Introduction au signal et bruit
Travaux pratiques

Gabriel Dauphin

September 12, 2025

Contents

Chapter 1

Séance 1 de travaux pratiques

1.1 Préparation à faire avant la séance

Ceci constitue une partie théorique à faire avant la séance et à inclure dans le document `pdf` à rendre après la séance.

1. Choisissez un montage électronique simple pour lequel une source de tension ou d'intensité est considérée comme l'entrée d'un filtre et la tension aux bornes d'un composant ou l'intensité traversant un composant est considéré comme la sortie. Ce montage devra fonctionner ici en régime linéaire (par exemple l'amplificateur opérationnel devra être utilisé en rétro-action stable et non comme un comparateur).

Vous préciserez les valeurs numériques de chaque composant. Vous pouvez vous inspirer du montage présenté en section 1.1. Il est souhaitable que les caractéristiques du filtre soient numériquement simples à simuler, vous pouvez modifier les valeurs des composants pour que ce soit le cas.

2. En vous inspirant dé la section 1.1 et à partir d'une analyse physique du montage proposé, justifiez que le filtre proposé peut être modélisé avec une relation linéaire et temps invariante.

3. En vous inspirant dé la section 1.1, déterminez la réponse statique du filtre et le comportement limite à haute fréquence.

4. Calculez la réponse fréquentielle du filtre en utilisant vos connaissances d'électronique.

5. Déduisez une relation entrée-sortie décrite avec une équation différentielle.

6. Déduisez la réponse impulsionnelle théorique (il n'est pas nécessaire de conduire les calculs jusqu'au bout, il suffit qu'on puisse programmer le calcul de la réponse impulsionnelle).

Dans un premier temps, on considère dans ce TP ce montage et cette étude théorique.

Montage étudié

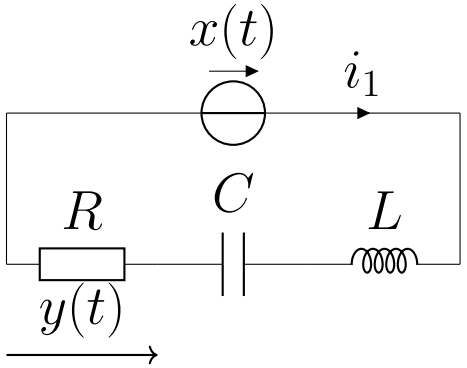

Figure 1.1: Montage correspondant au filtre étudié d'entrée la tension $x(t)$ et de sortie la tension $y(t)$. $R = 3\Omega$, $C = 0.5\text{F}$, $L = 1\text{H}$.

Analyse physique rapide du montage

Tous les composants sont linéaires entre l'intensité et la tension aussi la relation entre $x(t)$ et $y(t)$ est linéaire. On suppose ici que les composants ont des caractéristiques fixes, donc le système est temps invariant. À très basse fréquence, le condensateur se comporte comme un circuit ouvert et donc $\widehat{Y}(f)$ est nul à très basse fréquence. À très haute fréquence, la bobine se comporte comme un circuit ouvert et donc $\widehat{Y}(f)$ est nul à très haute fréquence.

Calcul théorique de la réponse fréquentielle

En utilisant les impédances de chaque composant et en observant qu'il y a un diviseur de tension, on a

$$\widehat{H}(f) = \frac{\widehat{Y}(f)}{\widehat{X}(f)} = \frac{R}{R + j2\pi fL + \frac{1}{j2\pi fC}} = \frac{j2\pi fRC}{1 + j2\pi fRC - 4\pi^2 f^2 LC} \tag{1.1}$$

On peut remarquer que les affirmation de la section 1.1 sont confirmées par l'équation (1.1)

$$\begin{cases} \lim\limits_{f\to 0} \widehat{H}(f) = \lim\limits_{f\to 0} \frac{j2\pi fRC}{1+j2\pi fRC-4\pi^2 f^2 LC} = 0 \\ \text{et } \lim\limits_{f\to +\infty} \widehat{H}(f) = \lim\limits_{f\to +\infty} \frac{R}{R+j2\pi fL+\frac{1}{j2\pi fC}} = 0 \end{cases} \tag{1.2}$$

Écriture de la relation entrée-sortie avec une équation différentielle

On remplace chaque terme $j2\pi f$ par $\frac{d}{dt}$

$$LC\frac{d^2}{dt^2}y(t) + RC\frac{d}{dt}y(t) + y(t) = RC\frac{d}{dt}x(t) \tag{1.3}$$

Calcul théorique de la réponse impulsionnelle : solution 1

Les calculs qui suivent peuvent dans une certaine mesure être obtenus en utilisant la toolbox [1] `sympy` à installer sur `Python` https://https://docs.sympy.org et en util-

[1] Remarquez que certaines commandes de `sympy` commencent par une majuscule.

isant Wolfram https://www.wolframalpha.com/
Wolfram `Fourier transform of exp(-p*t)*heaviside(t)` *Il reste à multiplier par* $\sqrt{2\pi}$ *et à remplacer* ω *par* $2\pi f$.

En posant $x(t) = \delta(t)$, on sait que $y(t)$ est la réponse impulsionnelle $h(t)$ qui vérifie donc

$$LC\frac{d^2}{dt^2}h(t) + RC\frac{d}{dt}h(t) + h(t) = RC\frac{d}{dt}\delta(t) \tag{1.4}$$

Cette équation peut se mettre sous une forme plus classique

$$LC\frac{d^2}{dt^2}\tilde{h}(t) + RC\frac{d}{dt}\tilde{h}(t) + \tilde{h}(t) = \delta(t) \text{ et } h(t) = RC\frac{d}{dt}\tilde{h}(t) \tag{1.5}$$

Le polynôme $Q(p) = LCp^2 + RCp + 1$ a deux racines réelles ($R^2C^2 - 4LC > 0$) : $p_1 = \frac{-RC-\sqrt{R^2C^2-4LC}}{2LC}$ et $p_2 = \frac{-RC+\sqrt{R^2C^2-4LC}}{2LC}$.

Python :

Le calcul symbolique permet de faire faire la résolution.

```
import sympy
R=sympy.Symbol('R')
L=sympy.Symbol('L')
```

```
C=sympy.Symbol('C')
p=sympy.Symbol('p')
p1,p2=sympy.solve('L*C*p**2+R*C*p+1','p'); print(f"p1={p1}, p2={p2}")
sympy.simplify(p1+p2)
sympy.simplify(p1*p2)
```

Les valeurs de $p1$ et 2 sont négatives et distinctes. Aussi $\tilde{h}(t)$ se met sous la forme

$$\tilde{h}(t) = (c_1 e^{p_1 t} + c_2 e^{p_2 t})[\![t \geq 0]\!] \tag{1.6}$$

Pour trouver c_1 et c_2, on pourrait calculer sa transformée de Fourier et identifier avec $\widehat{H}(f)$. On peut aussi calculer les dérivées successives pour vérifier avec l'équation (1.5).

$$\begin{aligned} RC\tfrac{d}{dt}\tilde{h}(t) &= RC(c_1 p_1 e^{p_1 t} + c_2 p_2 e^{p_2 t})[\![t \geq 0]\!] + RC(c_1 + c_2)\delta(t) \\ LC\tfrac{d^2}{dt^2}\tilde{h}(t) &= LC(c_1 p_1^2 e^{p_1 t} + c_2 p_2^2 e^{p_2 t})[\![t \geq 0]\!] + (c_1 + c_2)\delta'(t) + LC(c_1 p_1 + c_2 p_2)\delta(t) \end{aligned} \tag{1.7}$$

Ces dérivées successives peuvent ensuite être réintroduites dans l'équation (1.5), on a alors l'égalité quand ceci est vérifié.

