Travaux pratiques de traitement numérique du signal

Caroline Kulcsár et Gabriel Dauphin

November 18, 2025

Données nécessaires à la réalisation des TP

Ces données peuvent être téléchargées sur

`http://www-l2ti.univ-paris13.fr/~dauphin/PRG.zip`

Une version électronique de ce document est disponible sur

`http://www-l2ti.univ-paris13.fr/~dauphin/tp_tns_te2.pdf`

SÉANCE 1

1 Signal sinusoïdal et échantillonnage, DSP, signal carré

1.1 Visualisation de données successives

Entrez les commandes suivantes :

```
t=TP1_2(0.1,2,0.5001,0.5);
start(t)
:
:
stop(t)
delete(t)
```

Ces commandes permettent de visualiser ce que l'on pourrait avoir comme signal en sortie d'un capteur correspondant à un signal sinusoïdal.

1. Que valent la période d'échantillonnage, la fréquence d'échantillonnage, la période de la sinusoïde et la fréquence de la sinusoïde ?

<u>Solution</u> :$T_e = 0.5$, $f_e = 2$, $T = 10$, $f_0 = 0.1$.

1.2 Visualisation de données au cours du temps

Entrez les commandes suivantes :

```
t=TP1_2(0.05,1,20,0.5);
start(t)
:
:
stop(t)
delete(t)
```

2. Que valent la période d'échantillonnage, la fréquence d'échantillonnage, la période de la sinusoïde et la fréquence de la sinusoïde ?

<u>Solution</u> :$T_e = 1$, $f_e = 1$, $T = 20$, $f_0 = 0.05$.

Syntaxe du programme utilisé

La syntaxe du programme est :

```
  t=TP1_2(f0,fe,duree,period);
  start(t);
  :
  :
  stop(t);
delete(t);
```

où les paramètres et fonctions utilisées signifient :

- `f0` : fréquence de la sinusoïde en Hertz.
- `fe` : fréquence d'échantillonnage en Hertz.
- `duree` : le signal est visualise entre l'instant actuel et `duree` secondes avant l'instant actuel.
- `period` : nombre de secondes entre chaque nouvelle evaluation, il est necessaire que `period` > 0.2 pour éviter qu'il y ait un risque que la dernière évaluation ne soit terminée avant que ne commence l'évaluation suivante.
- `start(t)` : permet de déclencher le démarrage.
- `stop(t)` : permet d'arrêter le démarrage.
- `delete(t)` : permet de supprimer l'objet t. Ceci est ABSOLUMENT nécessaire avant d'en créer un autre sous le même nom.
- `clear` ne permet pas de supprimer l'objet, elle ne supprime que l'appellation.

1.3 Échantillonnage d'un signal sinusoïdal, DSP

On considère le signal sinusoïdal causal à temps continu suivant :

$$x(t) = a\sin(\omega_0 t + \varphi)$$

où $\omega_0 = 2\pi f_0$ est la pulsation. Pour obtenir un signal temps discret, on échantillonne $x(t)$ avec une fréquence d'échantillonnage f_e.
Valeurs numériques : $f_0 = 50$ Hz, $a = 2$, $\varphi = \pi/2$.

3. Tracez x_n (utiliser `stem`) sur une durée de 100 ms pour les fréquences d'échantillonnage suivantes :

$$f_e = 75\text{Hz},\ \ f_e = 150\text{Hz},\ \ f_e = 1\text{kHz}.$$

Précisez pour chaque valeur de f_e si la condition de Shannon-Nyquist est vérifiée et commentez à partir de l'allure des figures.

