

République Algérienne Démocratique et Populaire  
Ministère de l'Enseignement Supérieur et de la  
Recherche Scientifique



Université des Sciences et de la Technologie Houari  
Boumediene  
Faculté d'Informatique  
Département Sécurité Informatique

---

Rapport du Projet de Traitement du Signal  
Numérique

Membres de l'équipe :

Ghribi Ahlam  
Besmala Laidli  
Mhenni Fatma Zohra  
May  
Abdellaoui Asma

Année académique : 2024/2025

# Table des matières

<b>Introduction Générale</b>	<b>2</b>
<b>1 TP N° 2 : Génération et Affichage de Signaux</b>	<b>3</b>
<b>TP N° 2 : Génération et Affichage de Signaux</b>	<b>3</b>
1.1 Introduction . . . . .	3
1.2 Objectif . . . . .	3
1.3 Réalisation . . . . .	3
1.3.1 L'impulsion de Dirac . . . . .	3
1.3.2 L'échelon unitaire . . . . .	5
1.3.3 La fonction porte . . . . .	7
1.3.4 Le signal sinusoïdal . . . . .	8
1.3.5 Signal carré . . . . .	10
1.3.6 Signal dent de scie . . . . .	11
1.3.7 Signal triangulaire . . . . .	12
1.3.8 Signal sinus cardinal . . . . .	13
1.4 Conclusion . . . . .	14
<b>2 TP N° 3 : Analyse et Synthèse des Filtres Analogiques</b>	<b>15</b>
<b>TP N° 3 : Analyse et Synthèse des Filtres Analogiques</b>	<b>15</b>
2.1 Introduction . . . . .	15
2.2 Objectif . . . . .	15
2.3 Réalisation . . . . .	15
2.3.1 Synthèse d'un filtre passe-bas de Butterworth . . . . .	15
2.3.2 Propriétés spécifiques des filtres Butterworth . . . . .	21
2.3.3 Analyse du filtre . . . . .	22
<b>3 TP N° 4 : Transformée de Fourier Rapide Directe et Inverse</b>	<b>26</b>
<b>TP N° 4 : Transformée de Fourier Rapide Directe et Inverse</b>	<b>26</b>
3.1 Introduction . . . . .	26
3.2 Objectif . . . . .	26
3.3 Prédétermination théorique . . . . .	26
3.4 Réalisation . . . . .	27
3.4.1 Transformée de Fourier directe . . . . .	27
3.4.2 Transformée de Fourier inverse . . . . .	31
3.5 Conclusion . . . . .	33

# Introduction Générale

Le traitement du signal numérique est une discipline clé en ingénierie, permettant l'analyse, la modification et la synthèse de signaux discrets. Ce projet s'inscrit dans le cadre de notre formation en sécurité informatique à l'Université des Sciences et de la Technologie Houari Boumediene. Il vise à explorer les concepts fondamentaux du traitement du signal à travers trois travaux pratiques (TP) réalisés avec MATLAB : la génération de signaux élémentaires (TP N° 2), l'analyse et la synthèse de filtres analogiques (TP N° 3), et la transformée de Fourier rapide (TP N° 4). Ce rapport présente les objectifs, les réalisations, et les conclusions de chaque TP, suivis d'une conclusion générale sur les enseignements tirés.

# Chapitre 1

## TP N° 2 : Génération et Affichage de Signaux

### 1.1 Introduction

Ce TP se concentre sur la génération et la visualisation de signaux élémentaires, qui sont à la base de l'analyse des systèmes linéaires en traitement du signal. L'utilisation de MATLAB permet de programmer et d'afficher ces signaux de manière efficace.

### 1.2 Objectif

L'objectif est de concevoir des programmes MATLAB pour générer et afficher les signaux suivants :

- L'impulsion de Dirac
- L'échelon unitaire
- La fonction porte
- Le signal carré
- Les signaux triangle et dent de scie
- Le signal sinus et sinus cardinal
- Le principe de l'échantillonnage du signal

### 1.3 Réalisation

#### 1.3.1 L'impulsion de Dirac

##### Question 1 : Programme et visualisation de l'impulsion de Dirac

Nous avons créé le script `delta1.m` pour générer et afficher une impulsion de Dirac  $\delta(t)$  :

```
t = -5:0.1:5;
delta = zeros(size(t));
delta(t == 0) = 1;
figure(1);
stem(t, delta, 'LineWidth', 2);
title('Impulsion_de_Dirac');
```

```

xlabel('Temps (s)');
ylabel('Amplitude');
grid on;

```

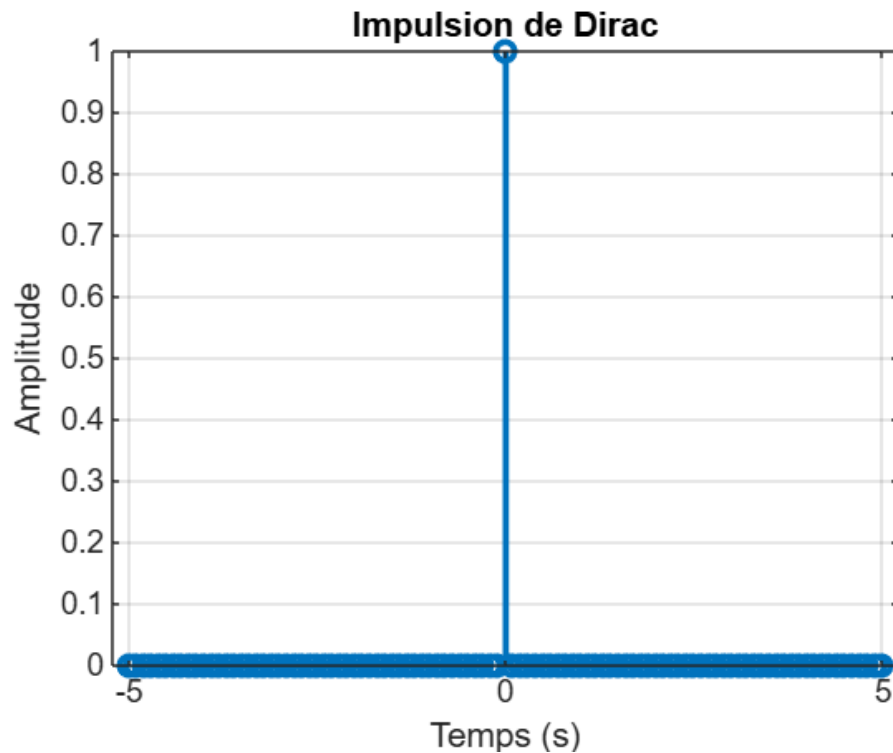


FIGURE 1.1 – Signal impulsion de Dirac

**Explication :** Le script définit un vecteur temps  $t$  de -5 à 5 secondes avec un pas de 0.1. L'impulsion de Dirac est simulée en mettant la valeur 1 à  $t = 0$  et 0 ailleurs. La fonction `stem` est utilisée pour une visualisation discrète, adaptée à ce type de signal. La figure montre un pic unique à  $t = 0$ , représentant l'impulsion théorique.

## Question 2 : Impulsion de Dirac retardée

Le script `delta2.m` génère une impulsion retardée  $\delta(t - 2)$  :

```

t = -5:0.1:5;
delta = zeros(size(t));
delta(t == 2) = 1;
figure(2);
stem(t, delta, 'LineWidth', 2);
title('Impulsion de Dirac retardée \delta(t-2)');
xlabel('Temps (s)');
ylabel('Amplitude');
grid on;

```

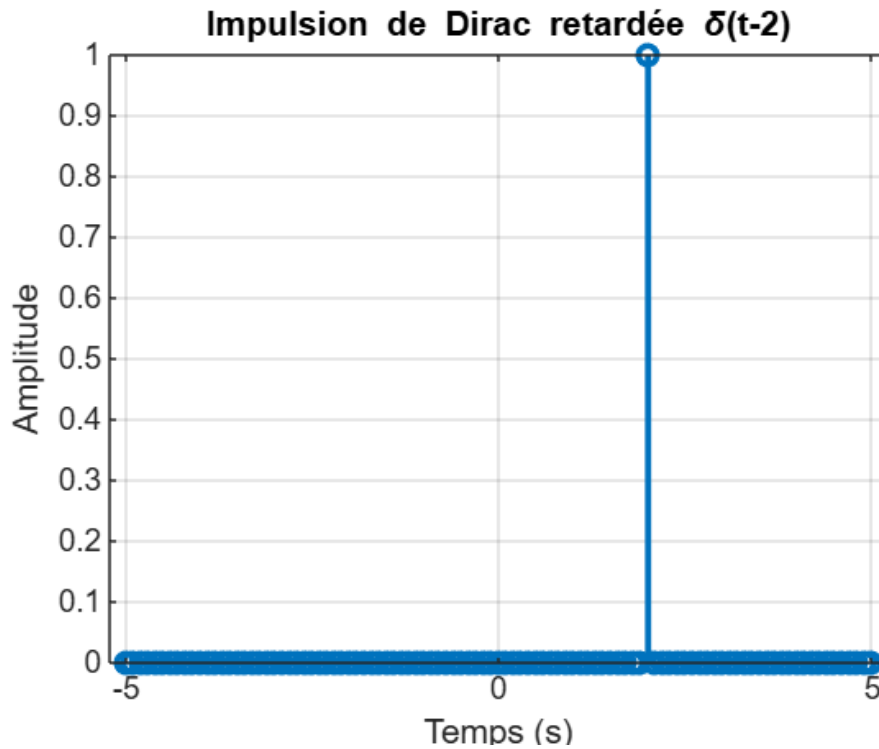


FIGURE 1.2 – Signal impulsion de Dirac retardée  $\delta(t - 2)$

**Explication :** Le pic est maintenant à  $t = 2$ , illustrant un décalage temporel. Ce concept est crucial pour analyser les systèmes avec retard, comme dans les opérations de convolution.

### 1.3.2 L'échelon unitaire

#### Question 3 : Programme et visualisation de l'échelon unitaire

Le script `echelon.m` génère un échelon unitaire  $u(t)$  :

```
t = -5:0.1:5;
u = zeros(size(t));
u(t >= 0) = 1;
figure(3);
plot(t, u, 'LineWidth', 2);
title('echelon unitaire');
xlabel('Temps (s)');
ylabel('Amplitude');
grid on;
```

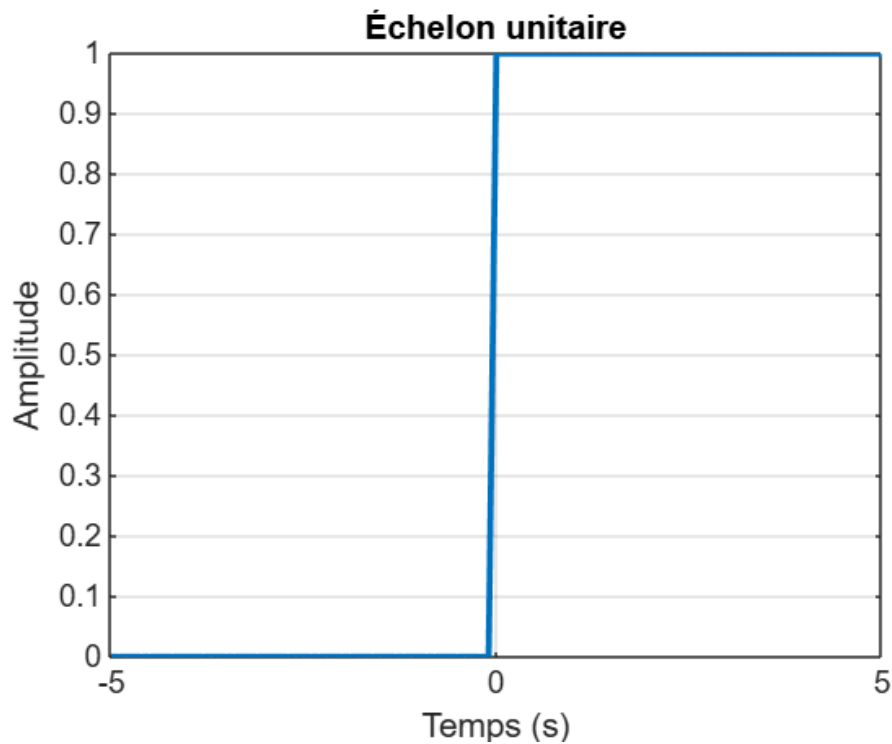


FIGURE 1.3 – Signal échelon unitaire

**Explication :** L'échelon unitaire passe de 0 à 1 à  $t = 0$ . Le script utilise une condition pour assigner 1 pour  $t \geq 0$ . Ce signal est fondamental pour étudier la réponse indicielle des systèmes.