$$\begin{cases} c_1(1 + RCp_1 + LCp_1^2) = 0 & \text{termes en facteur de } e^{p_1 t}[\![t \geq 0]\!] \\ c_2(1 + RCp_2 + LCp_2^2) = 0 & \text{termes en facteur de } e^{p_2 t}[\![t \geq 0]\!] \\ RC(c_1 + c_2) + LC(c_1 p_1 + c_2 p_2) = 1 & \text{termes en facteur de } \delta(t) \\ c_1 + c_2 = 0 & \text{termes en facteur de } \delta'(t) \end{cases} \Rightarrow \begin{cases} c_1 + c_2 = 0 \\ LC(c_1 p_1 + \end{cases}$$

La résolution de ce système donne

$$c_1 = \frac{1}{LC(p_1 - p_2)} \text{ et } c_2 = -\frac{1}{LC(p_1 - p_2)} \tag{1.9}$$

Et en utilisant $c_1 + c_2 = 0$, on a finalement

$$h(t) = RC\frac{d}{dt}\tilde{h}(t) = RC\frac{d}{dt}\tilde{h}(t) = RC(c_1 p_1 e^{p_1 t} + c_2 p_2 e^{p_2 t}) = \frac{R}{L}\frac{p_1}{p_1 - p_2}e^{p_1 t}[\![t \geq 0]\!] - \frac{R}{L}\frac{p_2}{p_1 -$$

Python :

```
from sympy.matrices import *
M=Matrix([[1,1],[p2,p1]])
B=Matrix([[R/L],[0]])
X=M.solve(B); c1=X[0]; c2=X[1]
print(f"c1={c1} c2={c2}")
#Pour vérifier
sympy.simplify(c1+c2)
sympy.simplify(p2*c1+p1*c2)
```

Calcul théorique de la réponse impulsionnelle : solution 2

Une seconde solution consiste à utiliser p_1 et p_2 calculé à partir de l'équation différentielle vérifiée par $\tilde{h}(t)$ et à supposer qu'ils sont valable pour $h(t)$ et que donc $h(t)$ serait

de cette forme-là. Et n'étant pas tout à fait sûr, on rajoute un possible Dirac.

$$h(t) = d_1 e^{p_1 t} [\![t \geq 0]\!] + d_2 e^{p_2 t} + d_3 \delta(t) \tag{1.11}$$

La transformée de Fourier de cette réponse impulsionnelle est

$$\widehat{H}(f) = d_1 \frac{1}{j2\pi f - p_1} + d_2 \frac{1}{j2\pi f - p_2} + d_3 \tag{1.12}$$

On identifie avec $\widehat{H}(f)$ pour trois fréquences quelconques, par exemple $f_0 = 0$, $f_1 = 1$ et $f_2 = 2$.

$$\begin{cases} d_1 \frac{1}{j2\pi f_0 - p_1} + d_2 \frac{1}{j2\pi f_0 - p_2} + d_3 = \widehat{H}(f_0) \\ d_1 \frac{1}{j2\pi f_1 - p_1} + d_2 \frac{1}{j2\pi f_1 - p_2} + d_3 = \widehat{H}(f_1) \\ d_1 \frac{1}{j2\pi f_2 - p_1} + d_2 \frac{1}{j2\pi f_2 - p_2} + d_3 = \widehat{H}(f_2) \end{cases} \tag{1.13}$$

Et on demande à l'ordinateur de résoudre ce système en utilisant des valeurs numériques de R, L, C.

1.2 Travail à effectuer pendant la séance

1.2.1 Préparation à l'utilisation de Python

Python :

Pour démarrer l'utilisation de Python, je propose de choisir deux répertoires, le premier que j'appelle `rep_prg` contient les programmes, notamment ceux disponibles sur mon site, et un autre répertoire que j'appelle `rep_tra` où vous mettez le travail effectué notamment les données et les figures.

Je propose ensuite d'effectuer les lignes suivantes qui utilisent un module `seb` que j'ai écrit pour ce cours et qui est disponible sur https://gabrieldauphin.neocities.org/L3SPI ou sur https://www-l2ti.univ-paris13.fr/~dauphin/L3SPI `plt` et `np` sont les modules `matplotlib` et `numpy` qui servent à tracer des graphes et à manipuler des vecteurs.

```
import sys
sys.path.append('rep_prg')
import os
os.chdir('rep_tra')
import seb
plt,np=seb.debut()
```

1.2.2 Visualisation de la réponse fréquentielle

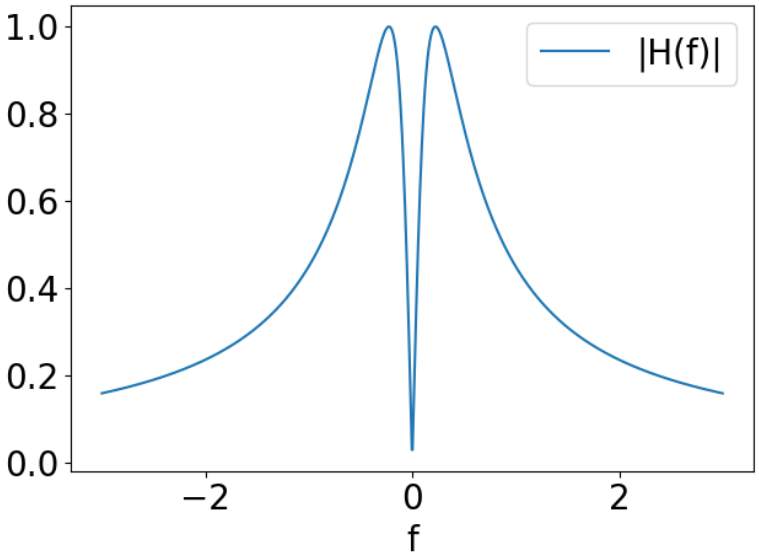

Figure 1.2: Module de la réponse fréquentielle

- Créez une échelle de fréquences avec le vecteur `f` par exemple entre -3 et 3 et représentez graphiquement la réponse fréquentielle

On appelle vecteur un tableau de valeurs composé d'une ligne (on parle alors de vecteur ligne) ou d'une seule colonne (on parle de vecteur colonne). La fonction `linspace` de `numpy` permet de générer un ensemble de valeurs en indiquant la première, la dernière et le nombre de ces valeurs. Ici ceci permet de construire l'échelle en fréquence.

```
f=np.linspace(-3,3,10**3)
```

Cette instruction génère 1000 valeurs entre -3 et 3. Cette notion est utile pour générer un graphique, elle permet d'obtenir le graphe 1.2 représentant le module de la réponse fréquentielle.

On définit les valeurs des variables. En Python il est possible d'allouer plusieurs variables en même temps.

```
R,C,L = 3,0.5,1
```

On obtient la réponse fréquentielle, (π, j et le carré sont implémentés avec `np.pi` et `1j` et `**2`).

```
H=1j*2*np.pi*f*R*C/(1+1j*2*np.pi*f*R*C-4*(np.pi**2)*(f**2)*L*C)
```

On obtient alors un vecteur ligne de même taille que `f` et contenant successivement toutes les valeurs complexes de H pour chacune des valeurs du vecteur `f`.

Pour faire le graphe, on commence par

```
fig,ax = plt.subplots()
```

Ensuite pour la courbe on rajoute [2] On implémente le module avec `np.abs`. Le troisième argument permet de rajouter une légende.

```
ax.plot(f,np.abs(H),label='|H(f)|')
```

[2]Et si on avait plusieurs courbes, il suffit de rajouter une ligne par courbe.

