filtre digital exact se rapproche de celui analogique, mais il est moins efficace pour les hautes fréquences. Alors que le filtre approximé a un comportement asymptotique, et de ce fait filtre très bien les hautes fréquences.

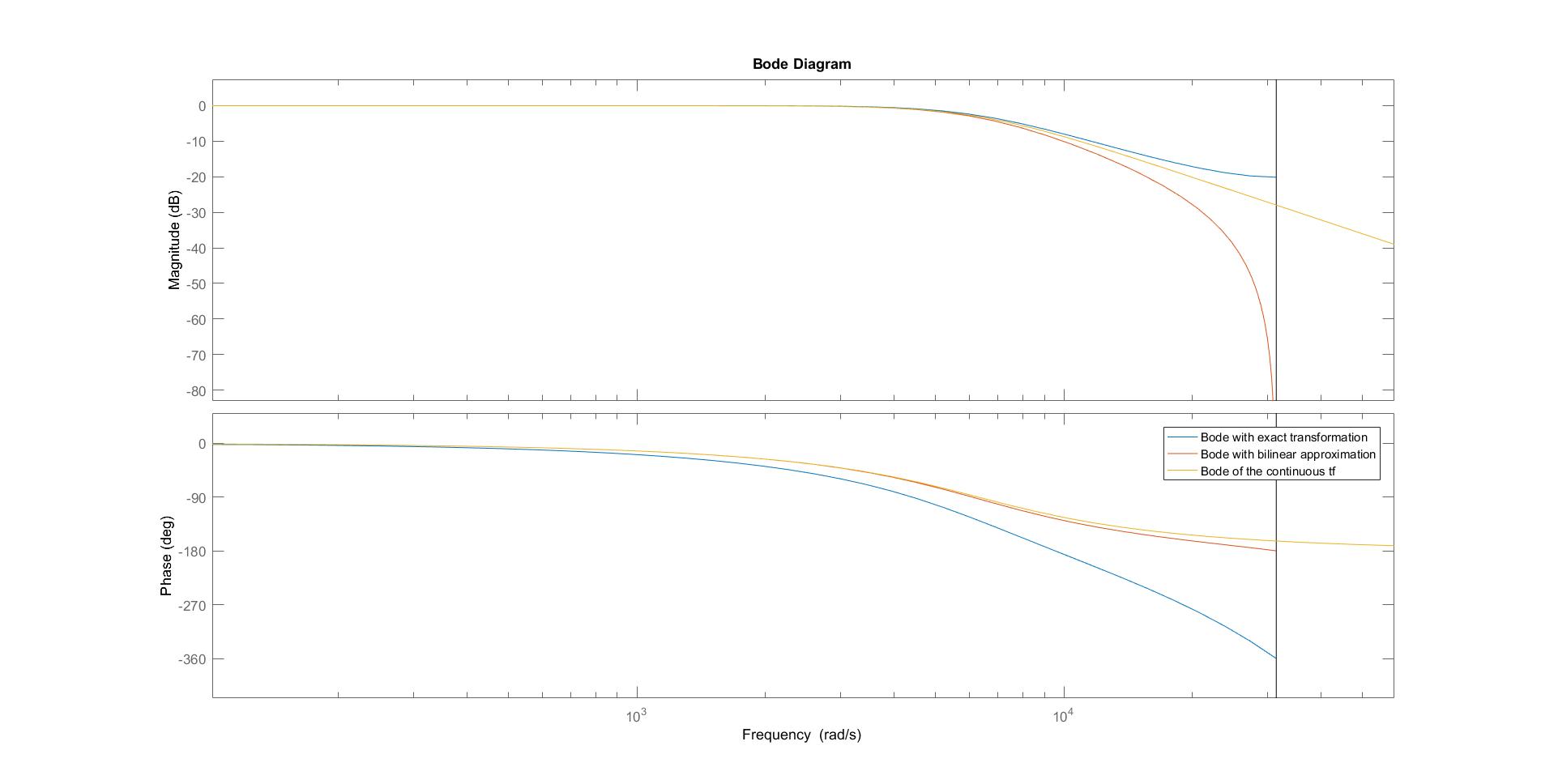
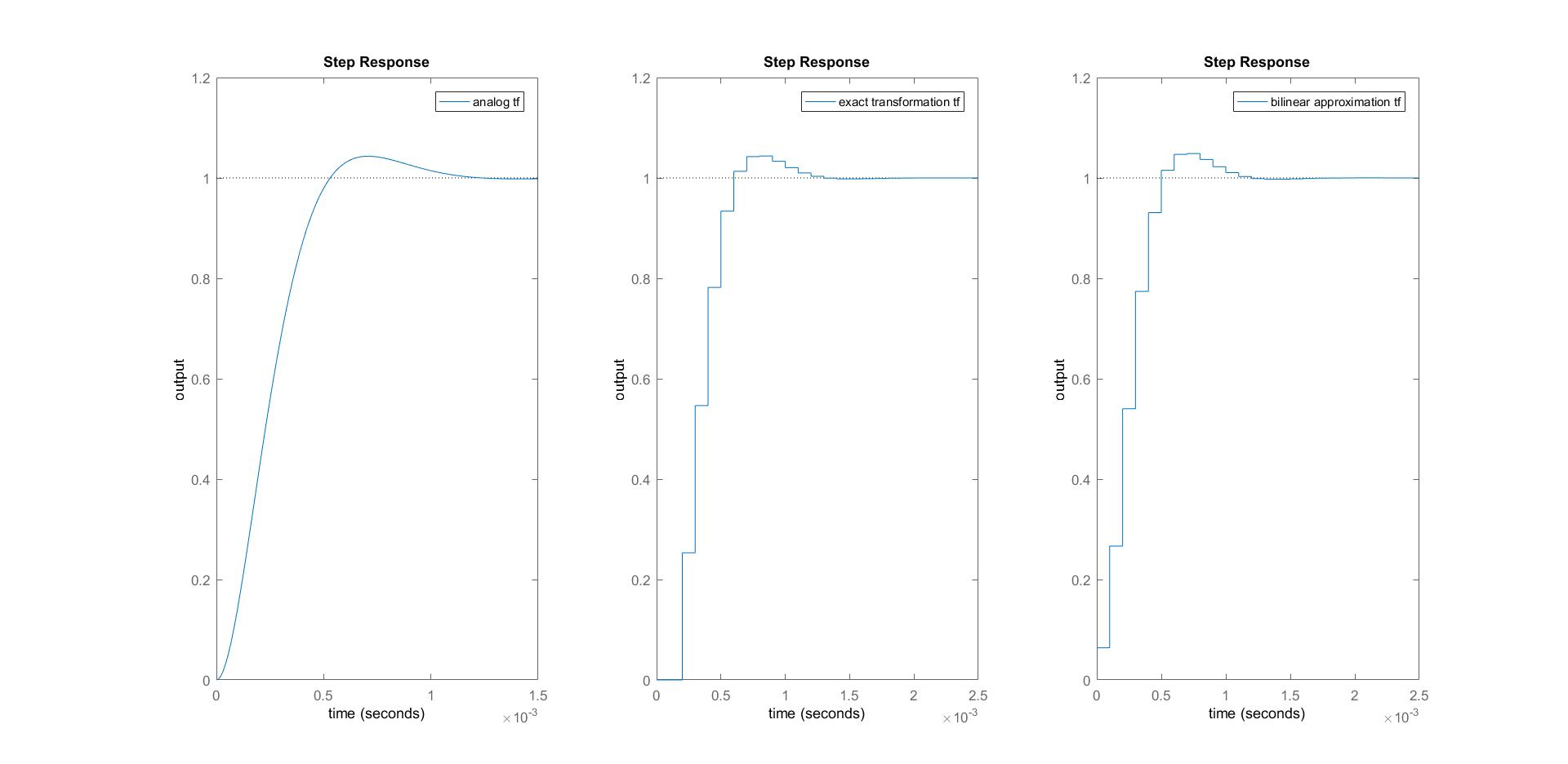


Figure 3: diagramme de Bode des 3 filtres

## 1.4 Simulation du comportement des filtres

Pour tester les filtres dimensionnés précédemment et le filtre analogique, on leurs injecte un saut indiciel, un sinus à 100Hz et un sinus à 2 kHz

Le comportement des filtres numériques est quasiment équivalent et très proche de celui analogique. Dans tous les cas le gain DC vaut 1, et la stabilisation a lieu après environ 1.3 seconde.