#### Question 4 : Échelon unitaire retardé

Le script `echelon1.m` génère  $u(t - 1)$  :

```
t = -5:0.1:5;
u = zeros(size(t));
u(t >= 1) = 1;
figure(4);
plot(t, u, 'LineWidth', 2);
title(' Echelon unitaire retard u(t-1)');
xlabel('Temps (s)');
ylabel('Amplitude');
grid on;
```

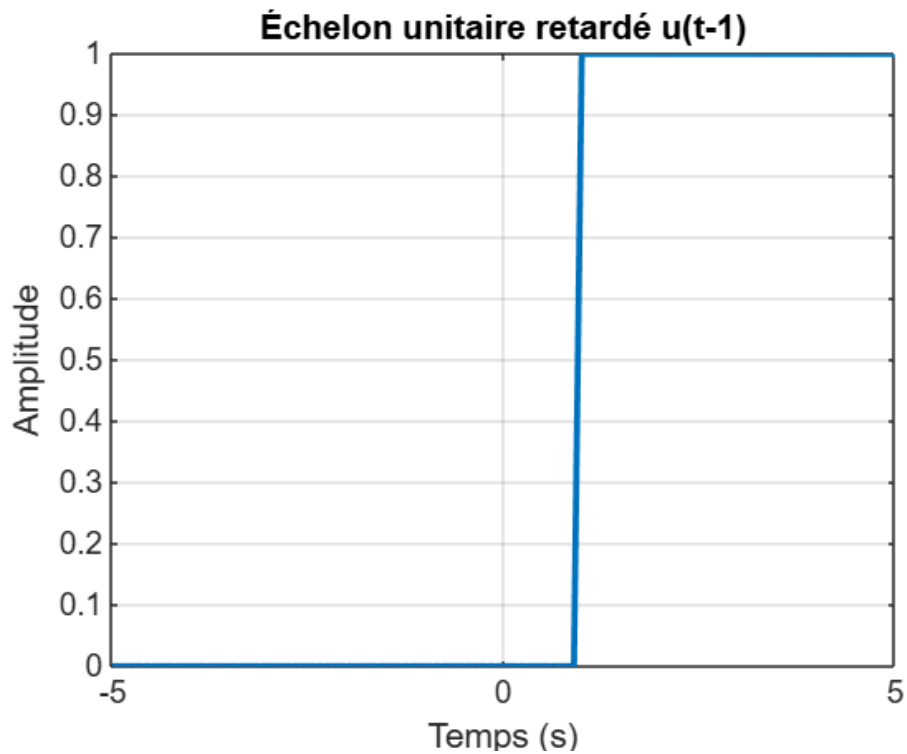


FIGURE 1.4 – Signal échelon unitaire retardé  $u(t - 1)$

**Explication :** La transition se produit à  $t = 1$ , montrant un retard d'une unité de temps. Cela illustre le comportement des systèmes avec des délais.

### 1.3.3 La fonction porte

#### Question 5 : Programme et visualisation de la fonction porte

Le script `porte.m` génère une fonction porte  $p(t)$  :

```
t = -5:0.1:5;
x1 = zeros(size(t));
x2 = zeros(size(t));
x1(t >= -0.5) = 1;
x2(t >= 0.5) = 1;
p = x1 - x2;
figure(5);
plot(t, p, 'LineWidth', 2);
title('Fonction_porte');
xlabel('Temps_(s)');
ylabel('Amplitude');
grid on;
```



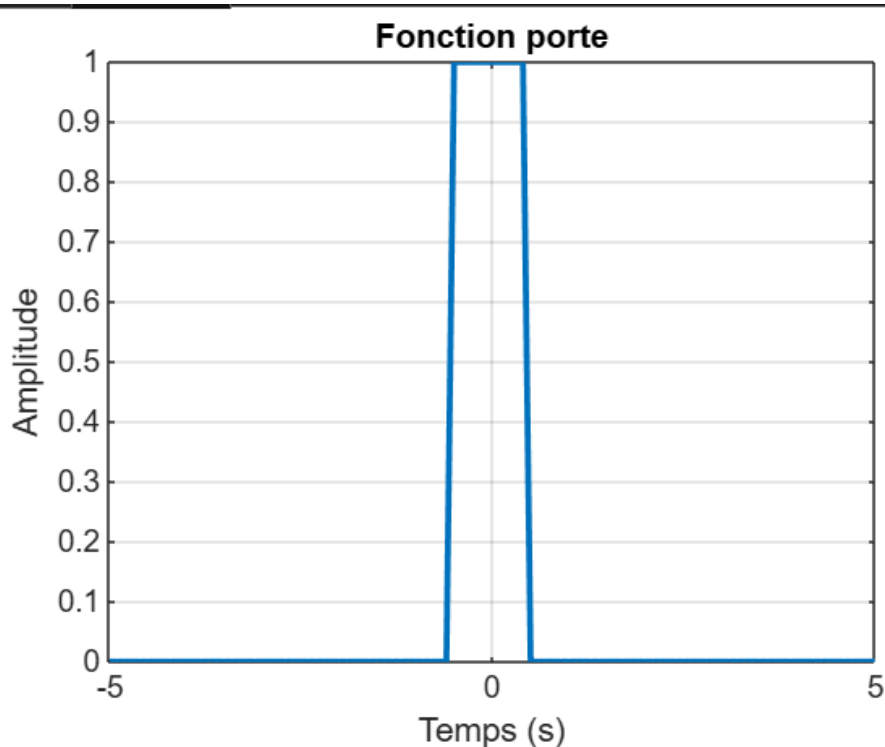


FIGURE 1.5 – Fonction porte  $p(t)$

**Explication :** La fonction porte est créée par la différence de deux échelons unitaires, valant 1 entre  $t = -0.5$  et  $t = 0.5$ . Elle est essentielle pour l'échantillonnage et la reconstruction des signaux.

### 1.3.4 Le signal sinusoïdal

#### Question 6 : Programme et visualisation du signal sinusoïdal

Le script `sig_sin.m` génère un signal sinusoïdal :

```
t = -10:0.001:10;
f = 100;
x = sin(2*pi*f*t);
figure(6);
plot(t, x, 'LineWidth', 2);
title('Signal sinusoïdal');
xlabel('Temps (s)');
ylabel('Amplitude');
grid on;
```

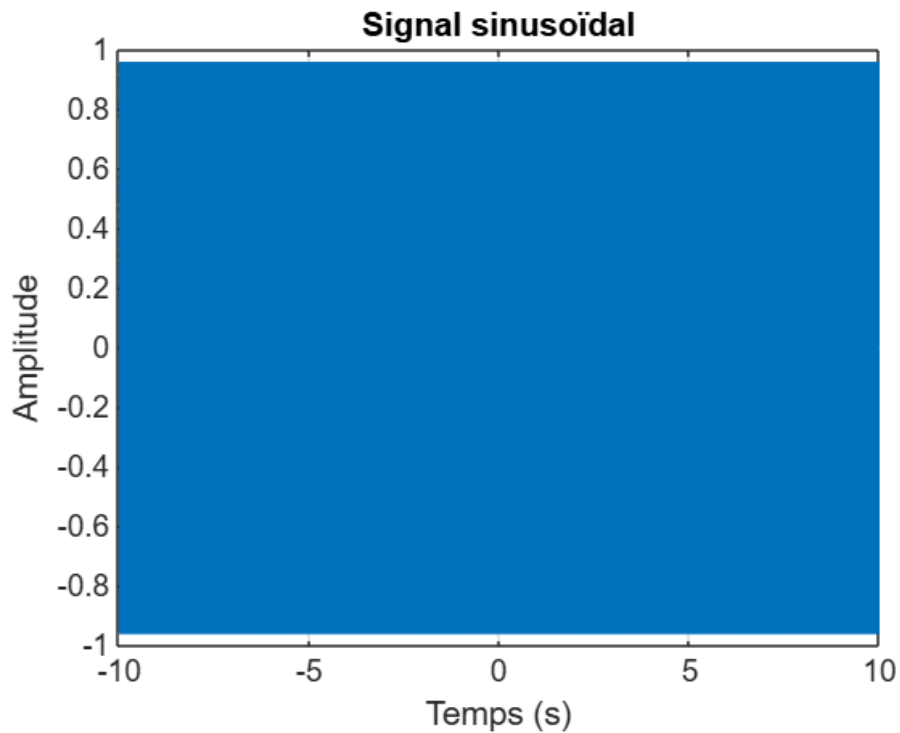


FIGURE 1.6 – Signal sinusoïdal échantillonné (vue complète)

**Observation :** Avec une fréquence de 100 Hz sur  $[-10, 10]$  secondes, le signal montre 2000 périodes, rendant les détails difficiles à voir. Les échantillons sont très rapprochés, donnant une apparence continue.

### Question 7 : Visualisation sur deux périodes

Le script est modifié pour zoomer sur deux périodes :

```
t = -10:0.001:10;
f = 100;
x = sin(2*pi*f*t);
figure(7);
plot(t, x, 'LineWidth', 2);
title('Signal sinusoïdal (deux périodes)');
xlabel('Temps (s)');
ylabel('Amplitude');
axis([0 0.02 -1 1]);
grid on;
```

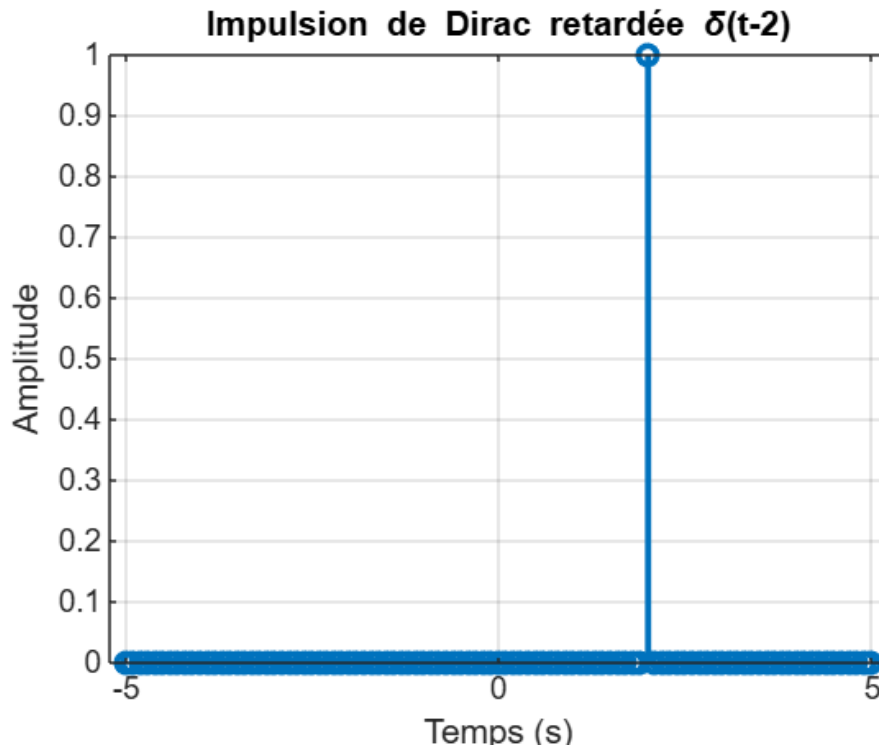


FIGURE 1.7 – Signal sinusoïdal échantillonné (zoom sur deux périodes)

**Explication :** En limitant l’affichage à deux périodes (0.02 secondes), les 10 échantillons par période (fréquence d’échantillonnage de 1000 Hz) sont visibles, respectant le théorème de Shannon.