Sur le graphe on peut préciser ce que signifie l'axe des abscisses

```
ax.set_xlabel('f')
```

L'affichage de la légende est déclenchée par

```
ax.legend()
```

La gestion de la taille de la figure est faite avec

```
plt.tight_layout()
```

Je propose de sauvegarde la figure

```
fig.savefig('nom_figure.png')
```

La figure apparaît lorsqu'on exécute cette commande

```
fig.show()
```

Et comme on va utiliser à nouveau ces calculs il est intéressant de construire une fonction qui calcule $\widehat{H}(f)$. Notez qu'il est très important ici de laisser deux espaces au début de chaque ligne après la première ligne pour indiquer que les instructions font parti de la fonction appelée `H1`.

```
def H1(R,L,C,f):
  """fonction de la reponse frequentielle obtenue à partir du montage da
  import numpy as np
  H=1j*2*np.pi*f*R*C/(1+1j*2*np.pi*f*R*C-4*(np.pi**2)*(f**2)*L*C)
  return H
```

On peut faire appel à cette fonction avec

```
ax.plot(f,np.abs(H1(f)),label='|H(f)|')
```

1.2.3 Détermination numérique de la réponse impulsionnelle

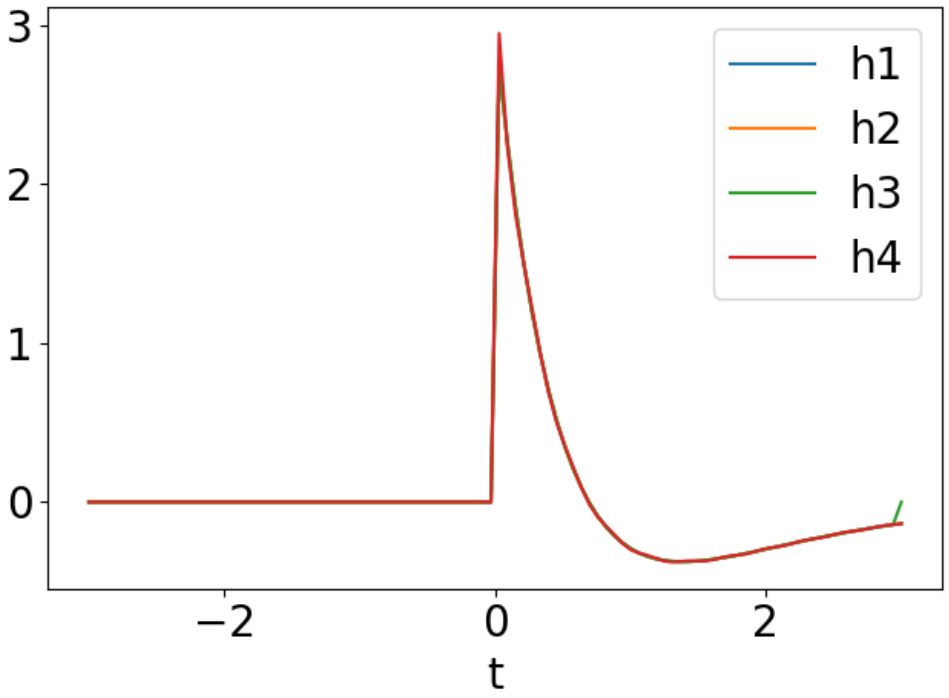

Figure 1.3: Réponse impulsionnelle $h(t)$.

- Créez une échelle de temps avec le vecteur `t` par exemple entre -3 et 3 et représentez graphiquement cette réponse impulsionnelle (voir section **??**, p. **??**) soit avec `h1` notée $h_1(t)$ soit avec `h2` notée $h_2(t)$.

- Déterminez $h_3(t)$ la réponse impulsionnelle à partir de l'équation différentielle (1.5) en s'inspirant de la section **??**.

- Déterminez $h_4(t)$ la réponse impulsionnelle cette fois-ci à partir de la réponse fréquentielle (1.1) en s'inspirant de la section **??**.

La figure 1.3 montre $h_1(t)$, $h_2(t)$, $h_3(t)$, $h_4(t)$. Ici l'implémentation est indiquée dans la section **??**.

Implémentation de la première solution

La partie théorique permet de calculer la réponse impulsionnelle liée à la première solution

Python :

On calcule d'abord p_1 et p_2.

```
p1=(-C*R - np.sqrt(C*(C*R**2 - 4*L)))/(2*C*L)
p2=(-C*R + np.sqrt(C*(C*R**2 - 4*L)))/(2*C*L)
```

On peut ensuite vérifier si les formules sont justes.

```
assert np.abs(L*C*p1**2+R*C*p1+1)<1e-8
assert np.abs(L*C*p2**2+R*C*p2+1)<1e-8
```

On calcule ensuite c_1 et c_2

```
c1,c2 = R/L*p1/(p1-p2), -R/L*p2/(p1-p2)
```

Puis on en déduit la réponse impulsionnelle.

```
y=c1*np.exp(t*p1)*(t>=0)+c2*np.exp(t*p2)*(t>=0)
y[t<0]=0
```

La dernière ligne a l'objectif de donner une valeur nulle lorsque $t < 0$ y compris quand les expressions de $y(t)$ pour $t \geq 0$ n'ont pas de sens.

Je propose de mettre toutes ces lignes de code dans une fonction notée `h1` dépendant de `R,L,C,t`.

Implémentation de la deuxième solution

La deuxième solution utilise le calcul déjà effectué de p_1 et p_2, que l'on peut retrouver avec

```
p1,p2=np.roots(np.array([L*C,R*C,1]))
```

Pour simplifier le code, je propose de définir une fonction notée `Hp` calculant $\frac{1}{j2\pi f - p}$

```
def Hp(f,p):
  return 1/(1j*2*np.pi*f-p)
```

Je considère trois fréquences

```
f=np.array([0,0.1,0.2])
```

La matrice contenant les paramètres à gauche de l'équation matricielle (1.13)

```
M = np.array([[Hp(f[0],p1),Hp(f[0],p2),1],[Hp(f[1],p1),Hp(f[1],p2),1],[H
```

Le vecteur colonne contenant les valeurs complexes souhaitées de la réponse fréquentielle sont les valeurs à droite de l'équation matricielle (1.13)

```
B = np.array([[H1(R,L,C,f[0])], [H1(R,L,C,f[1])], [H1(R,L,C,f[2])]])
```

L'ordinateur résout le système d'équations linéaires

```
d1,d2,d3=np.linalg.solve(M,B); d1,d2,d3=d1[0],d2[0],d3[0]
```

On vérifie qu'il était inutile de considérer un Dirac supplémentaire

```
assert np.abs(d3)<1e-10, (f"d3={d3:.2e}")
```

La réponse impulsionnelle trouvée est alors

```
y=d1*np.exp(p1*t)*(t>=0)+d2*np.exp(p2*t)*(t>=0)
y[t<0]=0
```

Je propose de la même façon de mettre ces lignes de code dans une deuxième fonction `h2`.

Détermination de $h(t)$ à partir de l'équation différentielle

On peut trouver $\tilde{h}$ en utilisant directement la fonction `sol_eq_diff` du module `seb` à partir de l'équation (1.4).

```
y1=seb.sol_eq_diff((L*C,R*C,1),t)
```

Notez que le premier argument est un `tuple` composé de trois arguments qui sont respectivement les coefficients devant $\frac{d^2}{dt^2}y(t)$, $\frac{d}{dt}y(t)$, $y(t)$.

On trouve alors $h(t)$ avec l'équation (1.5)

```
y2=R*C*seb.deriver(t,y1)
```

Je propose de mettre ces lignes de code dans une fonction `h3`.