Solution :

```
a=2; f0=50; fe=75; tn=0:1/fe:(100e-3-1/fe); xn=a*sin(2*pi*f0*tn+pi/2);
figure(1); stem(tn,xn);
```

4. On échantillonne $x(t)$ à la fréquence $f_e = 500$ Hz sur l'intervalle $[0,\ 40\text{ ms}]$. Le buffer utilisé n'a que 8 bits. Le signal est quantifié sur l'intervalle $[-2,\ 2]$.
   - Que vaut N_q, le nombre de valeurs quantifiées possibles ?
     Solution : $N_q = 2^8 = 256$.
   - Quelle est la valeur Q du pas de quantification ?
     Solution : $Q = 2 * a/N_q = \frac{4}{256} = 0.0156$

– Tracer en fonction du temps le signal quantifié x_n^q et le signal non quantifié x_n (utiliser `floor`), pour cette question et pour rendre l'erreur de quantification plus visible, on considérera un buffer de 3 bits. **Solution :**

```
a=2; f0=50; fe=500; tn=0:1/fe:(40e-3-1/fe); xn=a*sin(2*pi*f0*tn+pi/2);
b=3;
Q=2*a/(2^b);
xq=(floor(xn/Q)+0.5)*Q;
figure(1); plot(tn,xn,tn,xq);
```

– On appelle ici l'erreur de quantification, le signal formé de la valeur absolue de la différence entre le signal quantifié et le signal non-quantifié. Il s'agit ici d'une quantité qui varie au cours du temps. Calculez la moyenne de cette erreur de quantification. Que dire de l'erreur de quantification moyenne ?
Solution :

```
a=2; f0=50; fe=500; tn=0:1/fe:(40e-3-1/fe); xn=a*sin(2*pi*f0*tn+pi/2);
b=8;
Q=2*a/(2^b);
xq=(floor(xn/Q)+0.5)*Q;
err=mean(abs(xn-xq)), %0.0022
```

– Cette erreur moyenne de quantification varie en fonction du nombre de bits. Tracez la courbe de l'évolution de cette erreur en fonction du nombre de bits utilisées pour la quantification, utilisez une échelle logarithmique pour l'axe des ordonnées. Commentez la courbe obtenue.
Solution : Voir figure 1.

```
 a=2; f0=50; fe=500; tn=0:1/fe:(40e-3-1/fe); xn=a*sin(2*pi*f0*tn+pi/2);
b_l=1:12;
Q_l=2*a./(2.^b_l);
err_l=zeros(size(b_l));
for Q_=1:length(Q_l)
  Q=Q_l(Q_);
  xq=(floor(xn/Q)+0.5)*Q;
  err_l(Q_)=mean(abs(xn-xq));
end
figure(1); semilogy(b_l,err_l);
```

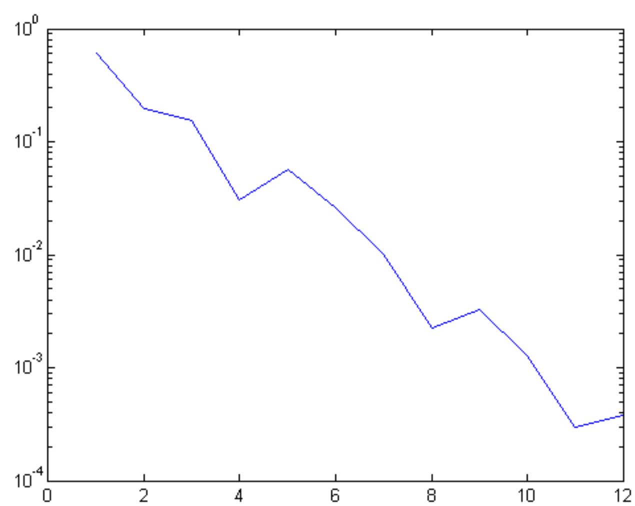


Figure 1: corrige de l'erreur moyenne en fonction du nombre de bits en échelle logarithmique

5. Exprimez la densité spectrale de puissance (ou DSP ou spectre), notée $S_x(f_k)$, en fonction de la transformée de Fourier discrète (TFD) $\hat{X}(f_k)$ de x_n. Donnez l'échelle en fréquence associée à la DSP (la valeur des fréquences f_k pour lesquelles la DSP $S_x(f_k)$ est définie).

On échantillonne $x(t)$ à la fréquence $f_e = 100.2$Hz sur [0, 3 s]. Le signal échantillonné est noté x_n.

6. Visualiser x_n en fonction du temps. Commenter.
7. La condition de Shannon-Nyquist est-elle vérifiée ?
8. Visualiser la DSP de x_n (utiliser `fft`) en fonction de la fréquence. Retrouve-t-on le pic de puissance associé à la fréquence f_0 du signal $x(t)$?