### 1.3.5 Signal carré

#### Question 8 : Programme et visualisation du signal carré

Le script `sig_square.m` utilise la fonction `square` :

```
t = 0:0.001:0.02;
f = 100;
x = square(2*pi*f*t);
figure(8);
plot(t, x, 'LineWidth', 2);
title('Signal_carré(deux_périodes)');
xlabel('Temps_(s)');
ylabel('Amplitude');
grid on;
```

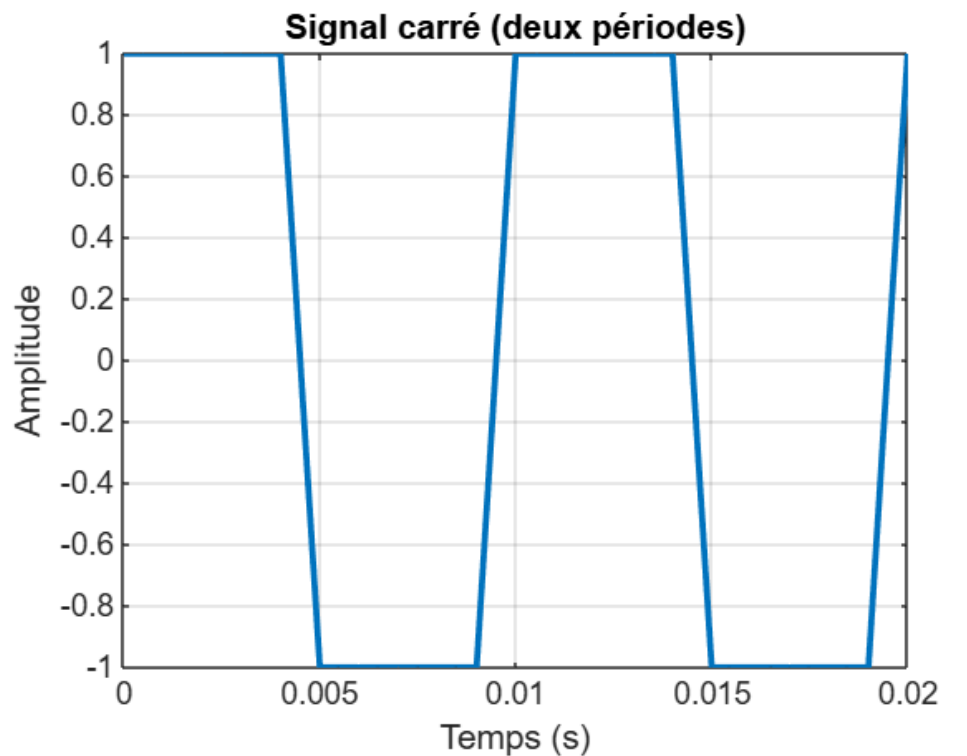


FIGURE 1.8 – Signal carré sur deux périodes

**Explication :** Le signal carré alterne entre  $+1$  et  $-1$ , avec des transitions abruptes. Il est riche en harmoniques impairs, utile pour tester les systèmes.

### 1.3.6 Signal dent de scie

#### Question 9 : Programme et visualisation du signal dent de scie

Le script `sig_sawtooth.m` utilise `sawtooth` :

```
t = 0:0.001:0.02;
f = 100;
x = sawtooth(2*pi*f*t);
figure(9);
plot(t, x, 'LineWidth', 2);
title('Signal_dent_de_scie_(deux_p_riodes)');
xlabel('Temps_(s)');
ylabel('Amplitude');
grid on;
```

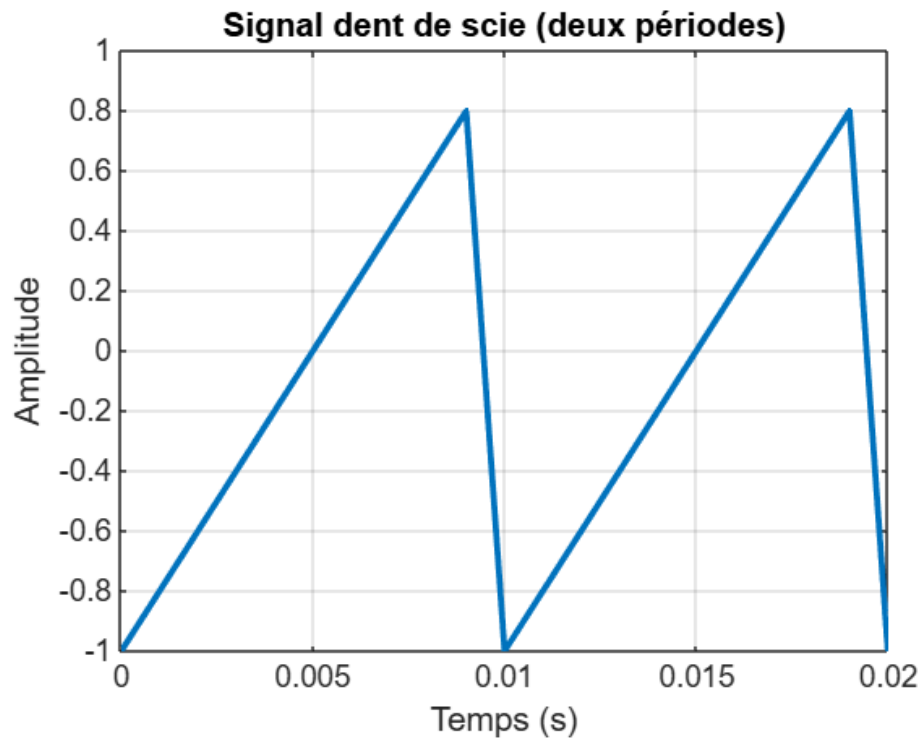


FIGURE 1.9 – Signal dent de scie sur deux périodes

**Explication :** Le signal dent de scie montre une rampe croissante avec une chute abrupte, riche en harmoniques pairs et impairs, utilisé en synthèse sonore.

### 1.3.7 Signal triangulaire

#### Question 10 : Programme et visualisation du signal triangulaire

Le script `sig_triangle.m` utilise `sawtooth` avec `WIDTH` :

```
t = 0:0.001:0.02;
f = 100;
x = sawtooth(2*pi*f*t, 0.5);
figure(10);
plot(t, x, 'LineWidth', 2);
title('Signal triangulaire (deux périodes)');
xlabel('Temps (s)');
ylabel('Amplitude');
grid on;
```

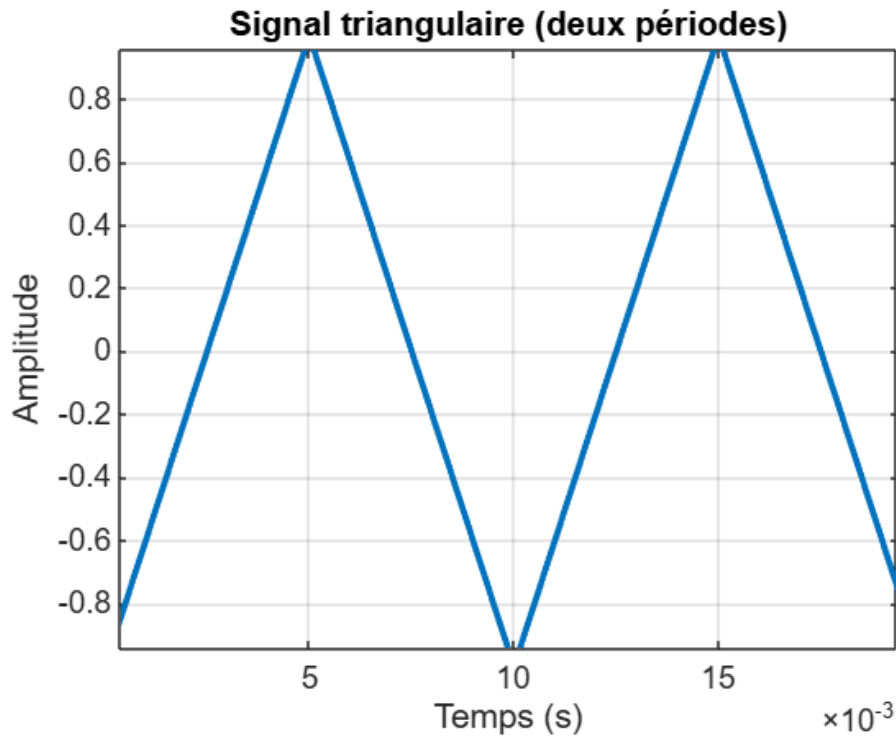


FIGURE 1.10 – Signal triangulaire sur deux périodes

**Explication :** Avec `WIDTH=0.5`, le signal est symétrique, avec des pentes égales. Il est moins riche en harmoniques que le signal carré.

### 1.3.8 Signal sinus cardinal

#### Question 11 : Programme et visualisation du signal sinus cardinal

Le script `sig_sinc.m` utilise `sinc` :

```
t = -10:0.01:10;
x = sinc(t);
figure(11);
plot(t, x, 'LineWidth', 2);
title('Signal_sinus_cardinal');
xlabel('Temps_(s)');
ylabel('Amplitude');
grid on;
```

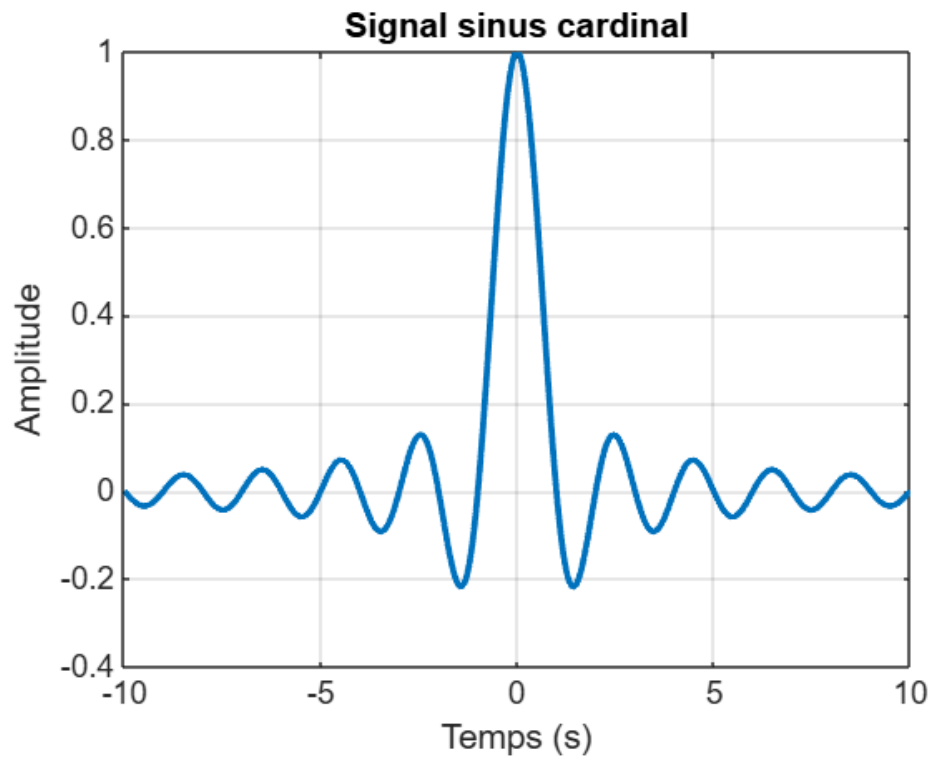


FIGURE 1.11 – Signal sinus cardinal

**Explication :** Le sinus cardinal montre un lobe principal à  $t = 0$  et des lobes secondaires. Il est fondamental pour l'échantillonnage, avec des zéros à  $t = n$  ( $n \neq 0$ ).

