Sorbonne Université

Faculté des Sciences et Ingénierie

Outils Mathématiques pour le Traitement des Signaux

Textes des TP de l'UE LU2EE201

Guido Valerio

Génie Électrique et Électronique de Paris (GeePs)
Sorbonne Université

Mai 2020

Table des matières

Usage de Matlab sur un ordinateur personnel

Vous pouvez préparer les TP avant chaque séance sur un ordinateur personnel, grâce à la licence Matlab de Sorbonne Université.

Pour utiliser cette licence, vous devez créer un compte sur le site Mathworks en utilisant uniquement l'adresse email de l'université (noms de domaine : `@etu.upmc.fr` ou `@etu.sorbonne-universite.fr`). Une autre adresse e-mail ne marchera pas.

Le site pour la création du compte est https://fr.mathworks.com/mwaccount/register

Installation de Matlab sur votre ordinateur

Vous pouvez télécharger Matlab à partir de votre compte MathWorks (mais c'est volumineux, donc long à télécharger selon la connexion que vous avez). L'identifiant/mot de passe permet d'activer la licence après l'installation.

Utilisation de Matlab online

Pour Matlab online (https://fr.mathworks.com/products/matlab-online.html), il faut accéder avec votre identifiants MathWorks et vous trouverez un site web qui fonctionne comme un Matlab installé sur machine. Vous pouvez faire un upload des fichiers avec lesquels travailler. Enregistrez souvent les modifications, puisque après un certain temps de non activité la session se déconnecte !

Prise en main de Matlab

La première séance TP est dédiée à une prise en main de Matlab pour la définition et la visualisation des signaux temps-continu.

Une préparation aux commandes de base de Matlab est conseillée avant la séance, en utilisant la licence Matlab Sorbonne sur son propre ordinateur. Nombreux outils sont disponible. Notamment, le cours en ligne :
https://fr.mathworks.com/learn/tutorials/matlab-onramp.html (chapitres 1-10) permet de maîtriser les commandes à utiliser dans les TP de cette UE.

Instructions sur la préparation des comptes rendus des TP

Ces quatre séances de TP sont pensées comme une aide pour la compréhension des sujets des cours magistraux à travers la programmation et la visualisation de simples opérations de traitement des signaux.

Les TP2, TP3 et TP4 sont notés sur le comportement/participation à la séance et sur un compte rendu préparé par chaque binôme *pendant* chaque séance et rendu *avant la fin* de chaque séance.

Le binôme doit répondre aux questions du sujet *dans* le fichier Matlab, en utilisant des lignes de commentaires (les commentaires commencent avec le symbole % et ne sont pas simulés par Matlab).

Le fichier final doit contenir toutes les réponses (graphes inclus). Le fichier avec les graphes et les output créés doit être sauvegardé en format pdf avec la commande `publish`.

Par exemple, le fichier `TP2.m` complété par le binôme doit être sauvegardé en pdf avec la commande `publish('TP2.m','pdf')`. Cette commande doit être écrite dans la fenêtre de commande (et non dans le fichier `TP2.m`!).

Un fichier pdf `TP2.pdf` est alors créé dans le dossier `html`. Le binôme doit le renommer en ajoutant les prénoms et noms des deux étudiants. Le pdf ainsi-créé est le compte-rendu qui sera évalué par l'encadrant de la séance. Essayez de créer un compte-rendu déjà la première séance pour être prêts lorsque le compte-rendu sera obligatoire et noté.

A noter que s'il y a des erreurs dans le code, la commande `publish` ne marche pas. Dans ce cas, si vous n'arrivez pas à identifier/corriger les erreurs, commentez les lignes qui posent des problèmes et lancez la commande `publish`.

1 Définition des signaux et calcul de leurs propriétés

Ouvrez le fichier `TP1.m` et modifiez le code pour répondre aux questions suivantes. N'hésitez pas à chercher des exemples sur l'aide de Matlab ou à faire des recherches en ligne pour avoir plus d'informations sur les règles de syntaxe de Matlab.

1.1 Visualiser le graphe d'un signal en Matlab

Visualiser les deux signaux $f(t) = 2t$ et $g(t) = t^2$ dans l'intervalle $t \in [-5, 5]$.