1.4 Visualisation de données au cours du temps lorsque la fréquence d'échantillonnage est faible par rapport à la fréquence de la sinusoïde

Entrez les commandes suivantes :

```
t=TP1_2(0.57,2,200,0.5);
start(t)
:
:
stop(t)
delete(t)
```

1.5 Signal carré et DSP

On s'intéresse à un signal carré $x(t)$ de période $T_0 = 20$ ms défini par

$$\begin{cases} x(t) = 1 & \text{si } t \in [0, T_0/2[\ , \\ x(t) = -1 & \text{si } t \in [T_0/2, T_0[\ , . \end{cases}$$

Ce signal est échantillonné à la fréquence f_e pour donner le signal x_n. On pourra utiliser la fonction Matlab `square`.

9. On prend $f_e = 500$ Hz, et on échantillonne sur $[0, 40\text{ms}]$. Tracez le signal x_n (utiliser `square`).
10. Visualisez la DSP de x_n en fréquence centrée (pour cela on utilisera `fft` et `fftshift`).
11. Quelle est la valeur de la DSP correspondant à la fréquence nulle ? Expliquer.
12. Quelle est la fréquence du second terme de la DSP en fonction de f_e ?
13. Quelle est la fréquence du dernier terme de la DSP en fonction de f_e ?
14. Expliquez la différence entre les DSP des signaux sinusoïdal et carré.

1.6 Signal et bruit blanc

On considère le signal sinusoïdal x_n défini à la question I.3. Ce signal est mesuré par un appareil induisant un bruit de mesure additif gaussien B_n de moyenne nulle et de variance $\sigma^2 = 0,3$. La mesure est définie par l'équation

$$y_n = x_n + b_n.$$

15. Tracer la DSP de y_n. Expliquer le résultat.
16. Que faudrait-il faire comme opération sur les fréquences (donc sur les coefficients de la transformée de Fourier) pour obtenir un signal moins bruité que y_n ?

SEANCE 2

2 Fonction de corrélation, détection radar

1. Une estimée de la fonction de corrélation de deux signaux x_n et y_n discrets périodiques est donnée par

$$C_{xy}(\ell) = \frac{1}{N} \sum_{k=0}^{N-1} x_k y_{k-\ell}.$$

   On note $C_b(\cdot)$, $C_x(\cdot)$ et $C_y(\cdot)$ les fonctions d'autocorrélation des signaux b_n, x_n et y_n, en fonction de l'écart (ℓ) entre les échantillons. En utilisant la fonction `xcorr`, calculer les fonctions de corrélation C_b, C_x et C_y pour les écarts (lag en anglais) compris entre $-L$ et $+L$. Prendre par exemple $L = 40$.

2. Visualiser C_b, C_x et C_y pour σ =0,5, $\sigma = 2$, et $\sigma = 5$ en les affichant superposées, avec les valeurs d'écart en abscisse. Comparer les figures. Où se situe la principale différence ? Comment l'expliquer ?

3. On va quand même utiliser cette fonction `xcorr` pour détecter la présence d'un motif dans un signal reçu.

   On reprend pour cela le principe de fonctionnement simplifié du radar pour la détection de cibles. Le radar envoie *via* des ondes un motif de signal ; ce signal est réfléchi par les cibles et le radar le reçoit en retour, noyé bien évidemment dans du bruit. Connaissant le motif envoyé, on peut, par corrélation, détecter la présence de ce motif dans le signal bruité reçu (et donc connaître la distance de la cible : on connaît en effet la vitesse de propagation des ondes et le motif détecté a parcouru deux fois la distance radar–cible).

   Afin de simuler la détection par corrélation, il faut d'abord choisir un motif à envoyer, et étudier la forme de la corrélation obtenue. On propose de tester l'autocorrélation des signaux suivants :

   – bruit blanc centré, par exemple `m=randn(1,50)`

   – sinusoïde pure, par exemple `m=sin(2*pi*(1:50)*0.1)`

   – sinusoïde modulée par une gaussienne, par exemple
   `m=sin(2*pi*(1:50)*0.1).*exp(-(-24:25).^2/10^2)`
   Choisir un motif m facilement détectable par corrélation que vous utiliserez pour la suite (justifiez ce choix).