Figure 4: comportement lors d'un saut unité

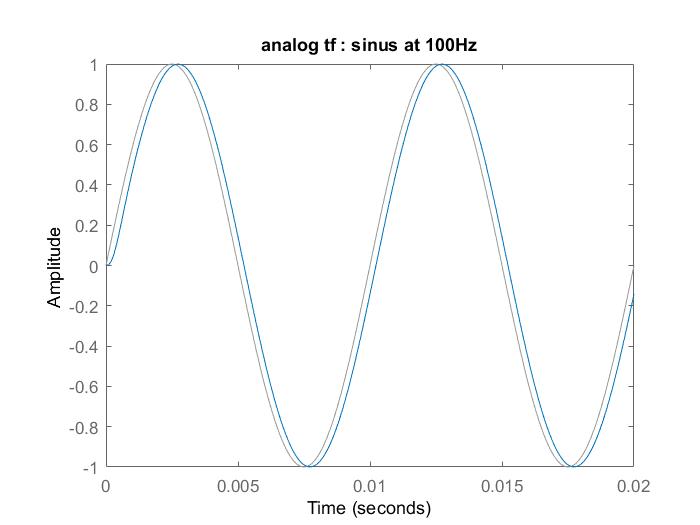


Figure 5: injection d'un signal sinusoïdal 100 Hz dans le filtre analogique

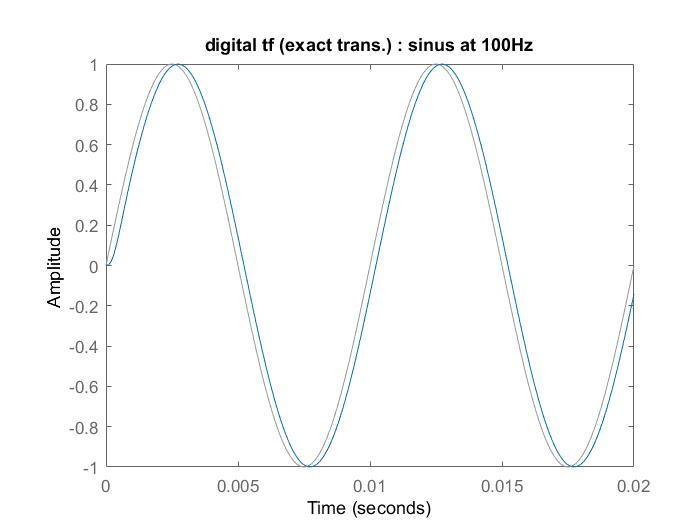


Figure 6:injection d'un signal sinusoïdal 100 Hz dans le filtre numérique exact

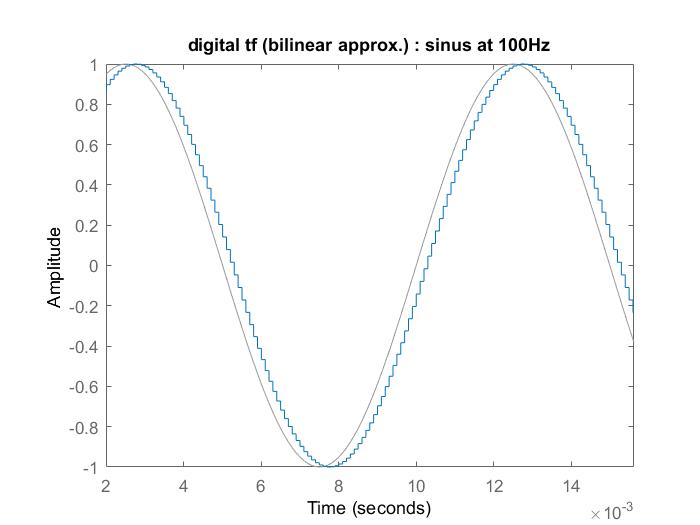


Figure 7:injection d'un signal sinusoïdal 100 Hz dans le filtre numérique bilinéaire

Il n’est pas possible de différencier les 3 filtres lors de ce test. Excepté que les filtres numériques ont des steps dus à la période d’échantillonnage.