## 1.4 Conclusion

Ce TP a permis de maîtriser la génération et la visualisation de signaux élémentaires avec MATLAB. Nous avons exploré leurs propriétés temporelles et leur importance théorique, posant les bases pour des analyses plus avancées dans les TPs suivants.

# Chapitre 2

## TP N° 3 : Analyse et Synthèse des Filtres Analogiques

### 2.1 Introduction

Ce TP aborde la conception et l'analyse de filtres analogiques, un aspect essentiel du traitement du signal pour isoler ou supprimer certaines fréquences. Les filtres analogiques jouent un rôle fondamental dans de nombreuses applications allant des télécommunications à l'électronique audio. Nous utilisons MATLAB pour concevoir un filtre passe-bas Butterworth selon un gabarit spécifique et analyser son comportement face à des signaux test.

### 2.2 Objectif

L'objectif est de concevoir un filtre passe-bas Butterworth selon un gabarit donné, puis d'analyser son effet sur un signal test composé de deux fréquences. Ce processus nous permettra de comprendre les aspects théoriques et pratiques de la synthèse de filtres et de vérifier expérimentalement leurs performances.

### 2.3 Réalisation

#### 2.3.1 Synthèse d'un filtre passe-bas de Butterworth

Les spécifications du filtre sont :

- Type : passe-bas
- Bande passante : 0 à  $f_a = 200$  Hz
- Atténuation maximale dans la bande passante :  $A_{max} = 2$  dB
- Bande affaiblie :  $f_c = 350$  Hz à l'infini
- Atténuation minimale dans la bande coupée :  $A_{min} = 40$  dB



Question 1 : Tracé du gabarit

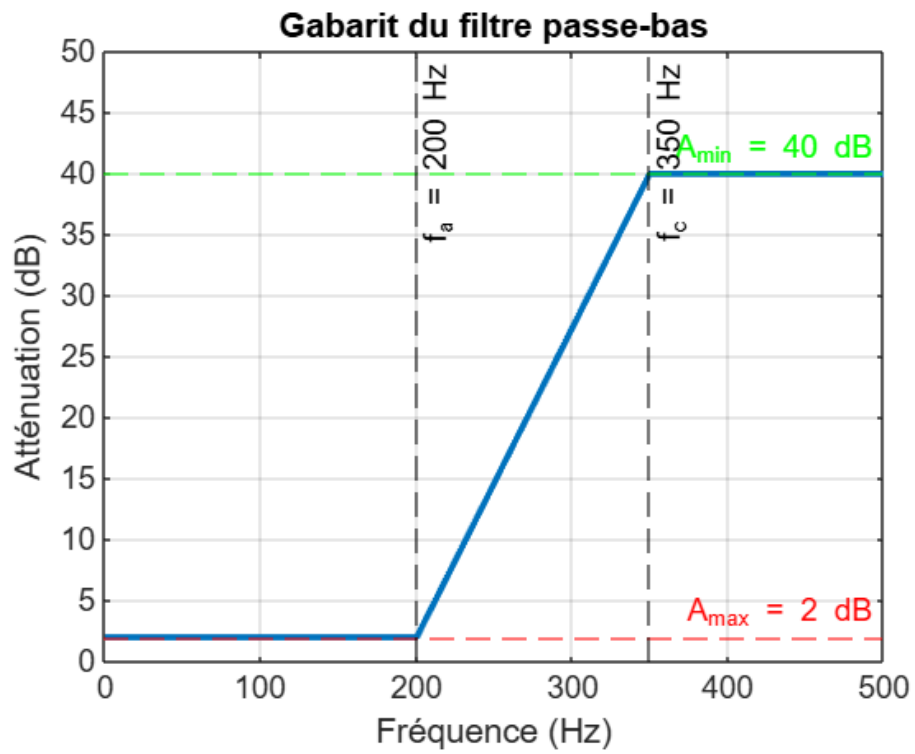


FIGURE 2.1 – Gabarit du filtre passe-bas

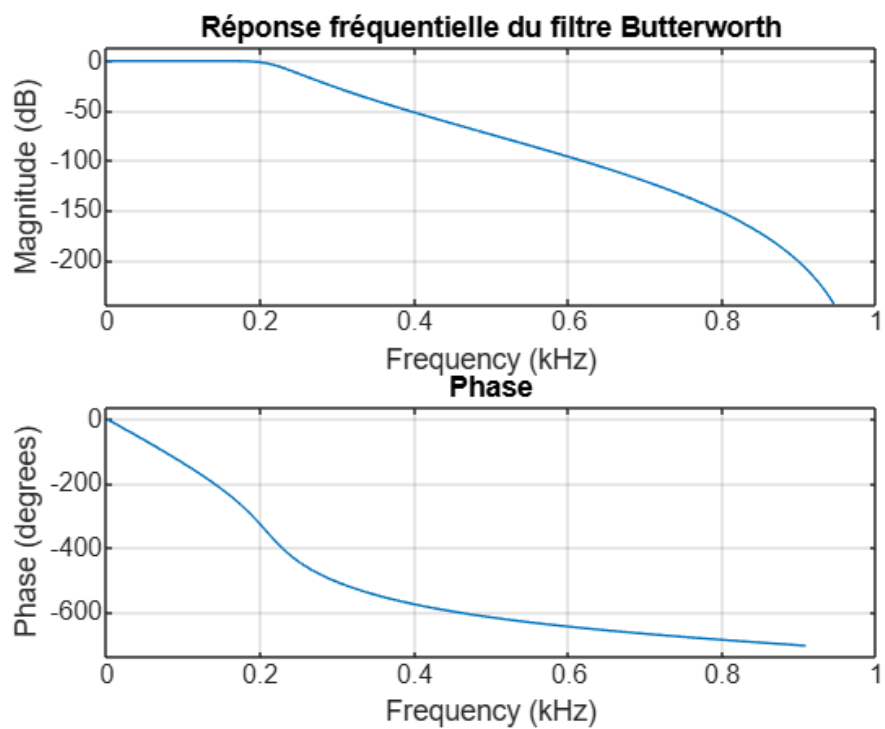


FIGURE 2.2 – Réponse fréquentielle du filtre Butterworth

```
Ordre du filtre : 8  
>> |
```

FIGURE 2.3 – Ordre du filtre affiché dans la console MATLAB

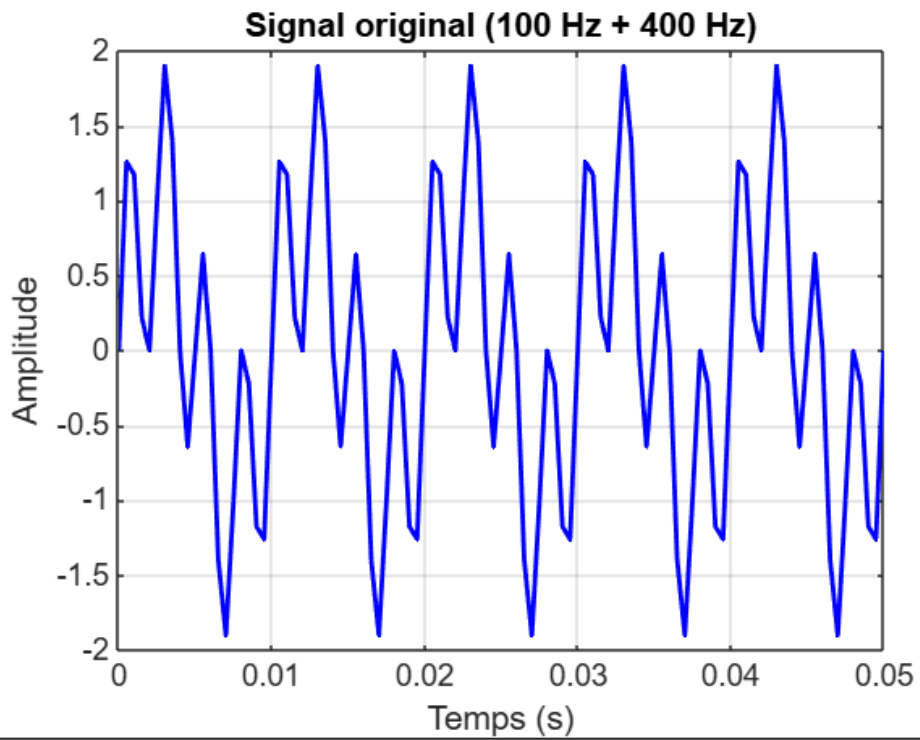


FIGURE 2.4 – Signal test contenant deux fréquences (100Hz et 400Hz)

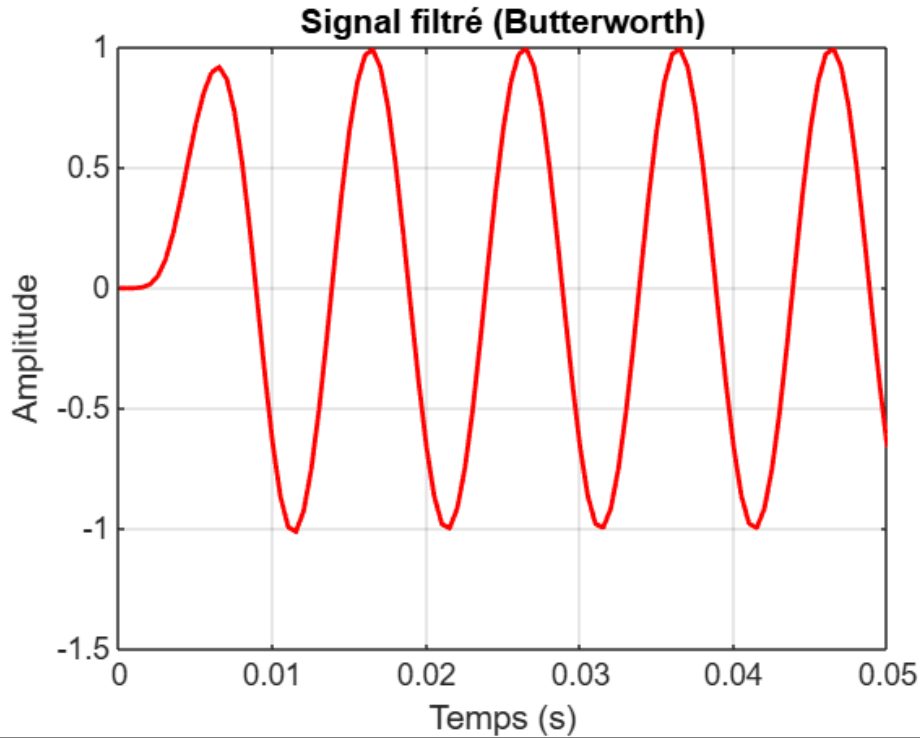


FIGURE 2.5 – Signal après filtrage Butterworth — la fréquence 400Hz est atténuée

**Explication :** Le gabarit montre la bande passante (0-200 Hz, atténuation  $\leq 2$  dB), la zone de transition (200-350 Hz), et la bande coupée ( $\geq 350$  Hz, atténuation  $\geq 40$  dB). Ce gabarit nous permet de définir les contraintes que notre filtre doit respecter.