4. Simuler le signal reçu noté r, pour un faible nombre de cibles :

   – créer un signal reçu (non bruité) de longueur 300, avec 3 cibles, certaines cibles pouvant être très proches

   – rajouter un bruit blanc gaussien centré de variance à choisir.

5. Calculer l'intercorrélation entre le motif et le signal reçu, et l'afficher pour les décalages positifs uniquement. Qu'observe-t-on ?

6. Recommencer le calcul avec différents niveaux de bruit pour le signal reçu. Que constate-t-on ? Comment pourrait-on faire pour avoir une détection automatique ?

7. Écrivez le lien entre la densité spectrale d'énergie d'un signal y_n et sa fonction d'autocorrélation, lorsque ce signal est temps discret et non-périodique et nul en dehors des instants $n = 0, \ldots, n = N - 1$.

8. Démontrez la question 7.

9. Écrivez le lien entre la densité spectrale de puissance d'un signal x_n et sa fonction d'autocorrélation, lorsque ce signal est temps discret et périodique de période N.

10. Démontrez la question 9.

11. Dans cette expérimentation on se place dans le cadre où le vrai signal est périodique de période N. Dans Matlab, la fonction de corrélation `xcorr` est définie par

$$C(x,y,\ell) = \sum_{k=0}^{N-1} x(k)y(k-\ell), \text{ avec } x(k) = y(k) = 0 \text{ si } k < 0 \text{ ou } k \geq N,$$
$$\texttt{xcorr}(x,y,L) = [C(x,y,-L)\dots C(x,y,0)\dots C(x,y,L)]$$

pour x et y vecteurs de taille N. En dehors de l'intervalle $[0,\dots,N-1]$, les échantillons sont pris égaux à 0. Pourquoi cette fonction `xcorr` ne peut être utilisée pour vérifier numériquement l'équation donnée à la question 9 ?

SEANCE 3

3 Sur-échantillonnage d'un signal discret

Le sur-échantillonnage consiste, à partir d'échantillons s_n obtenus par échantillonnage d'un signal $s(t)$ à la période T_e, à calculer des échantillons intermédiaires espacés de moins de T_e. Si l'on calcule par exemple un échantillon supplémentaire entre chaque échantillon disponible, on se retrouve donc avec des valeurs espacées de $T_e/2$.

On considère le signal s_n correspondant à l'échantillonnage sur [0, 10 s[à la fréquence f_e d'un signal à temps continu $s(t)$. On suppose que le vrai signal $s(t)$ est périodique de période $10s$. Le signal s_n est contenu dans le fichier `signalbase.mat`. Ce fichier est contenu dans un fichier appelé `PRG.zip`.

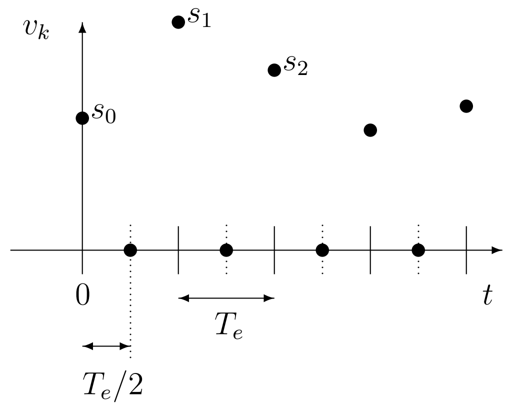
1. On a pris $f_e = 25$ Hz et l'on a obtenu 250 échantillons de s_n. Représenter la DSP de s_n en fréquences centrées en Hertz (échelle semi-logarithmique).

2. Calculer par la formule d'interpolation de Shannon (que vous tronquerez en justifiant votre approximation) la valeur du signal s à l'instant $t = 1{,}3$ s à partir des valeurs de s_n. Montrez que cela vaut approximativement

$$s(1,3) = 3,52$$

3. Quelle est la fréquence maximale observable dans s_n lorsque $f_e = 25$ Hz ?

On appelle v_n le signal obtenu en intercalant un zéro (échantillon nul) entre chaque échantillon s_n (voir figure).