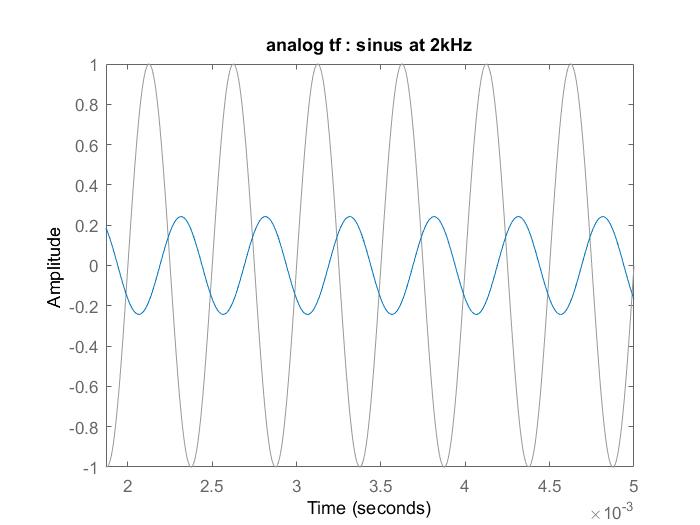


Figure 8: injection d'un signal sinusoïdal 2kHz dans le filtre analogique

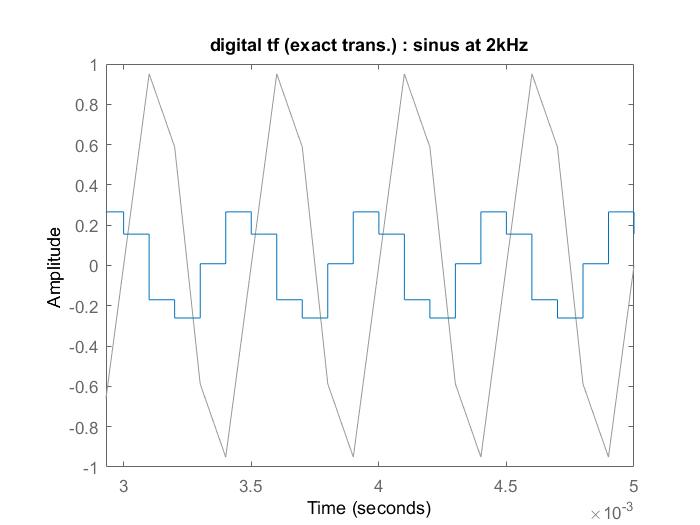


Figure 9: injection d'un signal sinusoïdal 2kHz dans le filtre numérique exact

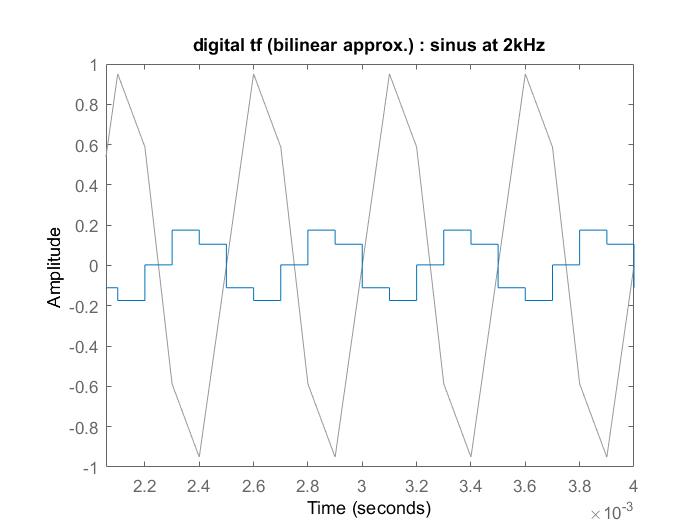


Figure 10:injection d'un signal sinusoïdal 2kHz dans le filtre numérique bilinéaire

A 2 kHz, les signaux numériques ne ressemblent presque plus à des sinus dû au faite qu’on se rapproche de la moitié de la fréquence d’échantillonnage qui est de 5 kHz. Mais il est quand même possible de comparer les figure 8,9 et 10 avec le diagramme de Bode (fig. 3).