L'intervalle $[-5, 5]$ est décrit comme un array de $n + 1$ éléments réels : le premier vaut -5, le dernier $+5$, et la distance entre deux éléments adjacents est de $10/n$:

$$-5, -5 + \frac{10}{n}, -5 + 2\frac{10}{n}, \cdots, 5 \quad (1)$$

Les fonctions sont calculées sur chaque élément de l'array :

$$f(-5), f\left(-5 + \frac{10}{n}\right), f\left(-5 + 2\frac{10}{n}\right), \cdots, f(5) \quad (2)$$

La visualisation d'une courbe se fait avec la commande `plot` : `plot(t,f)` visualise le graphe de la fonction $f(t)$ calculée dans l'intervalle t. Avant d'ouvrir un nouveau graphe, choisir son numéro (par exemple `1`) et donner la commande `figure(1)`.

1.2 Définir un nouveau signal en Matlab

1. Définir et visualiser les signaux suivants. Important : le nombre π est `pi`, l'unité imaginaire j est `1i`, la partie réelle, imaginaire, le module et a phase d'un complexe z sont `real(z)`,`imag(z)`,`abs(z)`,`angle(z)`.
   (a) $\cos(t)$ et $\sin(t)$. L'argument t doit-il être écrit en radiants ou en dégrée ?
   (b) sinc(t). Matlab a une fonction `sinc(t)`, mais elle ne correspond pas exactement au $\mathrm{sinc}(t)$ défini en cours. Quelle est la différence ?
   (c) exponentielle imaginaire e^{jt} (visualiser sa partie réelle, sa partie imaginaire, son module et sa phase en fonction de t)
2. Vérifier l'égalité suivante :

$$\cos(t + \alpha) = \cos(t)\cos(\alpha) - \sin(t)\sin(\alpha) \quad (3)$$

   en visualisant le terme de gauche et le terme de droite sur le même graphe. La commande `hold on` garde le graphe précédent et dessine le nouveau sur l'ancien.
3. Dessiner une fonction porte $\Pi_2(t)$ à l'aide de la fonction contenue dans le fichier `porte.m`
4. Dessiner dans l'intervalle $[-5, 5]$ une Gaussienne fenêtrée dans l'intervalle $[-1, 1]$.

1.3 Calcul de valeurs moyennes

Calculer la valeur moyenne des fonctions suivantes :

1. un sinus $\sin(t)$ en une période,
2. un sinus carré $\sin^2(t)$ en une période
3. l'exponentielle imaginaire e^{it} en $[0, \pi]$

Calculer ces valeurs à l'aide de la fonction `integral(fun,tmin,tmax)` qui calcule l'intégrale $\int_{\text{tmin}}^{\text{tmax}} \text{fun}(t)\mathrm{d}t$. La fonction doit être définie avant la ligne `integral` comme un “handle” de fonction :

```
fun = @(t) sin(t).^2
```

pour le cas du sinus carré.

Vérifier les résultats Matlab en les comparant avec les calculs analytiques.

1.4 Calcul d'énergies et puissances

1. Calculer l'énergie des signaux suivants :
   (a) un sinus $\sin(t)$ en une période,
   (b) un exponentielle monolaterale $e^{-at}u(t)$ avec $a = 5$ réel, en $[0, +\infty)$
   (c) une Gaussienne $e^{-\frac{t^2}{2}}$ sur l'axe réel entier,
   (d) l'exponentielle imaginaire e^{it} en $[0, \pi]$.

   Comparer les résultats Matlab avec les résultats analytiques.
2. Calculer les puissances des signaux suivants :
   (a) une constante égale à 10 en un intervalle arbitraire,
   (b) un signal périodique créneau à définir avec la fonction `porte` dans le fichier `porte.m` L'amplitude du signal est $A = 2$ et son rapport cyclique est $r = 1/3$.

   Comparer les résultats Matlab avec les résultats analytiques.

1.5 Calcul d'une série numérique

Calculer la somme de la série géométrique

$$\sum_{i=1}^{+\infty} r^i = \frac{r}{1-r} \tag{4}$$

pour plusieurs valeurs de $r \in (0, 1)$.

1. La série est calculée avec une boucle `for`. Sommer un nombre N de termes et vérifier que le résultat de la somme s'approche à la valeur limite.

2. La série est calculé avec une boucle `while`. A chaque itération i calculer l'erreur absolue au pas i (estimée comme le module du terme i). Sortir de la boucle lorsque l'erreur relative est inférieure pendant 10 itération successives à une précision voulue (par exemple 10^{-6}).
3. Visualiser l'erreur relatif vs. le nombre de termes retenus dans la série. L'erreur relatif est défini comme le rapport entre l'erreur absolu à l'itération i et le module du résultat de la série. Essayer une échelle linéaire (`plot`), semi-logarithmique (`semilogy`), et logarithmique (`loglog`).