4. Quelle est la fréquence maximale générée dans v_n par l'intercalage d'un zéro (échantillon nul) entre chaque échantillon s_n ?

5. Représenter la DSP de v_n en fréquences centrées en Hertz. Expliquer le résultat.

6. Pour réaliser un sur-échantillonnage de s_n, on va filtrer v_n par un filtre passe-bas. Quel est l'intérêt de ce type de filtrage pour le sur-échantillonnage ? Quelle doit-être la fréquence de coupure ? Quelle sera la fréquence d'échantillonnage correspondant au signal ainsi filtré ?

On propose le filtre H_1 défini par l'équation de filtrage suivante :

$$y_n = v_n + v_{n-1}.$$

7. $\boxed{\text{Que vaut } y_\ell \text{ en fonction de } s_n \text{ ? La relation entre } s_n \text{ et } y_n \text{ est-elle un filtre ?}}$

8. Visualisez la réponse en fréquence du filtre. Est-ce un filtre passe-bas, passe-bande, coupe-bande, passe-haut ? Définissez la fréquence de coupure et donnez sa valeur numérique.

9. Filtrer v_n et visualiser y_n. Comparer y_n avec s_n en affichant les courbes superposées.

On propose maintenant le filtre H_2 défini par

$$z_n = \frac{1}{2}(v_n + 2v_{n-1} + v_{n-2}).$$

10. $\boxed{\text{Que vaut } z_n \text{ en fonction de } s_n \text{ ?}}$

11. Visualiser la réponse en fréquence du filtre et la comparer avec celle de H_1.

12. Filtrer v_n et visualiser z_n. Le signal ainsi filtré contient-il plus d'information que v_n ? Comparer z_n avec s_n en affichant les courbes superposées, conclure.

13. Si l'on voulait réaliser un filtre idéal pour couper les fréquences indésirables dans v_n, quelle serait la réponse en fréquence d'un tel filtre (donnez le gabarit et précisez la fréquence de coupure) ?

14. Comment réaliser ce filtre idéal en utilisant la TFD ? Quels sont les inconvénients majeurs de ce filtre ?

SEANCE 4

4 Filtrage d'un signal bruité

4.1 Analyse de la relation entrée-sortie d'un filtre

Entrez les commandes suivantes :

```
t=TP4();
start(t)
:
:
stop(t)
delete(t)
```

Le graphe du haut permet de visualiser l'entrée du filtre. Il s'agit d'une sinusoïde bruitée par un bruit blanc gaussien additif.

1. Quelle est la fréquence d'échantillonnage et la fréquence de la sinusoïde ? **Solution :**$f_e = 2$, $f_0 = 0.1$.

Le graphe du milieu permet de visualiser la sortie du filtre, qui est approximativement une sinusoïde.

2. Quelle est la fréquence d'échantillonnage et la fréquence de la sinusoïde ?

   Solution :$f_e = 2$, $f_0 = 0.1$.

3. Quelle est approximativement l'amplitude de la sinusoïde du graphe du haut et quelle est celle de la sinusoïde du graphe du milieu ?

   Solution :$a_1 = 1$, $a_2 = 0.5$.

Le graphe du bas permet de visualiser la réponse fréquentielle du filtre.

4. Le filtre est-il un passe-bas, un passe-haut ? **Solution :**Passe-bas

5. Que vaut la fréquence de coupure du filtre ? Elle se lit au bas du trait fin vertical du graphique du bas et il est possible de faire un zoom. **Solution :**$f_c = 0.033$

6. Comment peut-on lire sur la réponse fréquentielle, le rapport entre l'amplitude de la sinusoïde du graphe du haut et l'amplitude de la sinusoïde du graphe du milieu ? **Solution :**C'est la réponse fréquencielle en la fréquence f_0, c'est-à-dire ici 0.35.