Détermination de la réponse impulsionnelle à partir de la réponse fréquentielle

On utilise ici la fonction `H1` donnant la réponse fréquentielle du filtre étudié en utilisant (1.1). On évalue cette réponse fréquentielle en utilisant un très grand nombre de valeurs de fréquences. Puis on lui applique la transformée de Fourier inverse `TFI` définie dans `seb.py`.

```
f1=np.linspace(-20,20,10**5)
y1=seb.TFI(f1,H1(R,L,C,f1),t)
```

Comme on sait qu'avant $t = 0$, cette réponse impulsionnelle est nulle et que cette réponse impulsionnelle est réelle, on rajoute

```
y1[t<0]=0
y2=np.real(y1)
```

Je propose de mettre ces instructions dans une fonction `h4` et d'afficher sur un même graphique ces trois ou quatre fonctions.

1.2.4 Détermination de la réponse du système à $x_a(t) = \mathbb{T}(t)$

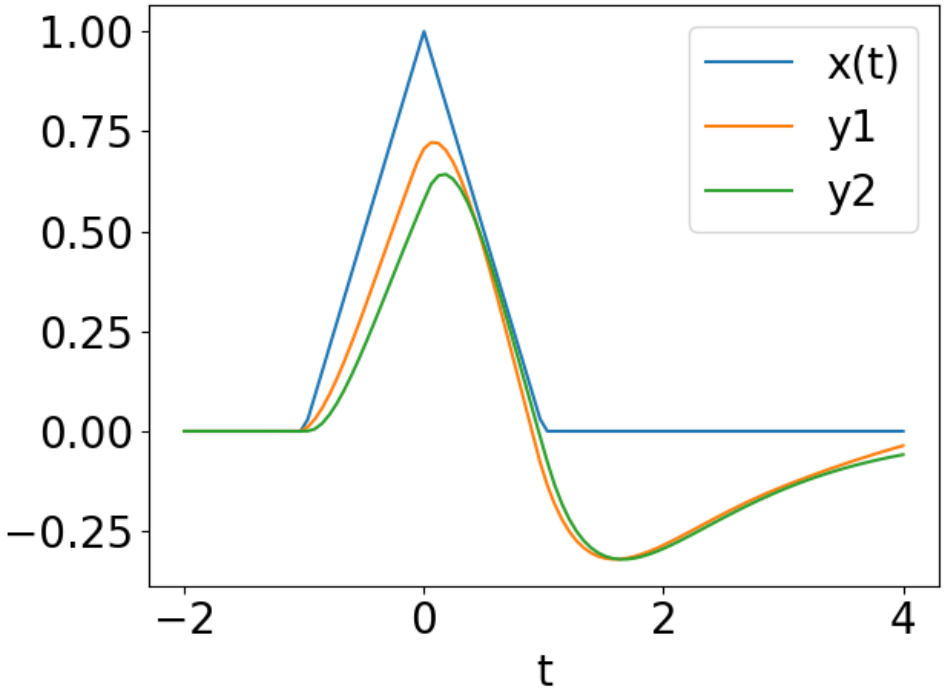

Figure 1.4: Courbe bleue en triangle : signal d'entrée. Les deux autres courbes sont le signal de sortie calculée de deux façons différentes et notéees `y1` et `y2`.

- Déterminez $y_{a1}(t)$ en utilisant la réponse impulsionnelle $h_1(t)$ ou $h_2(t)$ et la définition de $x_a(t)$ en utilisant la notion de produit de convolution.
- Calculez la réponse fréquentielle de la sortie $y_{a2}(t)$ en utilisant la réponse fréquentielle calculée en section 1.2.2 et la transformée Fourier de la fonction triangle pour en déduire $y_{a3}(t)$ avec

$$y_{a2}(t) = \mathrm{TF}^{-1}\left[\widehat{H}(f)\mathrm{TF}\left[\mathbb{T}(t)\right](f)\right](t) \tag{1.14}$$

  où $\widehat{H}(f)$ est la réponse fréquentielle du filtre considéré. Comme on sait que la sortie est réelle, il est intéressant de prendre la partie réelle de la sortie.

La figure 1.4 montre $y_{a1}(t)$, et y_{a2}.

Utilisation du produit de convolution

Une fonction simulant le produit de convolution entre deux signaux est disponible sur `seb.py`. Pour calculer le produit de convolution de deux signaux, elle requière ces deux signaux et leurs échelles de temps ainsi que l'échelle de temps du nouveau signal généré.

On créé l'échelle de temps du premier signal avec la fonction `arange` de `numpy` qui permet d'utiliser une période d'échantillonnage déjà fixée $T_e = \frac{1}{f_e}$. Ici `-1` est l'instant initial pour $x(t)$ et 1 est l'instant final pour $x(t)$.

```
tx = np.arange(-1,1,1/fe)
\begin{verbatim}
Pour choisir le début et la fin, il suffit de donner des valeurs qui per
de rallonger le signal sur des période où le signal est nul).
{\tt tx} est ensuite utilisée pour créer le signal d'entrée avec {\tt fo
\begin{verbatim}
x = seb.fonction_T(tx)
```

On créé une échelle de temps pour la réponse impulsionnelle. Il est important que cette deuxième échelle de temps ait la même période d'échantillonnage.

```
th = np.arange(0,4,1/fe)
```

De même que pour `tx`, le choix du début et de la fin de `th` est fait de façon à donner suffisamment d'informations. La valeurs de la réponse impulsionnelle sont récupérées avec `h1(R,L,C,th)` qui utilise la fonction `h1`.

On créé l'échelle de temps du signal à reconstruire

```
t=np.arange(-2,4,1/fe)
```

On obtient $y_{a1}(t)$ avec la fonction `convolution`

```
y1=seb.convolution(tx,x,th,h1(R,L,C,th),t)
```

Utilisation de la transformée de Fourier et de la transformée de Fourier inverse

Cette fois-ci on a besoin d'une échelle de temps pour calculer la réponse fréquentielle de $x(t)$. Celle-ci doit être suffisemment détaillée pour donner une bonne approximation à TF$[x(t)](f)$.

```
tx=np.linspace(-1,1,10**4)
```

On calcule $x(t)$ sur cette autre échelle de temps

```
x=seb.fonction_T(t)
```

On a besoin d'une échelle de fréquence assez détaillée pour en calculer la transformée de Fourier inverse.

```
f=np.linspace(-3,3,10**4)
```

On calcule TF$[x(t)](f)$ avec

```
X=seb.TF(t,x,f)
```

On obtient alors $\widehat{Y}(f)$ la transformée de Fourier de la sortie

```
Y=H1(f)*X
```

On choisit une échelle de temps pour `y2`, qui ne sert que pour l'affichage.

```
ty=np.linspace(-2,4,10**2)
```

On calcule la sortie en utilisant la transformée de Fourier inverse et en considérant la partie réelle.

```
y=np.real(seb.TFI(f,Y,ty)
```

1.3 Travail à rendre une semaine après la séance

En vous inspirant du travail effectué précédemment, répondez aux questions suivantes. Cette partie est à mettre après la partie correspondant à la section 1.1.

7. Créez une échelle de fréquences avec le vecteur `f` par exemple entre -3 et 3 et représentez graphiquement cette réponse fréquentielle.

8. Créez une échelle de temps avec le vecteur `t` par exemple entre -3 et 3 et représentez graphiquement cette réponse impulsionnelle soit avec `h1` notée $h_1(t)$ soit avec `h2` notée $h_2(t)$.

9. Déterminez $h_3(t)$ la réponse impulsionnelle à partir de la réponse fréquentielle (1.1)

10. Déterminez $h_4(t)$ la réponse impulsionnelle à

11. Choisissez un signal d'entrée noté $x(t)$.

12. Déterminez $y_1(t)$ la réponse du filtre à $x(t)$ en utilisant la réponse impulsionnelle $h_1(t)$ ou $h_2(t)$ et $x(t)$.