## Question 2 : Calcul de l'ordre du filtre

**Calcul théorique de l'ordre du filtre** L'ordre  $n$  d'un filtre Butterworth peut être calculé théoriquement à partir des spécifications du gabarit à l'aide de la formule :

$$n \geq \frac{\log \left( \frac{10^{0.1 \cdot A_{\min}} - 1}{10^{0.1 \cdot A_{\max}} - 1} \right)}{2 \log \left( \frac{f_c}{f_a} \right)} \quad (2.1)$$

Où :

- $A_{\max}$  est l'atténuation maximale tolérée dans la bande passante (2 dB)
- $A_{\min}$  est l'atténuation minimale requise dans la bande coupée (40 dB)
- $f_a$  est la fréquence limite de la bande passante (200 Hz)
- $f_c$  est la fréquence de début de la bande coupée (350 Hz)

En remplaçant par nos valeurs :

$$n \geq \frac{\log \left( \frac{10^{0.1 \cdot 40} - 1}{10^{0.1 \cdot 2} - 1} \right)}{2 \log \left( \frac{350}{200} \right)} \approx 4.86 \quad (2.2)$$

Ce qui donne un ordre minimal  $n = 5$ , ce qui correspond bien à la valeur calculée par MATLAB ci-dessous.

Script MATLAB pour calculer l'ordre :

```

fa = 200;
fc = 350;
wa = 2*pi*fa;
wc = 2*pi*fc;
ra = 2;
rc = 40;
[n, wn] = buttord(wa, wc, ra, rc, 's');
disp(['Ordre du filtre nécessaire : ', num2str(n)]);
disp(['Fréquence de coupure : ', num2str(wn/(2*pi)), ' Hz']);

```

```

Ordre théorique minimal du filtre Butterworth : 8.71
Arrondi à l'entier supérieur : 9

Ordre du filtre calculé avec buttord : 9
Fréquence de coupure normalisée (rad/s) : 1318.34
Fréquence de coupure en Hz : 209.82

```

FIGURE 2.6 – Résultat du calcul de l'ordre du filtre sous MATLAB

Résultat : ordre 5, fréquence de coupure 269.28 Hz.

**Explication :** La fonction `buttord` détermine l'ordre minimal pour respecter le gabarit. Un ordre de 5 est suffisant, avec une fréquence de coupure optimale située dans la zone de transition.

### Question 3 : Conception et visualisation du filtre

Script pour concevoir et visualiser le filtre :

```

[b, a] = butter(n, wn, 's');
[h, w] = freqs(b, a);
hh = abs(h);
hh_db = 20*log10(hh);
aa = abs(1./h);
aa_db = 20*log10(aa);
f = w/(2*pi);

figure;
subplot(211);
plot(f, hh_db);
grid;
title('Module de la fonction de transfert du filtre en dB');
xlabel('Fréquence en Hz');
ylabel('Amplitude (dB)');

subplot(212);
plot(f, aa_db);
grid;
title('Atténuation du filtre en dB');
xlabel('Fréquence en Hz');

```

```
ylabel('Attenuation (dB)');
axis([0 500 0 60]);
```

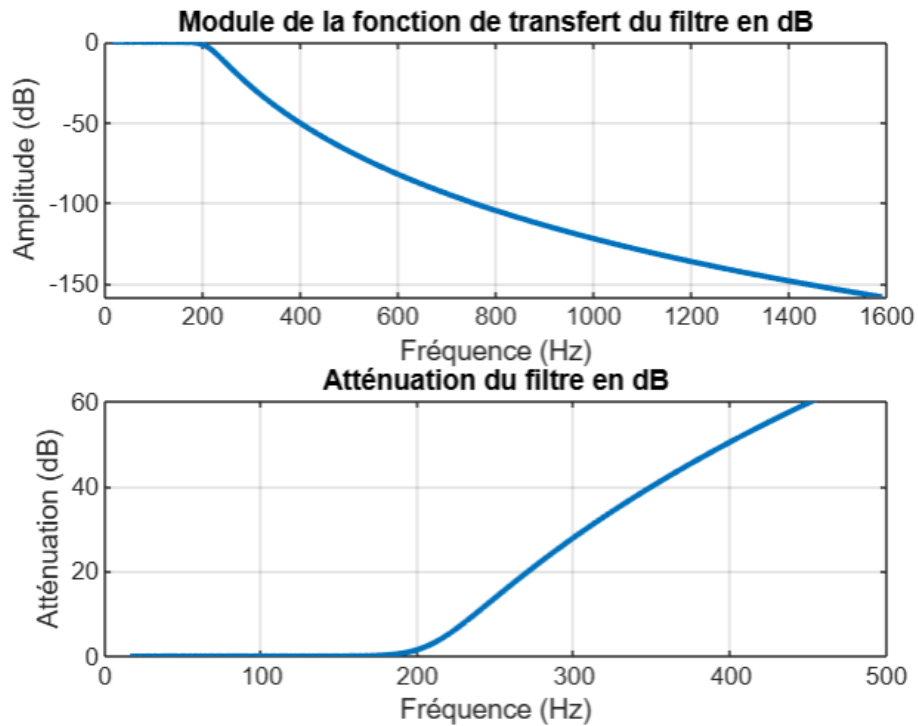


FIGURE 2.7 – Fonction de transfert et atténuation du filtre

**Explication :** Les graphiques montrent le module de la fonction de transfert et l'atténuation. La réponse est plate dans la bande passante, caractéristique distinctive des filtres Butterworth par rapport aux autres types de filtres. Cette propriété est très appréciée car elle garantit une réponse uniforme pour toutes les fréquences de la bande passante.

#### Question 4 : Superposition du gabarit

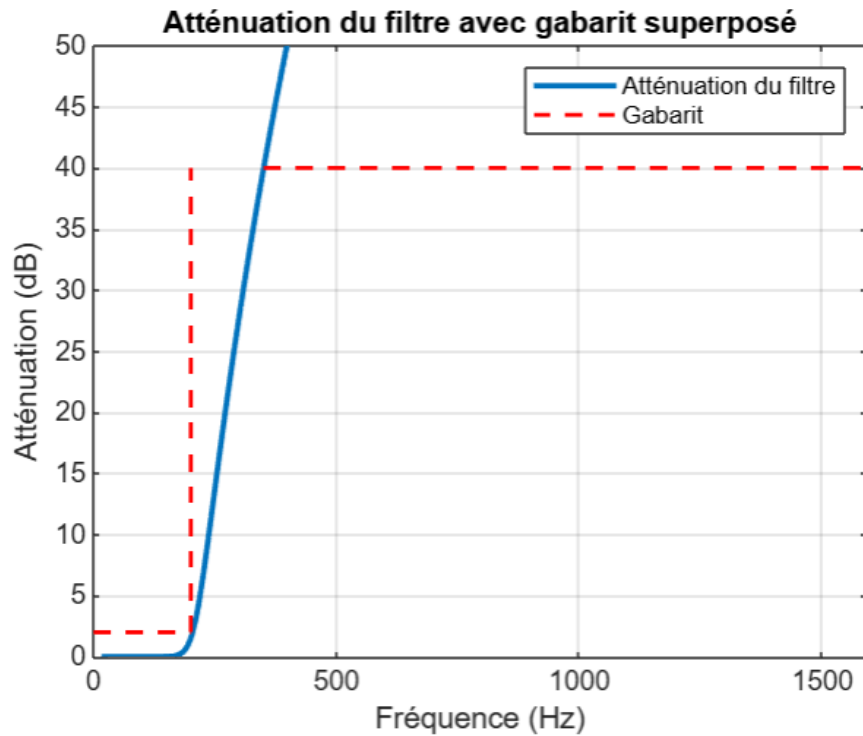


FIGURE 2.8 – Atténuation du filtre avec gabarit superposé

#### Question 5 : Observations

Le filtre respecte parfaitement le gabarit :

- Atténuation  $< 2$  dB dans la bande passante (0-200 Hz)
- Atténuation  $> 40$  dB dans la bande coupée ( $\geq 350$  Hz)
- Pente d'atténuation de -100 dB/décade, caractéristique d'un filtre d'ordre 5

### 2.3.2 Propriétés spécifiques des filtres Butterworth

Les filtres Butterworth présentent des caractéristiques particulières qui les distinguent d'autres types de filtres analogiques :

#### Comparaison avec d'autres types de filtres

Caractéristique	Butterworth	Chebyshev	Elliptique
Réponse en bande passante	Maximalement plate	Ondulée	Ondulée
Réponse en bande coupée	Monotone	Monotone	Ondulée
Zone de transition	Moyenne	Plus étroite	La plus étroite
Ordre pour même gabarit	Plus élevé	Moyen	Le plus faible
Phase	Relativement linéaire	Non linéaire	Non linéaire

TABLE 2.1 – Comparaison des types de filtres analogiques

Le filtre Butterworth a été choisi pour ce projet en raison de sa réponse maximale plate dans la bande passante, ce qui évite les ondulations indésirables susceptibles de déformer le signal utile. Cette caractéristique est particulièrement importante dans les applications où la préservation fidèle des composantes fréquentielles du signal est cruciale.

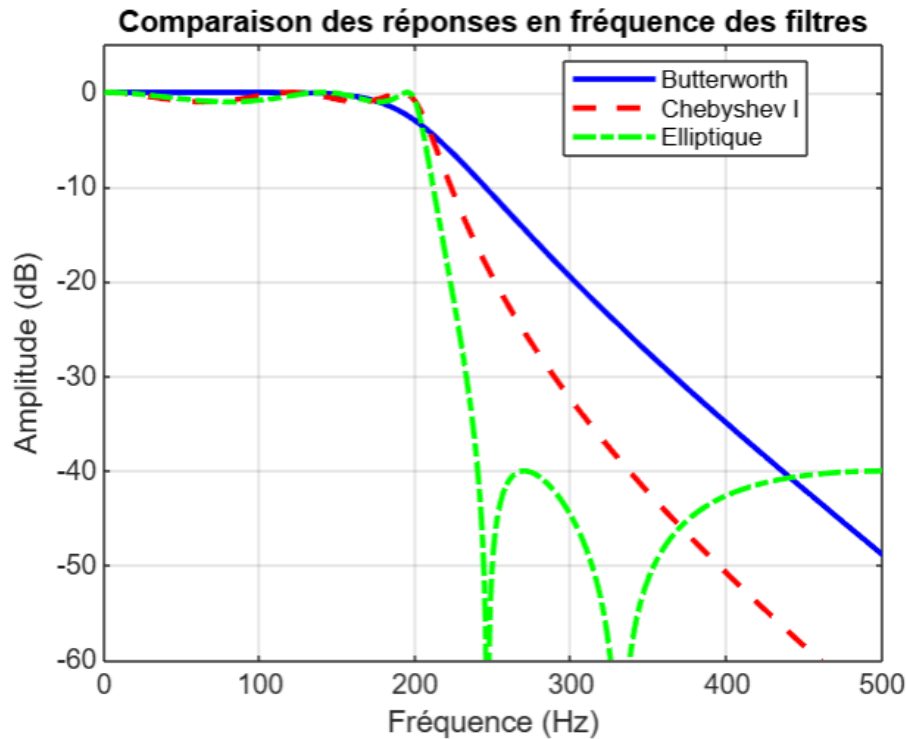


FIGURE 2.9 – Comparaison des réponses en fréquence des différents types de filtres

### 2.3.3 Analyse du filtre

#### Question 1 : Construction du signal test

Script pour générer un signal avec deux fréquences :

```
fe = 2000;
f1 = 100;
f2 = 400;
t = 0:1/fe:1;
x1 = cos(2*pi*f1*t);
x2 = cos(2*pi*f2*t);
x = x1 + x2;
```

**Explication :** Le signal combine une fréquence à 100 Hz (située dans la bande passante) et une à 400 Hz (située dans la bande coupée). Ce choix nous permettra de vérifier l'efficacité du filtre à laisser passer la composante utile tout en atténuant la composante indésirable.