4.2 Analyse d'un filtre

Le filtre qui transforme l'entrée visualisée par le graphe du haut en la sortie visualisée par le graphe du milieu est ainsi définie

$$H(z) = \frac{1-\rho}{1-\rho z^{-1}} \text{ et } y_n - \rho y_{n-1} = (1-\rho)x_n$$

$H(z)$ est la fonction de transfert et la deuxième équation est la relation entrée-sortie du filtre. ρ, appelée `rho`, est un paramètre entre 0 et 1. Sa réponse fréquentielle en $f = 0$ est 1, quelle que soit la valeur de ρ, en effet :

$$\widehat{H}(0) = H(1) = \frac{1-\rho}{1-\rho\times 1} = 1$$

Il est possible de modifier la valeur de ρ pendant la simulation, c'est-à-dire après `start(t)` et avant `stop(t)` avec la commande suivante qui fixe $\rho = 0.85$:

```
t=TP4_modifier_rho(t,0.85);
```

7. Ajustez ρ de façon que la sinusoïde en sortie ait à peu près à 80% de l'amplitude de la sinusoïde en entrée et que l'impact du bruit sur le signal de sortie soit assez faible. **Solution** **:**Par exemple $\rho = 0.65$.

8. Ajustez ρ de façon que le signal en sortie soit très similaire au signal en entrée. **Solution** **:**Par exemple $\rho = 0.1$.

On cherche à filtrer le bruit présent sur un signal. Le signal proposé est celui correspondant à `signalbase.mat`, signal dit «utile» et noté x ($x = x_n$, $k = 0, \ldots, N-1$) auquel est ajouté un bruit blanc b gaussien centré d'écart-type σ. Le signal bruité obtenu est noté y ($y = y_n$, $k = 0, \ldots, N-1$).

9. Soit un signal x noyé dans un bruit b. Définir le rapport signal-à-bruit (RSB) en fonction de la puissance des signaux x et b, et définir la puissance d'un signal. Comment obtient-on un RSB en dB ?

10. Générer le signal bruité y à partir de x, en ajoutant le bruit blanc b avec successivement plusieurs valeurs de σ : 1/10, 1/2, 1 et 2. Pour chaque valeur, donner le rapport signal-à-bruit. Que remarque-t-on ?

11. Afficher les DSP du signal utile x d'une part, et du bruit b de la question précédente d'autre part. En comparant les hauteurs des DSP en haute fréquence, justifier l'emploi d'un filtre passe-bas pour atténuer le bruit.

On propose de filtrer le signal y par un filtre moyenneur H causal, défini par la réponse impulsionnelle :

$$\begin{aligned} h_n &= 1/P, \text{ pour } 0 \leq n \leq P-1, \\ h_n &= 0 \text{ sinon.} \end{aligned} \tag{1}$$

12. Exprimer la sortie z_n du filtrage de y par H, et montrer que ce filtre est un filtre RIF.

13. Visualiser la réponse en fréquence de ce filtre et donner ses principales caractéristiques.

14. Représenter la sortie de ce filtrage. Choisir une valeur de P qui vous paraît convenir. Que conclure ?

On se propose maintenant de filtrer le signal par un filtre défini par

$$\begin{aligned} &a_1 z_n = a_2 z_{n-1} + y_n, \ n \geq 0 \\ &z_n = 0 \text{ si } n < 0. \end{aligned} \tag{2}$$

15. Montrer que ce filtre est un filtre RII.

16. Calculer les coefficients h_n de la réponse impulsionnelle théorique de ce filtre.

17. En prenant $a_1 = 1$ et a_2 au choix, vérifier numériquement le résultat de la question 16 sous Matlab à l'aide de la fonction `impz`.

18. Quelle condition faut-il sur a_1 et a_2 pour assurer la stabilité du filtre ?

19. A quoi cela servirait-il d'imposer la condition

$$\sum_{n\in\mathbb{Z}} h_n = 1 \quad ?$$

20. Choisir deux coefficients a_1 et a_2 pour assurer à la fois la stabilité du filtre et la condition ci-dessus.
21. Visualisez la sortie de ce filtre attaqué par y (y obtenu avec $\sigma = 0,5$). Réglez les coefficients pour avoir selon vous un «bon» filtrage du bruit.
22. Visualiser la réponse en fréquence de ce filtre avec les valeurs de a_1 et a_2 retenues et comparer ses principales caractéristiques avec le filtre moyenneur.
23. Comparer les deux techniques de filtrage de bruit proposées et conclure.