On constate que le résultat du filtre numérique exacte à un plus grand déphasage que les deux autres et que son atténuation est plus faible. C’est ce que l’on retrouve avec le digramme de Bode.

Le résultat du filtre numérique bilinéaire a à peu près le même déphasage que le filtre numérique et une tension de sortie légèrement plus bas. Ces constatations correspondent au digramme de Bode.

# 2. Synthèse d’un filtre IIR

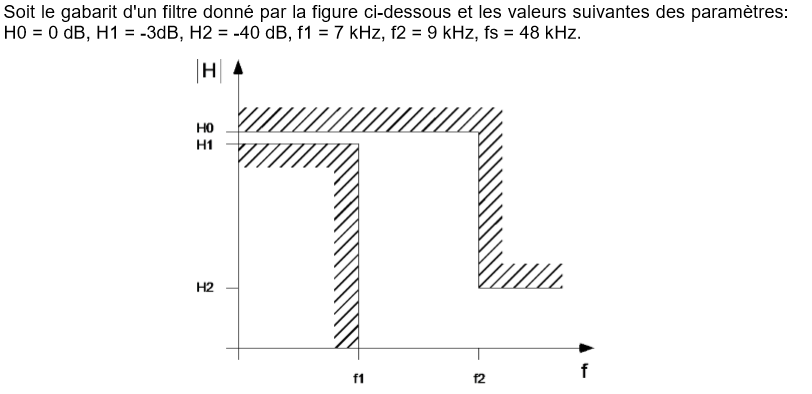


Figure 11: gabarit filtre passe-bas

## 2.1 Approximation semi-manuelle

Il faut premièrement transformer ce gabarit afin de pouvoir la méthode de développement des filtres analogiques.

Les fréquences sont légèrement modifiées et sont calculé à l’aide de la formule : f =

On obtient une nouvelle valeur de f1 à 7.534 kHz et f2 à 10.209 kHz.

Pour designer le filtre, les fonctions matlab **ellipord** et **ellip** sont utilisées afin de trouver l’ordre et la fréquence de coupure qui correspondent au gabarit. Finalement, grâce à la routine **zpk** on peut trouver la fonction de transfert analogique. On utilise ensuite la routine **bode** qui permet de visualiser la réponse en fréquence de la fonction de transfert et ainsi valider sa bonne forme.

Ensuite cette fonction analogique est transformée (à partir de ses coefficients z, p, k) en fonction de transfert numérique avec la méthode **bilinear** de matlab.

Un diagramme de Bode tracé avec les deux fonctions de transfert permet de vérifier le comportement du filtre.