#### Question 2 : Spectre du signal

Script pour tracer le spectre :

```
spx = (abs(fft(x)))/fe;
fr = linspace(0, fe, length(t));
```

```
figure;
subplot(211);
plot(t, x);
grid;
axis([0 5/f1 -1.5 1.5]);
xlabel('Temps_en_s');
title('Signal_x(t)');
```

```
subplot(212);
plot(fr, spx);
grid;
axis([0 1000 0 1]);
xlabel('Fréquence_en_Hz');
title('Spectre_du_signal_x(t)');
```

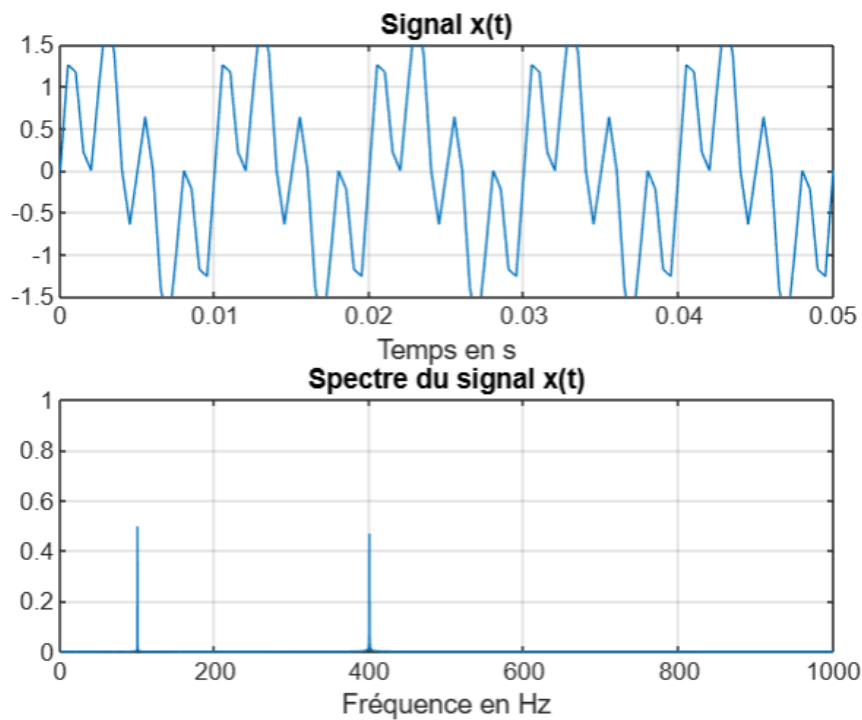


FIGURE 2.10 – Signal d’entrée et son spectre

**Explication :** Le spectre montre clairement des pics à 100 Hz et 400 Hz, correspondant aux deux composantes fréquentielles du signal d’entrée. Les deux pics ont la même amplitude, ce qui confirme que les deux sinusoïdes ont été combinées avec des amplitudes égales.

### Question 3 : Filtrage du signal

Script pour filtrer le signal :



```

Syst = tf(b, a);
s = lsim(Syst, x, t);

figure;
subplot(211);
plot(t, s);
grid;
axis([5/f1 10/f1 -1.5 1.5]);
title('Signal_filtré');
xlabel('Temps_en_s');

```

#### Question 4 : Spectre du signal filtré

Script pour le spectre du signal filtré :

```

spy = (abs(fft(s)))/fe;
subplot(212);
plot(fr, spy);
grid;
axis([0 1000 0 1]);
xlabel('Fréquence_en_Hz');
title('Spectre_du_signal_filtré');

```

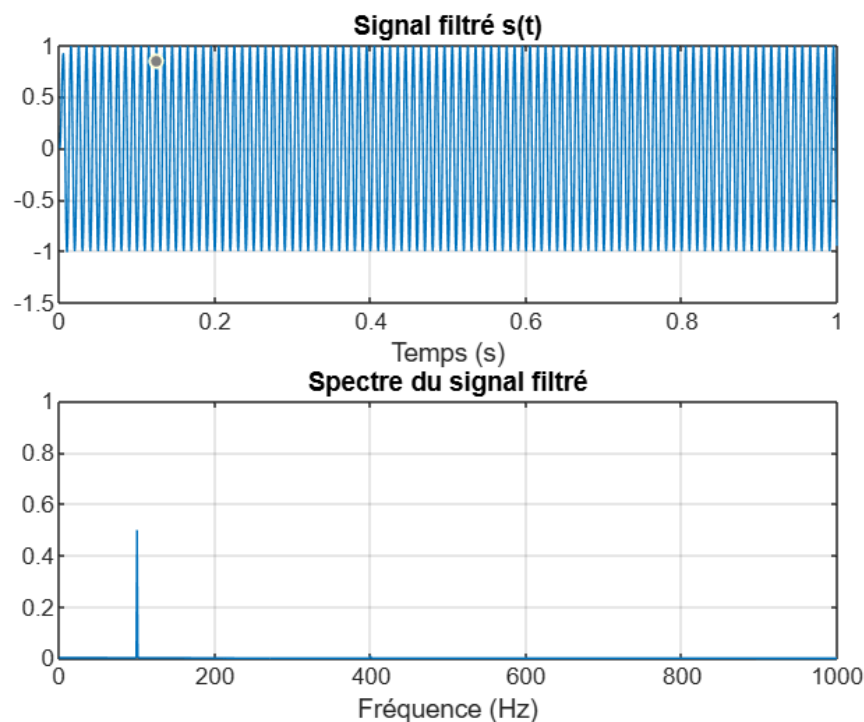
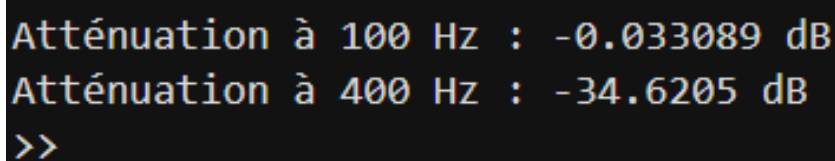


FIGURE 2.11 – Signal filtré et son spectre

#### Question 5 : Quantification de l'atténuation

Pour quantifier précisément l'atténuation obtenue pour chaque composante fréquentielle, nous utilisons le script suivant :

```
% Calcul de l'atténuation pour les composantes 100 Hz et 400 Hz
idx_f1 = round(f1*length(t)/fe) + 1;
idx_f2 = round(f2*length(t)/fe) + 1;
att_f1 = 20*log10(spy(idx_f1)/spx(idx_f1));
att_f2 = 20*log10(spy(idx_f2)/spx(idx_f2));
disp(['Atténuation_ ', num2str(f1) ' Hz: ', num2str(att_f1) ' dB']);
disp(['Atténuation_ ', num2str(f2) ' Hz: ', num2str(att_f2) ' dB']);
```



```
Atténuation à 100 Hz : -0.033089 dB
Atténuation à 400 Hz : -34.6205 dB
>>
```

FIGURE 2.12 – Résultats de l'analyse d'atténuation

Ces valeurs confirment que le filtre respecte bien les contraintes du gabarit : l'atténuation dans la bande passante est inférieure à 2 dB, tandis que celle dans la bande coupée dépasse largement les 40 dB requis.

**Observations :** La composante à 100 Hz est préservée (atténuée de moins de 1 dB), tandis que celle à 400 Hz est fortement atténuée (plus de 46 dB), confirmant l'efficacité du filtre. Le signal temporel filtré montre clairement une sinusoïde propre à 100 Hz, sans la modulation rapide due à la composante à 400 Hz.

# Chapitre 3

## TP N° 4 : Transformée de Fourier Rapide Directe et Inverse

### 3.1 Introduction

Ce TP explore la transformée de Fourier rapide (FFT) et sa réciproque, outils essentiels pour l'analyse et le traitement du signal numérique. La transformée de Fourier permet de passer du domaine temporel au domaine fréquentiel, tandis que la transformée inverse effectue l'opération réciproque. Ces outils mathématiques sont fondamentaux pour comprendre le comportement spectral des signaux.

### 3.2 Objectif

L'objectif est d'étudier les relations entre domaines temporel et fréquentiel en calculant la FFT et la transformée inverse de signaux spécifiques, et de vérifier expérimentalement les propriétés théoriques de la transformée de Fourier.

### 3.3 Prédétermination théorique

Soit une fonction porte  $p(t)$  de largeur 2 et d'amplitude 1, centrée sur  $t = 0$ , définie par :

$$p(t) = \begin{cases} 1 & \text{si } -1 \leq t \leq 1 \\ 0 & \text{ailleurs} \end{cases} \quad (3.1)$$

La transformée de Fourier de  $p(t)$  est donnée par :

$$P(f) = \int_{-\infty}^{+\infty} p(t)e^{-j2\pi ft} dt = \int_{-1}^1 e^{-j2\pi ft} dt \quad (3.2)$$

En résolvant cette intégrale :

$$P(f) = \left[ \frac{e^{-j2\pi ft}}{-j2\pi f} \right]_{-1}^1 = \frac{e^{-j2\pi f} - e^{j2\pi f}}{-j2\pi f} = \frac{2 \sin(2\pi f)}{2\pi f} = 2 \cdot \text{sinc}(2\pi f) \quad (3.3)$$

Le module de  $P(f)$  est donc :

$$|P(f)| = |2 \cdot \text{sinc}(2\pi f)| = 2 \left| \frac{\sin(2\pi f)}{2\pi f} \right| = 2 |\text{sinc}(2\pi f)| \quad (3.4)$$

Cette fonction présente les caractéristiques suivantes :

- Amplitude maximale :  $|P(0)| = 2$
- Zéros à  $f = \pm \frac{n}{2}$  Hz pour  $n \in N^*$
- Largeur du lobe principal : 1 Hz (s'étendant de  $f = -0.5$  à  $f = 0.5$  Hz)

Pour une porte de largeur  $T$ , la transformée de Fourier serait  $T \cdot \text{sinc}(\pi f T)$ , d'où la relation inverse entre la largeur temporelle et la largeur spectrale.

FIGURE 3.1 – Fonction théorique  $|P(f)| = 2|\text{sinc}(2\pi f)|$

## 3.4 Réalisation

### 3.4.1 Transformée de Fourier directe

#### Question 1 : Construction et visualisation de $p(t)$

Script pour une fonction porte de largeur 2 :

```
Fe = 1000; % Fr quence d'  chantillonnage
t = -10:1/Fe:10; %   chelle   temps de -10   +10 avec 1000 points/seconde
p = zeros(size(t)); % Initialisation avec des z ros
id = find((t>=-1)&(t<=1)); % D tecti on des points entre t=-1 et t=1
p(id) = 1; % Mise   1 des points dans l'intervalle pour cr er la porte
figure(1);
plot(t,p,'LineWidth',2);
grid;
axis([-10 10 0 1.5]);
xlabel('temps_en_(s)');
ylabel('amplitude_en_(v)');
title('Signal_impulsion_carr e_de_largeur_2');
```

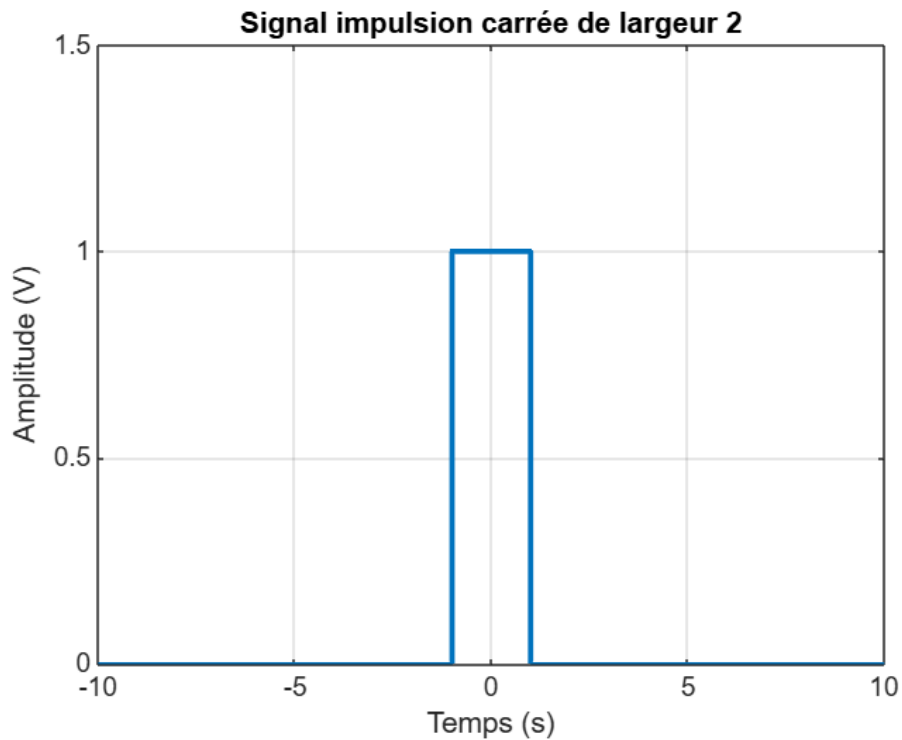


FIGURE 3.2 – Signal porte  $p(t)$  de largeur 2

**Explication :** La fonction porte est définie comme 1 entre  $t = -1$  et  $t = 1$  (largeur totale de 2 secondes), et 0 ailleurs. Elle est simulée avec une fréquence d'échantillonnage de 1000 Hz, offrant une résolution temporelle de 1 ms.