1.6 *Bonus* : Produits scalaires

Calculer le produit scalaire entre

1. un sinus $\sin(t)$ et un cosinus $\cos(t)$ en $[0, 2\pi]$,
2. deux Gaussiennes $e^{-\frac{t}{2\sigma_1^2}}$ et $e^{-\frac{t}{2\sigma_2^2}}$ avec variances $\sigma_1 = 2$ et $\sigma_2 = 4$.

Comparer les résultats Matlab avec les résultats analytiques.

2 Séries de Fourier

2.1 Définition et visualisation d'un signal périodique

Étudier les parties du code `TP2.m` dédiées à la définition du signal périodique $f(t)$ et de sa visualisation. Modifier le nombre des périodes visualisées (1, 3 et 5).

2.2 Calcul d'une série de Fourier

Étudier la partie du code `TP2.m` dédiées au calcul d'une série de Fourier. Visualiser les sommes partielles avec 1, 2, 3, 10, 100 harmoniques.

2.3 Visualisation du spectre d'un signal périodique

Étudier la partie du code `TP2.m` dédiées à la visualisation spectrale (en module). Calculer les coefficients c_n à partir des coefficients b_n donnés dans la partie précédente du code et les visualiser en module. Ajouter une figure pour visualiser la phase des coefficients c_n. Commenter l'allure du spectre de la fonction périodique en relation à sa régularité.

2.4 Convergence simple

Utiliser la boucle `for` pour calculer la série de Fourier de $f(t)$ en un point où la fonction est discontinue et en un point où la fonction est continue. Modifier le nombre d'harmoniques retenues dans la série et commentez les résultats.

Utiliser maintenant le boucle `while` pour sortir de la série lorsque l'erreur relatif est inférieur à 10^{-6} (suivre ce que vous avez fait en TP1, Exercice 1.5). Visualiser l'erreur relatif vs. le nombre d'harmoniques retenues dans la série comme dans la série de l'Exercice 1.5.

2.5 Convergence uniforme

Choisir un intervalle où étudier la convergence uniforme de la série de Fourier (l'intervalle $[0, T)$ devrait être le plus simple).

Visualiser l'écart entre la somme de N harmoniques et la fonction périodique pour plusieurs valeurs de N :

$$\mathrm{ecart}_N(t) = \left| \sum_{n=1}^{N} b_n \sin(n\omega t) - f(t) \right| \tag{5}$$

A partir du graphe, visualiser le max de l'écart en $[0, T]$. Il est atteint en quelle valeur de t ? Comment change-t-il avec N ? Commenter.

Définissons l'erreur uniforme `err_unif`. Il s'agit de la distance uniforme :

$$d_{\infty N} = \max_{t \in I} \{\mathrm{ecart}_N(t)\} \tag{6}$$

où I est l'intervalle où l'on étudie la convergence uniforme.

1. Visualiser $d_{\infty N}$ vs. N en prenant un intervalle ou $f(t)$ est continue.
2. Visualiser $d_{\infty N}$ vs. N en prenant un intervalle ou $f(t)$ est discontinue.

A-t-on convergence uniforme dans les deux cas ?

2.6 Convergence en énergie

Choisir un intervalle où étudier la convergence en énergie de la série de Fourier (l'intervalle $[0, T)$ devrait être le plus simple).

Visualiser le carré de l'écart entre la somme de N harmoniques et la fonction périodique $f(t)$ pour plusieurs valeurs de N :

$$\text{ecart}_N^2(t) = \left| \sum_{n=1}^{N} b_n \sin(n\omega t) - f(t) \right|^2 \tag{7}$$

Comment le carré d l'écart change-t-il avec N ? Commenter.

Définissons l'erreur L^2 `errL2`. Il s'agit de l'intégrale

$$d_{L^2 N} = \sqrt{\int_I \text{ecart}_N^2(t) \mathrm{d}t} \tag{8}$$

où I est l'intervalle où l'on étudie la convergence en énergie).

1. Visualiser $d_{L^2 N}$ vs. N en prenant un intervalle ou $f(t)$ est continue.
2. Visualiser $d_{L^2 N}$ vs. N en prenant un intervalle ou $f(t)$ est discontinue.

Utiliser la commande `area(t_interval_L2,ecart_carre)` pour colorer la surface définie par la fonction $\text{ecart}_N^2(t)$. Garder $N < 100$ pour s'assurer que les intégrales des écarts sont calculées avec une bonne précision.

A-t-on convergence en énergie dans les deux cas ?