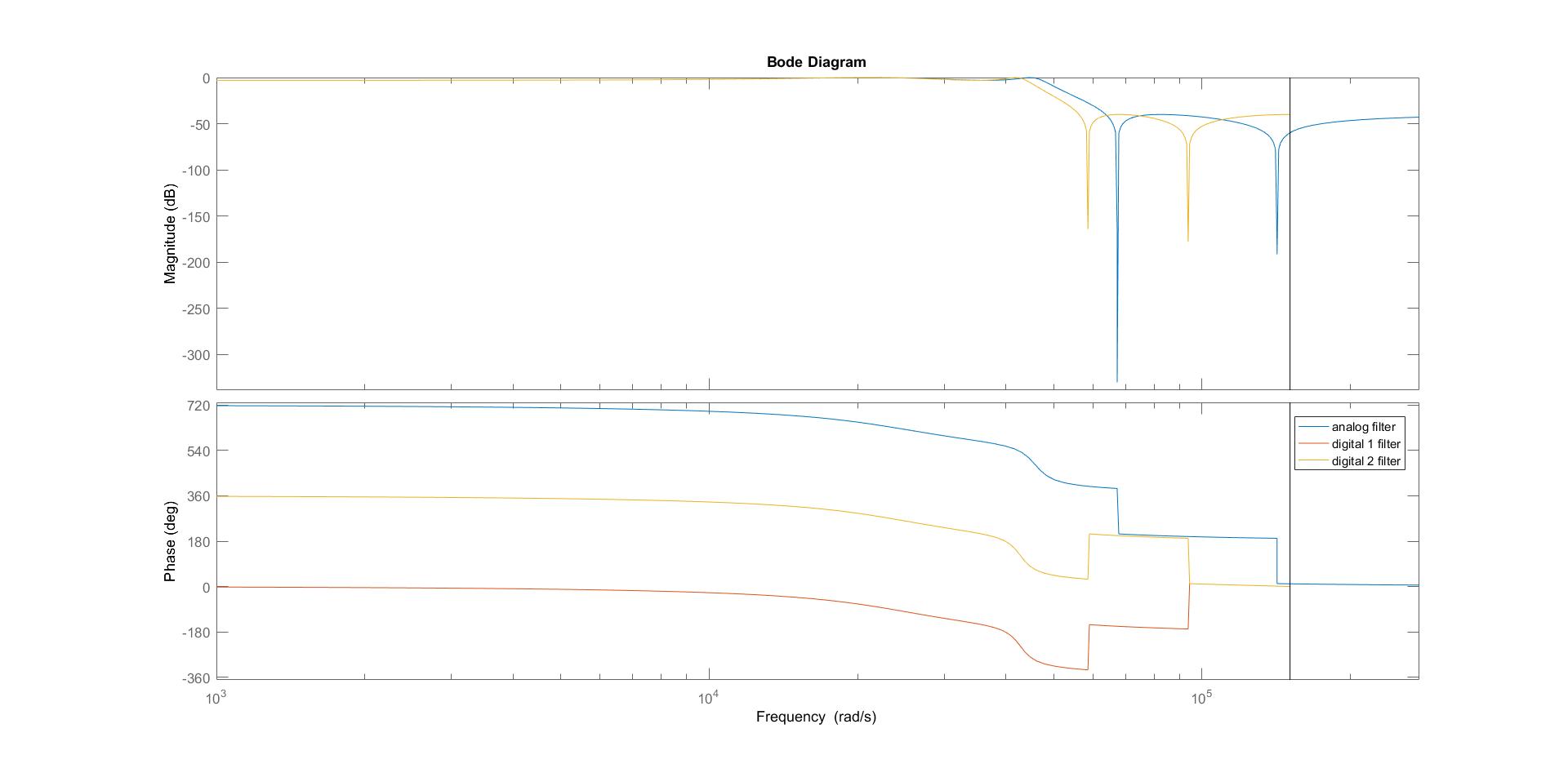
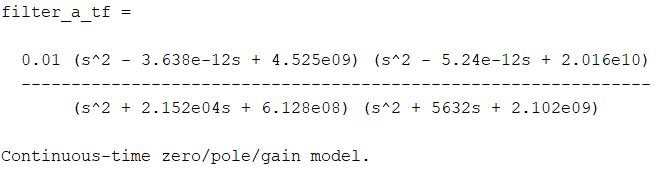


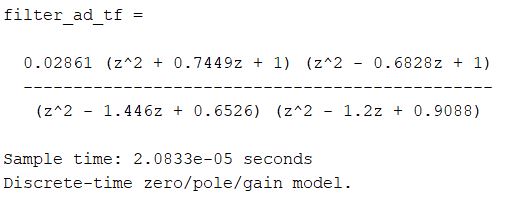
Figure 12: diagramme de bode des filtres analogiques et numériques

On peut constater que le filtre numérique se retrouve dans le gabarit de base.