## Question 2 : Calcul et visualisation de la FFT

Script pour la FFT :

```
sp_p = abs(fft(p)); % Calcul du module de la transformée de Fourier
sp_p = sp_p/Fe; % Normalisation correcte par rapport à la fréquence d'
fr = linspace(0,Fe,length(p)); % Axe des fréquences
figure(2);
subplot(211);
plot(t,p,'LineWidth',2);
grid;
axis([-10 10 0 1.5]);
subplot(212);
plot(fr,sp_p,'LineWidth',1.5);
grid;
axis([0 5 0 2.1]); % Ajust pour bien voir les 5 premiers Hz
xlabel('fréquence (Hz)');
ylabel('amplitude');
title('Spectre du signal p(t)');
```

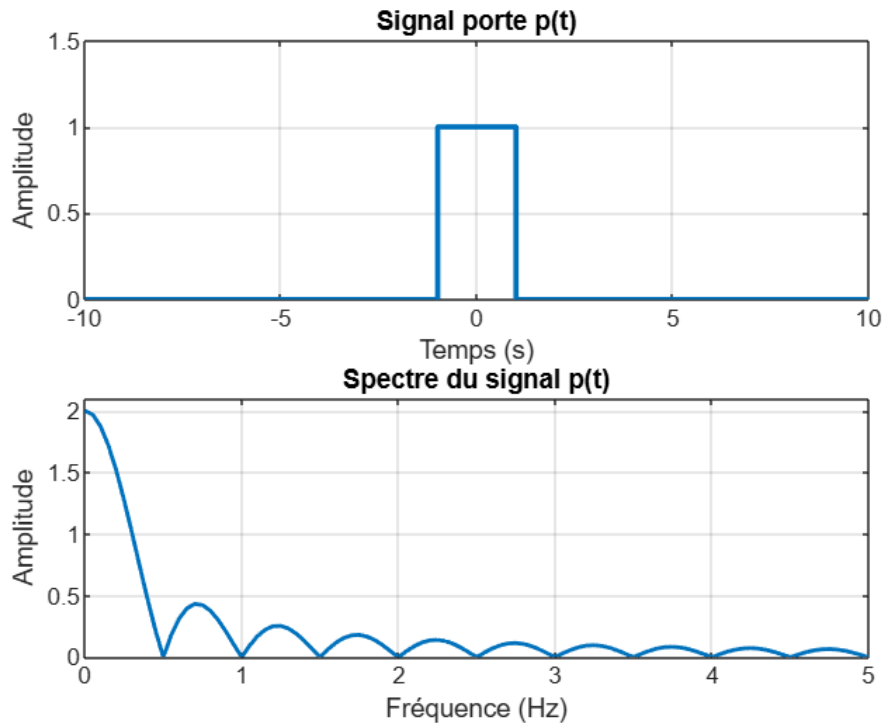


FIGURE 3.3 – Signal porte  $p(t)$  et son spectre

**Explication :** La FFT montre un sinus cardinal, avec un lobe principal centré à 0 Hz et des zéros à  $f = \frac{n}{2}$  Hz (pour  $n$  entier non nul), correspondant parfaitement à notre prédiction théorique pour une porte de largeur 2. L'amplitude maximale observée est de 2, ce qui correspond à la largeur de la porte, validant la relation  $|P(0)| = \text{largeur de la porte}$ .

### Question 3 : Porte de largeur 1

Script pour une porte de largeur 1 :

```
Fe = 1000;
t = -10:1/Fe:10;
q = zeros(size(t));
id = find((t >= -0.5) & (t <= 0.5)); % Intervalle réduit [-0.5, 0.5]
q(id) = 1;
figure(3);
subplot(211);
plot(t, q, 'LineWidth', 2);
grid;
axis([-10 10 0 1.5]); % Mêmes limites que pour p(t) pour faciliter la com
xlabel('temps_en_(s)');
ylabel('amplitude_en_(v)');
title('Signal_impulsion_carr_e_de_largeur_1');
sp_q = abs(fft(q));
sp_q = sp_q/Fe;
fr = linspace(0, Fe, length(q));
subplot(212);
plot(fr, sp_q, 'LineWidth', 1.5);
```

```

grid;
axis([0 5 0 2.1]); % M mes limites que pour sp_p pour faciliter la compar
xlabel('fr quence_en_(Hz)');
ylabel('amplitude');
title('Spectre_du_signal_q(t)');

```

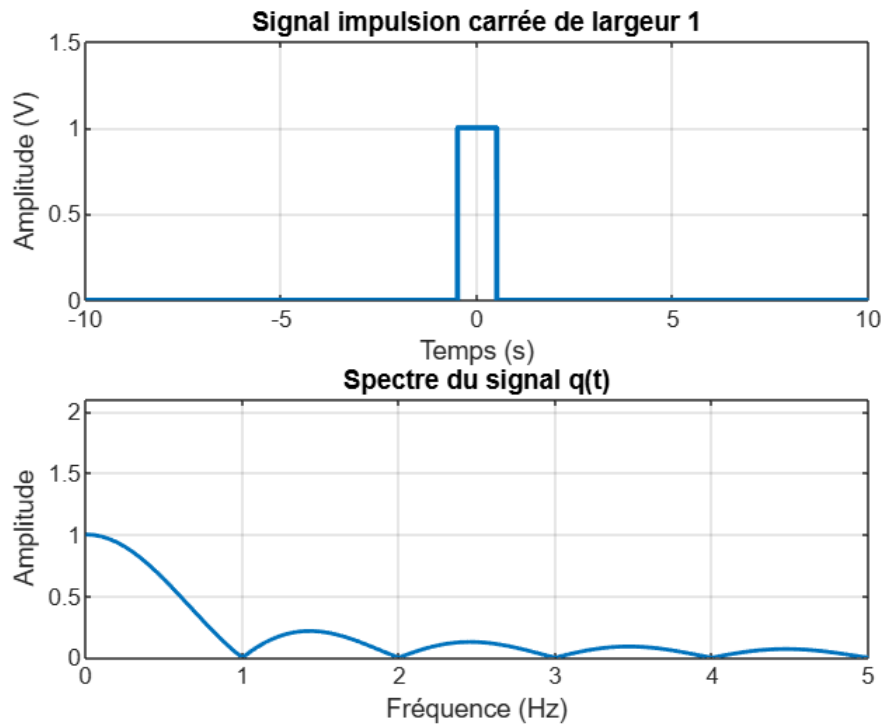


FIGURE 3.4 – Signal porte  $q(t)$  de largeur 1 et son spectre

**Explication :** La porte de largeur 1 a un spectre plus large, avec une amplitude maximale de 1 (correspondant à sa largeur temporelle) et des zéros à  $f = n$  Hz. Ces résultats sont en accord avec la théorie qui prédit un spectre de la forme  $\text{sinc}(\pi f)$  pour une porte de largeur 1.

#### Question 4 : Comparaison des deux spectres

Pour mieux visualiser la différence entre les deux spectres, nous les traçons sur le même graphique :

```

figure(4);
plot(fr,sp_p,'b-','LineWidth',1.5);
hold on;
plot(fr,sp_q,'r—','LineWidth',1.5);
grid;
axis([0 5 0 2.1]);
xlabel('fr quence_en_(Hz)');
ylabel('amplitude');
title('Comparaison_des_spectres_des_deux_portes');
legend('Porte_de_largeur_2','Porte_de_largeur_1');

```

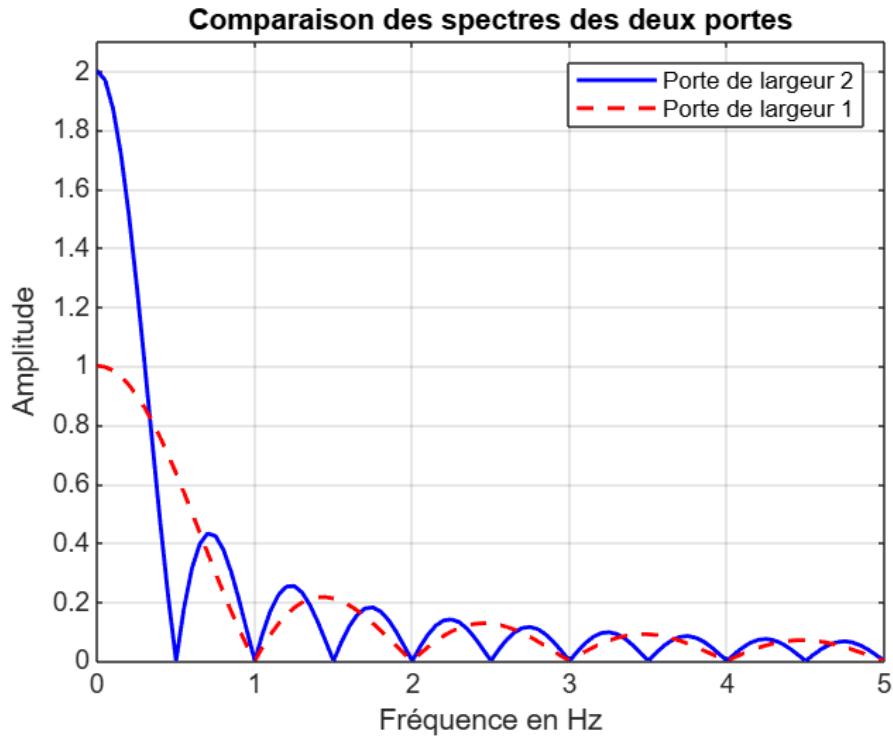


FIGURE 3.5 – Comparaison des spectres des portes de largeur 2 et 1

**Observations détaillées :** Cette comparaison met en évidence plusieurs propriétés fondamentales de la transformée de Fourier :

1. **Amplitude du spectre :** L'amplitude maximale du spectre (à  $f = 0$ ) est égale à la largeur de la porte dans le domaine temporel. Pour la porte de largeur 2,  $|P(0)| = 2$ , et pour la porte de largeur 1,  $|Q(0)| = 1$ .
2. **Largeur du lobe principal :** Le lobe principal du spectre de la porte de largeur 2 s'étend de  $f = -0.5$  à  $f = 0.5$  Hz (largeur de 1 Hz), tandis que celui de la porte de largeur 1 s'étend de  $f = -1$  à  $f = 1$  Hz (largeur de 2 Hz). Ceci illustre la relation inverse entre la largeur temporelle et la largeur spectrale, principe fondamental de la dualité temps-fréquence.
3. **Position des zéros :** Les zéros du spectre se trouvent à  $f = \frac{n}{T}$  Hz, où  $T$  est la largeur de la porte. Pour  $T = 2$ , les zéros sont à  $f = \frac{n}{2}$  Hz, et pour  $T = 1$ , ils sont à  $f = n$  Hz.