SEANCE 5

5 Synthèse de filtres

5.1 Illustration en temps réel

Entrez les commandes suivantes :

```
t=TP5();
start(t)
:
:
stop(t)
delete(t)
```

Le graphe du haut est la visualisation d'un signal sinusoïdal bruité par un bruit blanc gaussien additif.

1. Quelle est la fréquence d'échantillonnage et la fréquence de la sinusoïde ? **Solution :**$f_e = 2$, $f_0 = 0.1$.

Le graphe du bas est la visualisation en module, en échelle logarithmique et en fréquence centrée de la transformée de Fourier discrète calculée à partir des échantillons du signal d'entrée sur une durée de Δt.

2. Quelle est la durée Δt ? **Solution :**$\Delta t = 20$. C'est $\frac{1}{f_e}$ fois le nombre de points sur le graphique du bas ou c'est l'inverse de la distance en Hertz en deux points successifs.
3. Retrouvez sur le graphique du bas la fréquence de la sinusoïde présente dans le signal visualisée dans le graphique du haut ? **Solution :**Ce sont les deux pics associés à 0.1Hz et -0.1Hz.

Il est possible de modifier Δt qui est notée `delta_t` avec

```
t=TP5_modifier_delta_t(t,8);
```

Cette commande est à placer avant `stop(t);` et après `start(t);` Entrez cette commande.

4. Comment expliquez-vous que le point central ait un mouvement assez régulier ? **Solution :**Le point central est associé à $\widehat{X}(0)$ et vaut la moyenne du signal entre l'instant présent et l'instant présent moins Δt. Quand Δt est plus faible que $T/2$, cette moyenne évolue avec le temps.

Entrez la commande suivante :

```
t=TP5_modifier_delta_t(t,5);
```

5. Comment expliquez-vous que l'on ne retrouve plus les deux pics ? **Solution :**La faible durée sur laquelle le signal est analysée ne permet plus d'avoir une résolution fréquentielle suffisamment faible pour permettre de retrouver le pic à 0.1Hz.

En général, le filtrage des signaux passe par la détermination des caractéristiques souhaités ou la définition d'un gabarit, ces caractéristiques ou ce gabarit dépendent du traitement que l'on désire effectuer. Ainsi, les principaux gabarits rencontrés en pratique correspondent aux filtrages passe-bas, passe-bande, coupe-bande et passe-haut. On va tester différents types de filtrage sur le signal bruité $y = x + b$ où b est un bruit blanc gaussien centré de variance $\sigma^2 = 0,5$ et x est le signal échantillonné donné par `signalbase.mat` pour lequel on suppose maintenant une **fréquence d'échantillonnage égale à 1 kHz**.

5.2 Filtre passe-bas

Un filtre passe-bas idéal a une réponse fréquentielle qui est définie sur l'intervalle $[-f_e/2, f_e/2]$ par $\mathbf{1}_{[-f_c/2, f_c/2]}(f)$. Une façon d'implémenter une approximation de ce filtrage pour un signal défini sur une certaine durée consiste à considérer que le signal est périodique de cette durée à appliquer la transformée de Fourier de ce signal sur cette durée à multiplier la transformée obtenue par la réponse fréquentielle et enfin à appliquer la transformée de Fourier inverse.

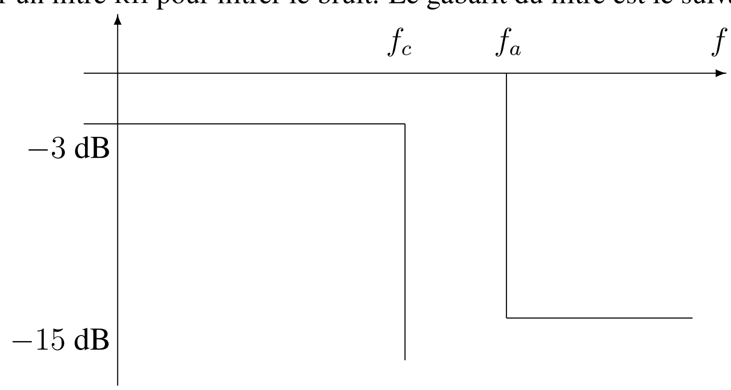
6. Choisissez une fréquence de coupure adéquate et réaliser ce filtrage, puis commenter les résultats en faisant varier f_c.