13. Calculez la réponse fréquentielle de la sortie $y_2(t)$ en utilisant la réponse fréquentielle et la simulation de la transformée de Fourier du signal d'entrée choisi.

$$y_2(t) = \mathrm{TF}^{-1}\left[\widehat{H}(f)\mathrm{TF}\left[\mathbb{T}(t)\right](f)\right](t) \tag{1.15}$$

Le document à rendre à un `pdf` dont la première page doit lister toutes les figures, toutes les formules et toutes les réponses aux questions. La suite du document expliquant la justification des réponses et/ou les programmes utilisés.

Chapter 2

Séance 2 de travaux pratiques

2.1 Préparation à faire avant la séance

1. Choisissez un signal ayant une représentation graphique simple. C'est ce signal noté $s_2(t)$ que vous étudierez à la place de $s_1(t)$ représenté sur la figure 2.1. Ce signal doit être non-nul seulement sur un intervalle petit (il n'est donc pas périodique).

2. Explicitez $s_2(t)$ au moyen des fonctions de base présentées dans le cours de façon similaire à l'équation (2.1).

3. Calculez l'intégrale de ce signal $\int_{-\infty}^{+\infty} s_2(t)\,dt$.

4. Si possible calculez la transformée de Fourier $\widehat{S}_2(f)$, ou donnez un algorithme permettant de le faire.

2.2 Travail à effectuer pendant la séance

2.2.1 Signal étudié

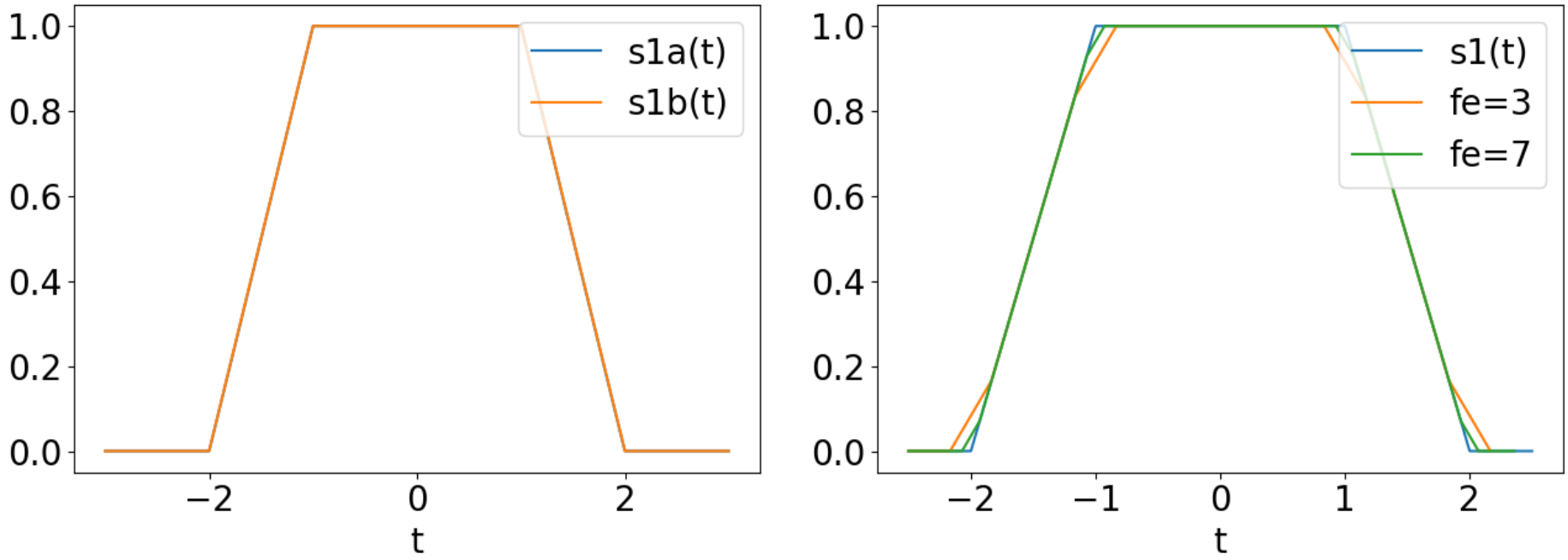

Figure 2.1: Signal étudié $s_1(t)$ et deux échantillonnages

Graphiquement on voit que le signal peut se mettre sous deux formes, la première

notée $s_{1a}(t)$ et la deuxième notée $s_{1b}(t)$.

$$s_1(t) = 2\mathbb{T}(t/2) - \mathbb{T}(t) = \mathbb{C}(t+1.5) + \Pi(t/2) + \mathbb{D}(t-1.5) \tag{2.1}$$

La gauche de la figure 2.1 montre le signal étudié $s_1(t)$ avec ces deux formes. La droite montre l'échantillonnage avec deux fréquences d'échantillonnages.

2.2.2 Calculs théoriques sur le signal $s_1(t)$

Graphiquement on peut voir la surface du signal qui est composé de deux triangles et d'un rectangle, il a donc une surface de $2(1\times1/2) + (1\times2) = 3$.

$$\int_{-\infty}^{+\infty} s_1(t)\,dt = 3 \tag{2.2}$$

Je propose d'utiliser l'expression de $s_1(t)$ en fonction de $\mathbb{T}(t)$ dont la transformée de Fourier vaut $\text{sinc}^2(f)$:

$$\widehat{S}_1(f) = 2\text{TF}\left[\mathbb{T}(t/2)\right](f) - \text{TF}\left[\mathbb{T}(t)\right](f) = 4\text{TF}\left[\mathbb{T}(t)\right](2f) - \text{TF}\left[\mathbb{T}(t)\right](f) = 4\,\text{sinc}^2(2f) -$$

On remarque que ces deux équations sont cohérentes : $3 = \widehat{S}_1(0) = 4 - 1$

2.2.3 Échantillonnage du signal

La droite de la figure 2.1 montre le signal étudié $s_1(t)$ et deux échantillonnages faits à deux fréquences différentes, ici $fe_a = 3\text{Hz}$ et $fe_b = 7\text{Hz}$.

Python :

L'équation (2.1) permet de définir une fonction

```
def s1a(t):
  return seb.fonction_T(t/2)*2-seb.fonction_T(t)
```

Pour simuler le ignal **s1** en tant que signal temps continu, on créé une échelle de temps très précise.

```
t=np.linspace(-2,2,10**4)
s1=s1a(t)
```

La fréquence d'échantillonnage élevée permettra de faire des calculs un peu plus précis. Pour simuler ce même signal en tant que signal temps discret avec une fréquence d'échantillonnage de $f_e = 3$Hz, on utilise la fonction **np.arange** qui permet de préciser exactement la fréquence d'échantillonnage choisie.

```
fe1=3
t=np.arange(-2,2,1/fe1)
s1=s1a(t)
```

L'implémentation de la droite de la figure 2.1 est réalisée dans la section **??**.