La fonction de transfert analogique donnée par le script Matlab :



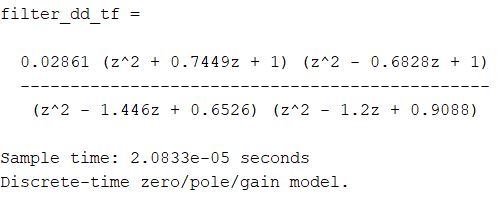
La fonction de transfert digitale obtenue par conversion de la fonction analogique :



## 2.2 Approximation Matlab

Matlab permet de d’approximer un filtre numérique directement avec les fonctions **ellipord** et **ellip,** mais il faut utiliser les fréquences normalisées par rapport à la fréquence d’échantillonnage.

Le résultat donne :



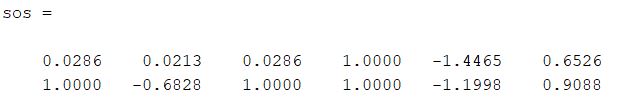
Mathématiquement, la fonction obtenue par la conversion analogique et celle calculée par le script directement en digital sont les mêmes.

Les deux courbes de Bode des filtres numérique dans la figure 12 se confondent.

## 2.3 Sections du 2ème ordre

Afin de séparer la fonction en z en multiplication de fonction du second ordre, la fonction **zp2sos** de matlab est utilisée.

Cette fonction donne les coefficients suivants :



Chaque ligne correspond à un étage du 2ème ordre :

* Les 3 premiers coefficients sont b01, b11, b21
* Les 3 derniers coefficients sont a01, a11, a21

## 2.4 Mesure

Afin de faire la mesure du filtre, la méthode **freqz** est utilisé. Le diagramme de réponse ci-dessous permet de vérifier le bon fonctionnement du filtre.

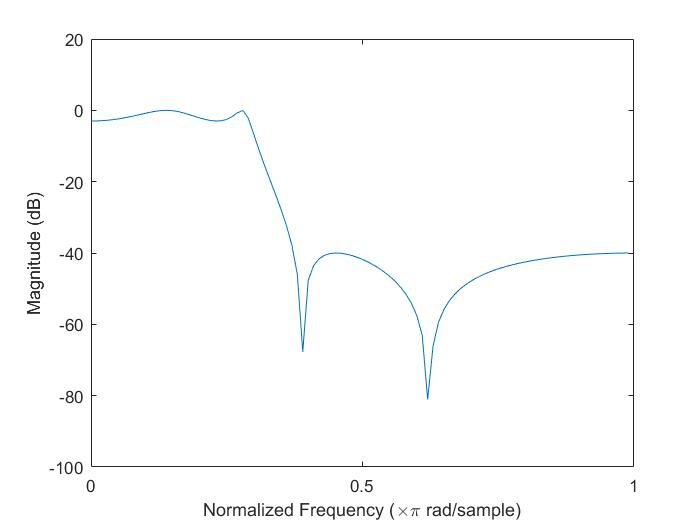


Figure 13: réponse en fréquence normalisée du filtre numérique

# 3.Réalisation DSP

Lors de cette partie, un filtre numérique sera réalisé avec un programme en C qui tournera sur un DSP en virgule fixe.

Pour la réalisation, la structure canonique en forme directe II transposée est choisie.

## 3.1 Forme directe II transposée

Cette structure permet d’utiliser deux fois moins de mémoire que la forme directe I. Elle est définie comme suit :

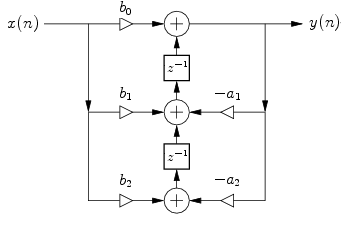


Figure 14: Forme direct II transposée

L’équation aux différences tirée du schéma ci-dessus donne :

y[n] = x[n] \* b0 + m1[n-1]

m1[n] = x[n] \* b1 – y[n] \* a1 + m2[n-1]

m2[n] = x[n] \* b2 – y[n] \*a2

en pseudo-code C cela donne :

y = x \* b0 + m1 ;

m1 = x \* b1 – y \* a1 + m2 ;

m2 = x\*b2 – y \* a2 ;

en implémentation C cela donne