7. Pour approcher le filtre idéal, on utilise un filtre RIF dont les coefficients correspondent à des fenêtres de pondération. Choisir une fenêtre qui vous paraît adéquate et synthétiser le filtre en utilisant `fir1`.

8. Affichez la fonction de transfert du filtre choisi (à l'aide de `freqz`) et commenter le graphique.

9. Appliquez un tel filtre au signal et examinez l'effet de la variation de la taille de la fenêtre sur les résultats.

10. On décide maintenant d'utiliser un filtre RII pour filtrer le bruit. Le gabarit du filtre est le suivant :



    Quelles valeurs donner à f_c et f_a (respectivement fréquence de coupure et d'atténuation) ?

11. Calculez numériquement l'ordre optimal en utilisant la fonction `buttord`. Dans l'aide en ligne, R_p désigne le gain en dB de la partie passante du filtre, R_s désigne le gain en dB de la partie non-passante. f_c (fréquence de coupure) désigne un bord de l'intervalle sur lequel le module de la réponse fréquentielle a un gain supérieur à R_p ; f_a (fréquence d'atténuation) désigne une extrémité de l'intervalle sur lequel le module de la réponse fréquentielle a un gain inférieur à R_s.

12. Calculez avec `butter` ce filtre et appliquez le filtre au signal y_n. Comparez les résultats obtenus en changeant l'ordre du filtre.

La méthode vue en cours permet de synthétiser un filtre numérique de Butterworth un passe-bas de fréquence de coupure $f_c^{\#} = f_e/3$ à l'ordre 2 et conduit à

$$H^{\#}(z) = \frac{3 + 6z^{-1} + 3z^{-2}}{(4 + \sqrt{6}) + 4z^{-1} + (4 - \sqrt{6})z^{-2}}$$

13. Montrez que le filtre obtenu est strictement le même que celui que l'on aurait obtenu avec `butter` à l'ordre 2.

    <u>Solution</u> :

```
[B_mat,A_mat]=butter(2,2/3);
B_cal=[3 6 3];
A_cal=[4+sqrt(6) 4 4-sqrt(6)];
abs(B_cal/A_cal(1)-B)<=1e-14,
abs(A_cal/A_cal(1)-A)<=1e-14,
```

5.3 Filtre coupe-bande

Le signal utile x est maintenant perturbé par un signal sinusoïdal de fréquence 50 Hz et d'amplitude 1,2, dû à un résidu de courant électrique par exemple. On peut, dans ce cas là, utiliser un filtre coupe-bande autour de 50 Hz.

14. On considère le signal

$$y_n = x_n + b_n$$

    où b_n est un bruit blanc centré de variance 0,5, et x_n le signal `signalbase.mat` perturbé par la sinusoïde. Générer y_n et visualiser le signal.

15. Visualiser la DSP de y_n. Que remarque-t-on ?

16. Réaliser un filtre RIF coupe-bande de fréquence caractéristique 50 Hz et de largeur de bande 10 Hz. Vérifier le gabarit à l'aide de `freqz`. Combien faut-il de coefficients ? Commenter.

17. Réaliser maintenant un filtre RII coupe-bande de fréquence caractéristique 50 Hz et de largeur de bande 10 Hz. Vérifier le gabarit à l'aide de `freqz`. Combien faut-il de coefficients ? Commenter.

18. Filtrer y par un filtre passe-bas puis par l'un des deux filtres ci-dessus de façon à supprimer le bruit et la sinusoïde indésirable. Justifier l'utilisation du filtre passe-bas avant le filtre coupe-bande.

19. Visualiser les résultats de filtrage pour différentes valeurs de largeur de bande et commenter les résultats obtenus.