2.2.4 Simulation de la transformée de Fourier

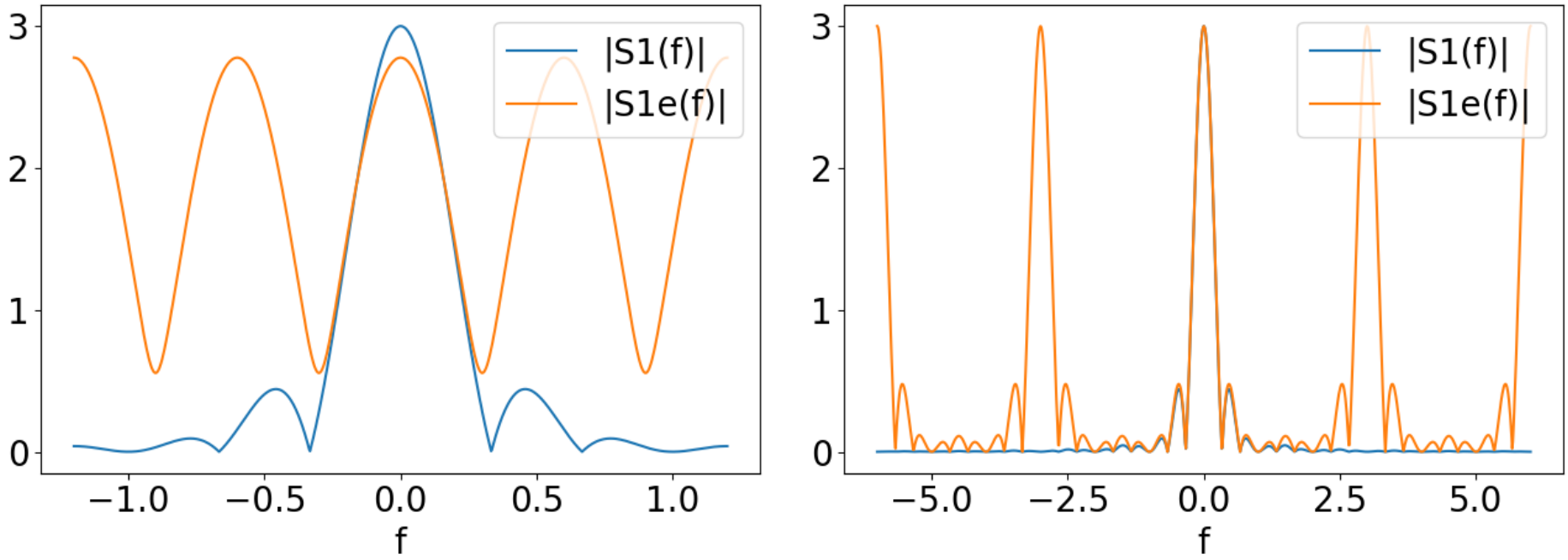

Figure 2.2: Spectre de $s_1(t)$ et à gauche de $s_1(t)$ échantillonné à 0.6Hz et à droite à 3Hz.

La figure 2.2 montre à gauche et à droite deux spectres périodique, ce sont les transformées de Fourier des signaux échantillonnés. C'est le même spectre en bleu et non-périodique qui est représenté à gauche et à droite.

Python :

La fonction `seb.TFTD` implémente la transformée de Fourier à temps discret, son utilisation est similare à `seb.TF`, on précise une échelle de temps et les valeurs du signal

ainsi qu'une échelle en fréquence souhaitée. Pour bien montrer la périodicité de période f_e, je considère une échelle de fréquence entre $-3f_e$ et $3f_e$.

```
fe1=3
t1=np.arange(-2,2,1/fe1)
f=np.linspace(-3*fe1,3*fe1,300)
S1e_a=seb.TFTD(ta,s1a(t1),f)
```

L'implémentation est réalisée dans la section **??**.

2.2.5 Simulation de la périodisation du spectre

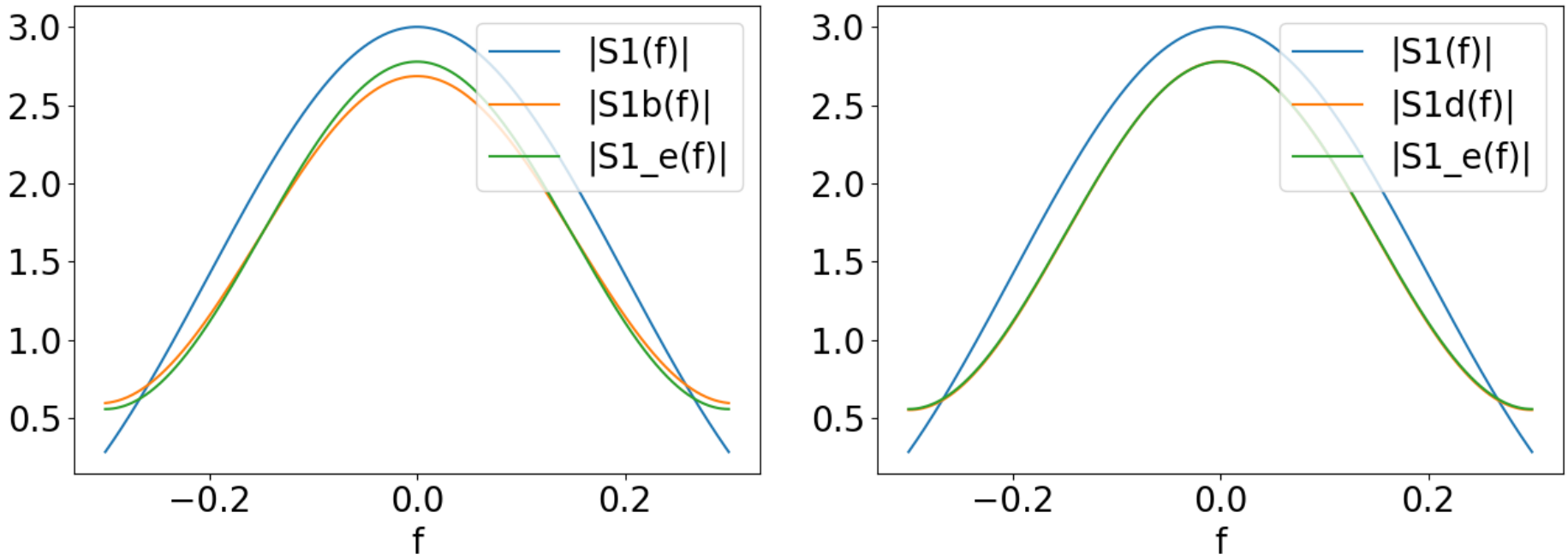

Figure 2.3: Spectre de $s_1(t)$ et à gauche de $s_1(t)$ échantillonné à 0.6Hz et à droite à 3Hz.

- Représentez $\left|\widehat{S}_1(f)\right|, \left|\widehat{S}_1(f) + \widehat{S}_1(f+f_e) + \widehat{S}_1(f-f_e)\right|, \left|\widehat{S}_1(f) + \widehat{S}_1(f+f_e) + \widehat{S}_1(f-f\right.$ et $\frac{1}{f_e}\left|\widehat{S}e_1(f)\right|$ avec une représentation centrée. Vous pouvez utiliser une version approchée ou exacte de $\widehat{S}_1(f)$.

La figure 2.3 montre dans les deux figures en haut $S_1(f)$ et avec une forme plus en cloche $\widehat{S}_1^{\#}(f) = \frac{1}{f_e}\text{TFTD}\left[s_1[n]\right](f)$. À gauche, $\widehat{S}_1^{\#}(f)$ est approchée avec $\left|\widehat{S}_1(f) + \widehat{S}_1(f+f_e) + \widehat{S}_1(f\right.$ on voit la différence entre les deux courbes. Cette différence avec $\widehat{S}_1^{\#}(f)$ disparaît à droite en visualisant $\left|\widehat{S}_1(f) + \widehat{S}_1(f+f_e) + \widehat{S}_1(f-f_e) + \widehat{S}_1(f+2f_e) + \widehat{S}_1(f-2f_e)\right|$. L'implémenta est réalisée dans la section **??**.