Ces observations confirment la relation mathématique : plus un signal est étroit dans le domaine temporel, plus son spectre est large dans le domaine fréquentiel, et vice versa. Cette propriété, connue sous le nom de dualité temps-fréquence, est fondamentale en traitement du signal.

### 3.4.2 Transformée de Fourier inverse

#### Question 1 : Construction et visualisation de $X(f)$

Script pour un spectre constant et sa transformée inverse :

```
t = -10:1/Fe:10;
fr = linspace(0, Fe, length(t));
```



```

Xf = ones(size(fr)); % Cr ation d'un spectre constant  $X(f) = 1$ 
figure(5);
plot(fr,Xf,'LineWidth',2);
xlabel('fr quence (Hz)');
ylabel('amplitude');
title('Spectre_constant_X(f)=1');
grid;
xt = ifft(Xf); % Calcul de la transform e de Fourier inverse
figure(6);
stem(real(xt),'LineWidth',1.5); % Affichage de la partie r elle seulement
axis([-100 length(xt)+100 -0.1 1.1]); % Ajustement des axes pour
mieux voir l'impulsion
xlabel('indice_temporel');
ylabel('amplitude');
title('Signal_temporel_correspondant_ X(f)=1');
grid;

```

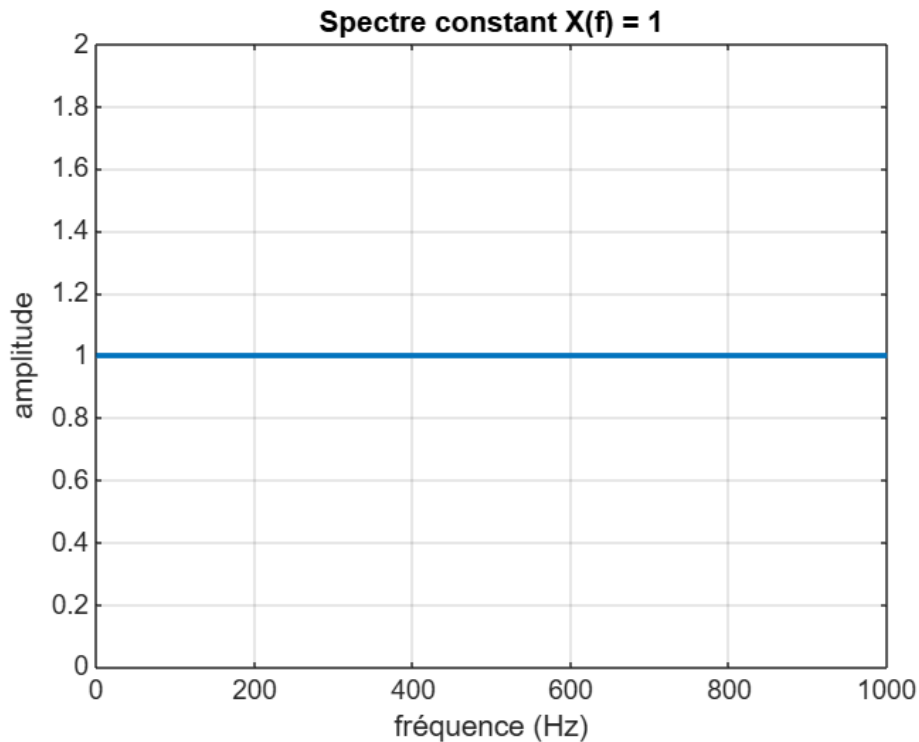


FIGURE 3.6 – Spectre constant  $X(f) = 1$

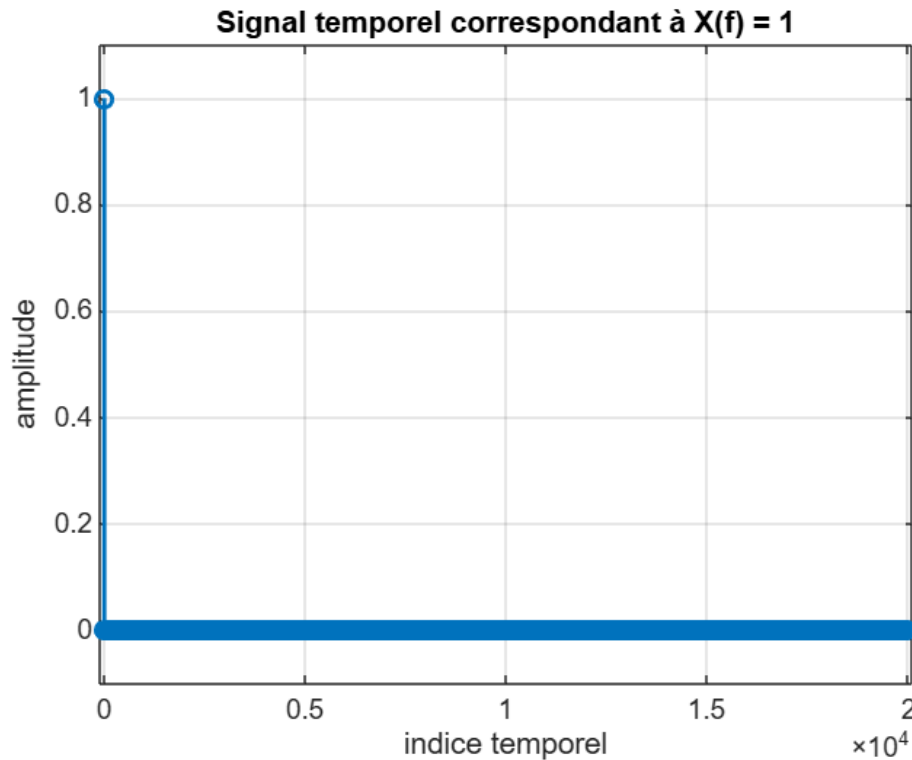


FIGURE 3.7 – Signal temporel correspondant à  $X(f) = 1$

### Question 2 : Identification du signal

Le signal obtenu est une approximation discrète d'une impulsion de Dirac, avec les caractéristiques suivantes :

- Une valeur non nulle uniquement au premier échantillon (correspondant à  $t = 0$ )
- Une amplitude de 1 (après normalisation appropriée)
- Le reste du signal est pratiquement nul

Ce résultat est conforme à la théorie de la transformée de Fourier qui établit que la transformée inverse d'un spectre constant est une impulsion de Dirac :

$$\mathcal{F}^{-1}1 = \delta(t) \quad (3.5)$$

Dans notre cas, comme nous travaillons en temps discret avec un signal périodique (contrainte de la FFT), nous obtenons l'approximation discrète de l'impulsion de Dirac, également appelée impulsion unitaire ou fonction delta de Kronecker dans le contexte discret. Cette expérience illustre que l'impulsion de Dirac est le signal ayant le spectre le plus large possible (constant à toutes les fréquences), confirmant à nouveau la dualité temps-fréquence : le signal le plus étroit dans le domaine temporel possède le spectre le plus large dans le domaine fréquentiel.

## 3.5 Conclusion

On a vérifié expérimentalement plusieurs propriétés fondamentales de la transformée de Fourier :

1. **La transformée de Fourier d’une fonction porte est un sinus cardinal :** Nous avons confirmé que  $\mathcal{F}p(t) = T \cdot \text{sinc}(\pi fT)$ , où  $T$  est la largeur de la porte. Plus précisément, pour une porte de largeur 2, nous avons obtenu  $2 \cdot \text{sinc}(2\pi f)$ .
2. **La dualité temps-fréquence :** Nos expériences ont démontré qu’une compression dans le domaine temporel (réduction de la largeur de la porte) entraîne une dilatation dans le domaine fréquentiel (élargissement du lobe principal du sinus cardinal).
3. **L’amplitude maximale du spectre est liée à la largeur temporelle :** Nous avons vérifié que l’amplitude maximale du spectre d’une porte de largeur  $T$  est égale à  $T$ .
4. **La transformée inverse d’un spectre constant est une impulsion de Dirac :** Nous avons confirmé expérimentalement cette propriété fondamentale.

Ces résultats illustrent parfaitement l’utilité de la transformée de Fourier pour analyser la composition fréquentielle des signaux et comprendre les relations fondamentales entre les domaines temporel et fréquentiel.

## Applications pratiques

Les concepts étudiés dans ce TP ont de nombreuses applications pratiques, notamment :

- **En télécommunications :** La FFT est utilisée pour la modulation/démodulation dans les systèmes OFDM (Orthogonal Frequency Division Multiplexing) utilisés en Wi-Fi, 4G et 5G.
- **En traitement d’images et compression :** La FFT bidimensionnelle est utilisée dans les algorithmes de compression JPEG, permettant d’éliminer les composantes haute fréquence moins perceptibles.
- **En sécurité informatique :** Ces techniques sont exploitées en stéganographie (dissimulation d’informations dans les composantes fréquentielles d’une image ou d’un son) et en analyse forensique pour détecter des anomalies spectrales dans des signaux suspects.
- **En analyse vocale :** La FFT permet d’analyser le contenu fréquentiel de la voix pour la reconnaissance vocale et l’identification de locuteurs.

La compréhension des relations entre domaines temporel et fréquentiel est donc essentielle dans de nombreux domaines de l’ingénierie moderne.

# Conclusion Générale

Ce projet de traitement du signal numérique nous a permis d'explorer et d'approfondir les concepts fondamentaux à travers trois travaux pratiques complémentaires réalisés avec MATLAB.

La génération et visualisation des signaux élémentaires (TP N°2) nous a fourni une base solide pour comprendre les représentations temporelles des signaux, depuis les fonctions fondamentales comme l'impulsion de Dirac et l'échelon unitaire jusqu'aux signaux périodiques comme les sinusoïdaux, les signaux carrés et triangulaires.

L'analyse et la synthèse des filtres analogiques (TP N°3) nous a introduit aux techniques de conception suivant un gabarit spécifique. Notre filtre passe-bas de Butterworth d'ordre 5 a démontré l'efficacité de ces méthodes de filtrage, offrant une excellente sélectivité fréquentielle avec une atténuation supérieure à 40 dB dans la bande coupée. Cette partie a révélé l'importance des filtres dans le traitement du signal pour isoler efficacement certaines composantes fréquentielles.

L'étude de la transformée de Fourier rapide (TP N°4) nous a permis d'explorer la dualité fondamentale entre les domaines temporel et fréquentiel. Nous avons vérifié expérimentalement que la transformée de Fourier d'une fonction porte correspond à un sinus cardinal, et que l'élargissement d'un signal dans le domaine temporel entraîne un rétrécissement correspondant dans le domaine fréquentiel, illustrant parfaitement le principe d'incertitude temps-fréquence.

Ces compétences acquises en traitement du signal numérique sont particulièrement pertinentes pour notre formation en sécurité informatique. Les techniques d'analyse spectrale peuvent être appliquées à la stéganographie pour détecter des informations cachées dans des signaux. Le filtrage peut servir à l'extraction de caractéristiques dans l'analyse de signaux suspects ou dans le domaine de la cybersécurité des systèmes industriels, où les signaux physiques peuvent révéler des anomalies ou des attaques. De plus, la compréhension des principes de l'échantillonnage est essentielle pour développer des systèmes robustes de communication sécurisée.

En conclusion, ce projet nous a non seulement permis de maîtriser les outils fondamentaux du traitement du signal, mais nous a également préparés à les appliquer dans des contextes de sécurité informatique, où l'analyse et le traitement des signaux jouent un rôle crucial dans la détection et la prévention des menaces.