2.2.6 Périodisation de $s_1(t)$

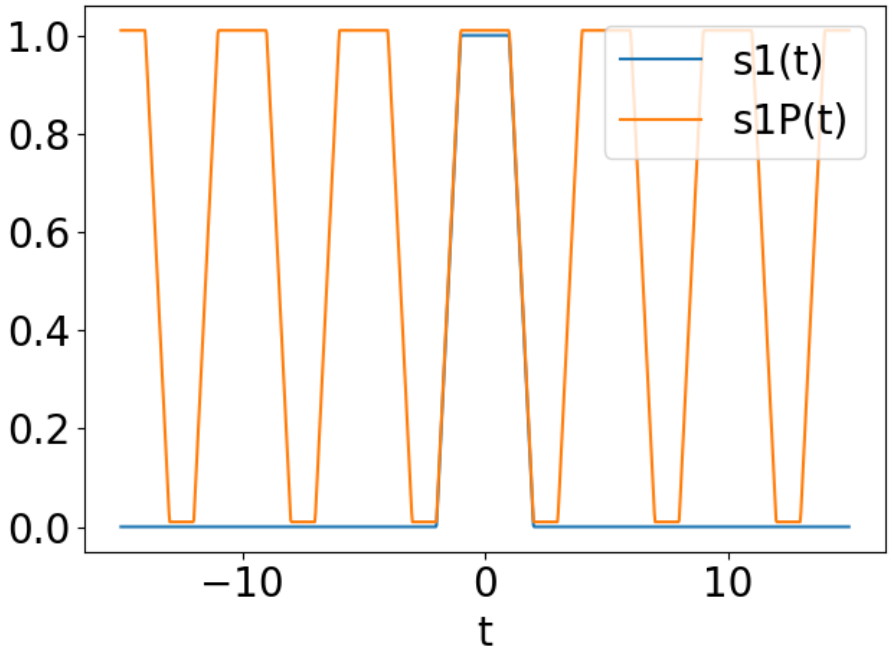

Figure 2.4: Signal $s_1(t)$ en bleu périodisé en $s_1^P(t)$ en répétant le motif défini sur $[-\frac{5}{2}, \frac{5}{2}]$.

La figure 2.4 montre $s_1(t)$ et $s_1^P(t)$ périodisés. La courbe de $s_1^P(t)$ est légèrement surélevée pour montrer la différence avec $s_1(t)$. L'implémentation est dans **??**.

2.2.7 Simulation de la transformée de Fourier

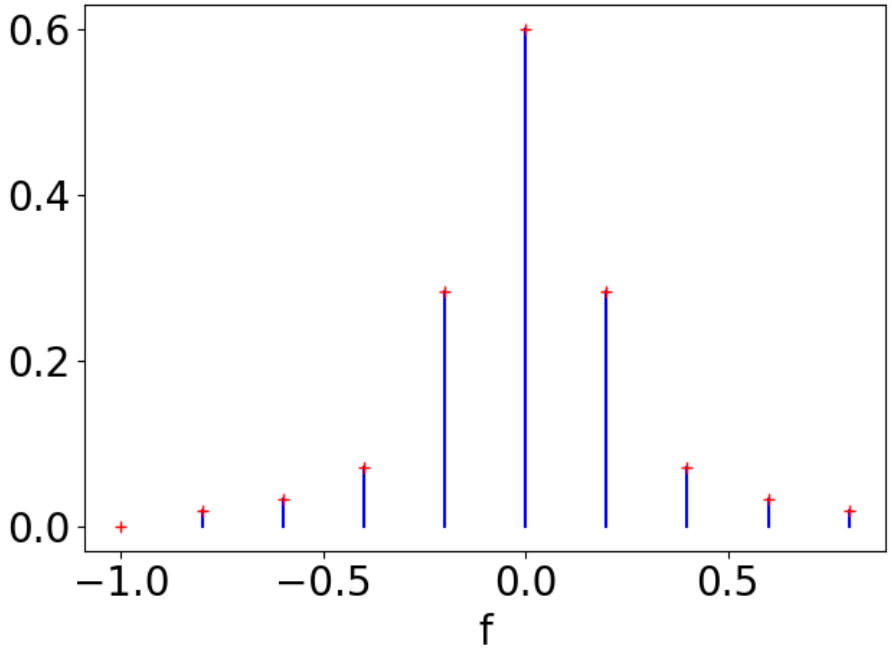

Figure 2.5: Signal $s_1(t)$ en bleu périodisé en $s_1^P(t)$ en répétant le motif $[-\frac{5}{2}, \frac{5}{2}]$.

La figure 2.5 montre des raies représentant les coefficients de la série de Fourier de $s_1^P(t)$. Comme la restriction de $s_1^P(t)$ sur $[-\frac{5}{2}, \frac{5}{2}]$ coïncide avec $s_1(t)$, ces raies peuvent être obtenues avec $\widehat{S}_1(f)$ pour $f_k = \frac{k}{T}$, ces points sont les plus indiqués en rouge.

2.2.8 Simulation de la périodisation de la transformée de Fourier

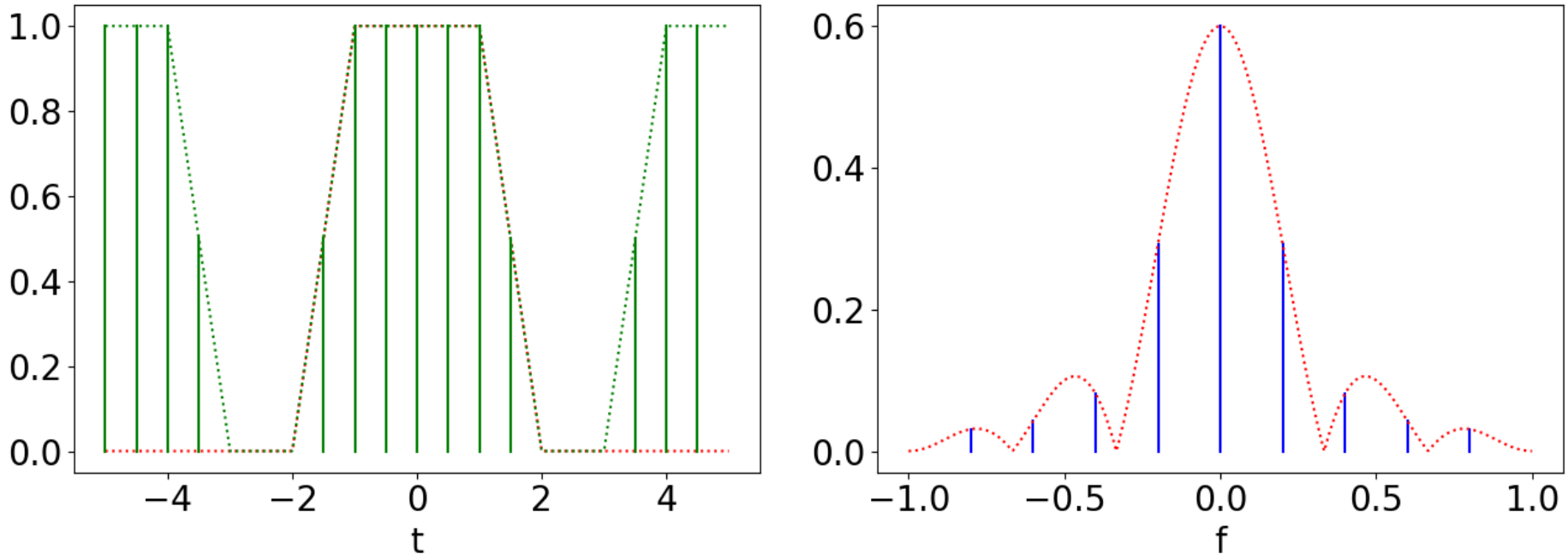

Figure 2.6

- Choisissez une fréquence d'échantillonnage fe_c tel que la période T soit un multiple de $\frac{1}{fe_c}$. On note $N = Tfe_c$.
- On note $s_1^{\#P}[n]$ le signal $s_1^P(t)$ échantillonné à fe_c. Représentez $s_1^{\#P}[n], s_1^{\#}[n], s_1^P(t)$ sur le même graphique.
- Représentez sur la même figure $\text{TFD}[s_1^{\#P}[n]][k]$ et $\frac{1}{N}\text{TFTD}[s_1^{\#P}[n]]\left(k\frac{fe_c}{N}\right)$

La figure 2.6 est implémentée dans **??**

2.3 Travail à rendre une semaine après la séance

5. Choisissez deux fréquence d'échantillonnage fe_a et fe_b, montrez que les transformées de Fourier des signaux échantillonnés ne sont pas les mêmes que celle du signal non-échantillonné et qu'elles sont périodiques de période f_e.

Chapter 3

Séance 3 de travaux pratiques

3.1 Préparation à faire avant la séance

1. Choisissez la distribution de probabilité du bruit blanc considéré.
2. Choisissez un filtre défini par une équation différentielle, on note ici $\mathcal{H}$ sa relation entrée sortie.
3. Calculez la réponse impulsionnelle $h(t)$, (ou donnez un algorithme).
4. Calculez la réponse fréquentielle $\widehat{H}(f)$, (ou donnez un algorithme).

5. Choisissez une fréquence d'échantillonnage f_e pour simuler le bruit considéré.

6. Choisissez deux instants d'observation t_0 et t_1.

3.2 Travail à effectuer pendant la séance

Dans un premier temps, on considère dans ce TP, la loi de probabilité gaussienne, centrée et d'écart-type 1, le filtre défini par cette équation différentielle.

$$\frac{d}{dt}y(t) + y(t) = x(t-1) \tag{3.1}$$

Calcul de la réponse impulsionnelle

Je pose $\tilde{h}(t)$ la solution de l'équation différentielle

$$\frac{d}{dt}y(t) + y(t) = \delta(t) \tag{3.2}$$

Le polynôme associé est $p+1$, il a une racine $p=-1$, donc les solutions sont de type

$$\tilde{h}(t) = Ae^{-t}[\![t \geq 0]\!](t) \tag{3.3}$$

Pour trouver A, je remplace $\tilde{h}(t)$ dans l'équation (3.2) et je trouve

$$\delta(t) = \frac{d}{dt}\tilde{h}(t) + \tilde{h}(t) = \left(-Ae^{-t}[\![t \geq 0]\!](t) + A\delta(t)\right) + Ae^{-t}[\![t \geq 0]\!](t) = A\delta(t) \tag{3.4}$$

J'en déduis que $A = 1$ et $\tilde{h}(t) = e^{-t}[\![t \geq 0]\!](t)$

La réponse impulsionnelle de $\mathcal{H}$ est obtenue en mettant en entrée $x(t) = \delta(t)$ qui devient $x(t-1) = \delta(t-1)$ qui modifie la relation entrée-sortie de (3.2) en retardant l'entrée de $t = 1$ et par suite en retardant la sortie de $t = 1$. Donc $h(t) = \tilde{h}(t-1)$.

$$h(t) = e^{-(t-1)}[\![t \geq 1]\!](t) \tag{3.5}$$

Calcul de la réponse fréquentielle

À partir de l'équation différentielle (3.1) et en remarquant qu'un retard devient un déphasage proportionnel à la fréquence, on a

$$\widehat{H}(f) = \frac{e^{-j2\pi f}}{1 + j2\pi f} \tag{3.6}$$

Estimation de la densité spectrale en sortie pour un bruit blanc gaussien

Je note $S_x(f), S_y(f)$ les densités spectrales de puissance de $\overset{r}{X}(t)$ et $\overset{r}{Y}(t)$. Comme $\overset{r}{X}(t)$ est un bruit blanc centré et d'écart-type 1, $S_x(f) = S_x(0) = 1$.

Du coup $S_y(f) = |\widehat{H}(f)|^2 = \frac{1}{1+4\pi^2 f^2}$

Estimation de la moyenne et de la variance en sortie pour un bruit blanc gaussien

Le gain statique $\widehat{H}(0) = 1$ d'après l'équation (3.6). Aussi la moyenne statistique de la sortie est la même que la moyenne statistique de l'entrée qui vaut 0.

$$\mathrm{E}\left[\overset{r}{Y}(t)\right] = \widehat{H}(0)\mathrm{E}\left[\overset{r}{X}(t)\right] = 0 \tag{3.7}$$

Je note $S_x(f), S_y(f)$ les densités spectrales de puissance de $\overset{r}{X}(t)$ et $\overset{r}{Y}(t)$. Comme $\overset{r}{X}(t)$ est un bruit blanc centré et d'écart-type 1, $S_x(f) = S_x(0) = 1$.

$$\mathrm{Var}\left[\overset{r}{Y}(t)\right] = \gamma_x(0) = \int_{-\infty}^{+\infty} S_y(f)\, df = \int_{-\infty}^{+\infty} \frac{1}{1 + 4\pi^2 f^2\, df} \tag{3.8}$$

Au moyen de deux changements de variable $u = 2\pi f$ et $\theta = \arctan(u)$, on calcule l'intégrale

$$\int_{-\infty}^{+\infty} \frac{df}{1 + 4\pi^2 f^2} = \frac{1}{2\pi}\int_{-\infty}^{+\infty} \frac{du}{1 + u^2} = \frac{1}{2\pi}\int_{-\frac{\pi}{2}}^{\frac{\pi}{2}} d\theta = \frac{1}{2} \tag{3.9}$$

Finalement on obtient

$$\mathrm{Var}\left[\overset{r}{Y}(t)\right] = \frac{1}{2} \tag{3.10}$$

Estimation de l'autocorrélation en sortie pour un bruit blanc gaussien

Je note $\gamma_x(t), \gamma_y(t)$ les densités spectrales de puissance de $\overset{r}{X}(t)$ et $\overset{r}{Y}(t)$. Comme $\overset{r}{X}(t)$ est un bruit blanc centré et d'écart-type 1, $\gamma_x(t) = \delta(t)$.

Et on a aussi

$$\gamma_y(t) = \mathrm{TF}^{-1}\left[S_y(f)\right](t) \tag{3.11}$$

Pour trouver cette transformée de Fourier inverse, on suppose qu'elle s'écrit sous la forme $\gamma_y(t) = be^{-a|t|}$ La valeur absolue fait qu'on découpe en deux l'intégrale à effectuer.

$$S_y(f) = b\int_{-\infty}^{0} b^{at}e^{-j2\pi ft}\,dt + b\int_{0}^{+\infty} b^{-at}e^{-j2\pi ft}\,dt \tag{3.12}$$

La deuxième intégrale vaut

$$\int_0^{+\infty} e^{-at}e^{-j2\pi ft}\,dt = \left[\frac{e^{-t(a+j2\pi f)}}{a+j2\pi f}\right]_0^{+\infty} = \frac{1}{a+j2\pi f} \tag{3.13}$$

Après un changement de variable $t' = -t$, la première intégrale vaut

$$\int_{-\infty}^{0} e^{at}e^{-j2\pi ft}\,dt = \int_0^{+\infty} e^{-at}e^{j2\pi ft}\,dt = \int_0^{+\infty} e^{-at}e^{-j2\pi(-f)t}\,dt = \frac{1}{a+j2\pi(-f)} \tag{3.14}$$

Du coup avec les deux termes on trouve que

$$S_y(f) = b\left(\frac{1}{a+j2\pi f} + \frac{1}{a-j2\pi f}\right) = \frac{2ab}{a^2+4\pi^2f^2} \tag{3.15}$$

Par identification avec la vraie valeur de $S_y(f)$, on trouve que $a = 1$ et $b = 0.5$.

$$\gamma_y(t) = 0.5e^{-|t|} \tag{3.16}$$

3.2.1 Représentation de la densité de probabilité du signal en sortie en $t = t_0$ pour une fréquence d'échantillonnnage f_e