2.7 Série de Fourier d'une fonction *continue*

La fonction $g(t)$ de période $T = 1$ est définie par le motif $g_0(t)$

$$g_0(t) = \begin{cases} \left(t - \dfrac{1}{2}\right)^2 & \text{si } t \in [0, 1) \\ 0 & \text{ailleurs} \end{cases} \tag{9}$$

Sa série de Fourier est

$$g^{\text{SF}}(t) = \frac{1}{12} + \sum_{n=1}^{+\infty} \frac{\cos(2\pi n t)}{\pi^2 n^2} \tag{10}$$

— Visualiser les sommes partielles de la série de Fourier associée à $g(t)$ pour 1, 2, 3, 10, 100 harmoniques.

— Visualiser le spectre de $g(t)$ sur le même graphe que le spectre de $f(t)$ (Exercice 2.3). Expliquer les différences.

3 Transformée de Fourier et Convolutions

3.1 Calcul de transformées de Fourier

Calculer la transformée de Fourier des signaux suivantes :

— $f(t) = \Pi_a(t), \quad a = 4$

— $g(t) = e^{-\frac{t^2}{2\sigma^2}}, \quad \sigma = 2$

— $h(t) = \begin{cases} \frac{1}{2} & \text{si } |t| \in \left[0, \frac{1}{2}\right] \\ \frac{1}{2}\left(\frac{3}{2} - |t|\right) & \text{si } |t| \in \left[\frac{1}{2}, \frac{3}{2}\right] \\ 0 & \text{ailleurs} \end{cases}$ (utiliser la fonction `trapeze.m` avec $a = 1, b = 2$)

Pour les calculs et les visualisations, utiliser les fonctions de visualisation et les paramètres définis dans le fichier `TP3_1_Fourier.m`.

1. Dessiner le graphe des signaux $f(t), g(t), h(h)$.
2. Calculer leurs transformées avec la fonction `Fourier_transform.m` et en dessiner les parties réelles et imaginaires, le module et la phase.
3. Pour tous les signaux étudiés, on connaît l'expression analytique de transformées de Fourier. Dans chaque cas, dessiner le graphe de la transformée analytique superposés au graphe de la transformée calculée par Matlab.
4. Étudier comment la transformée de Fourier change sous des petites variations de
   (a) la largeur de la porte
   (b) l'amplitude de la porte
   (c) la variance de la Gaussienne
   (d) les paramètres a et b du trapèze, et notamment lorsque le trapèze s'approche à un triangle et à une porte.

3.2 *Bonus* : Translation

Calculer la transformée de Fourier d'une porte translatée $\Pi_4(t-\tau)$, avec $\tau = 0.2$. Dessiner le graphe de la transformée analytique superposé au graphe de la transformée calculée par Matlab (en module et phase).

Commenter l'effet de petites variations du retard τ.

3.3 *Bonus* : Modulation

Calculer la transformée de Fourier d'une porte modulée $\Pi_4(t)\cos(2\pi\nu_0 t)$, avec $\nu_0 = 15$. Dessiner le graphe de la transformée analytique superposés au graphe de la transformée calculée par Matlab (en module).

Commenter l'effet de petites variations de la fréquence ν_0.

3.4 Convolutions

3.4.1 Deux portes en temps

Ouvrir le fichier `TP3_2_Convolution.m` et utiliser ce code pour calculer la convolution entre deux portes $\Pi_4(t)$ et $\Pi_2(t)$.

Pour ce calcul il faudra compléter la définition de la fonction $\Pi_2(t-\tau)$, inversée et translatée.

Dessiner les deux fonctions et leur produit à l'aide du code de visualisation déjà écrit dans la boucle `for`. Visualiser les trois cas de supports disjoints, partiellement superposés et entièrement superposés. (On considère seulement $t \geq 0$ grâce à la parité des portes.)

Dessiner le résultat de la convolution numérique faite en Matlab superposé avec le résultat connu de la convolution de deux portes.

3.4.2 Deux portes en fréquence

Écrire la transformée de Fourier de la convolution $\hat{f}(\nu) = \mathcal{TF}\left[\Pi_4(t) * \Pi_2(t)\right]$. Utiliser le théorème de la transformée d'une convolution !

Utiliser la dernière partie du fichier `TP3_2_Convolution.m` pour calculer la transformée inverse de $\hat{f}(\nu)$ et la comparer sur le même graphe avec le résultat de la convolution calculé à la question précédente 3.4.1.

3.4.3 *Bonus* : Deux trapèzes en temps

Calculer maintenant la convolution entre deux trapèzes égales (utiliser la fonction `trapeze.m`) de paramètres $a = 1$ et $b = 2$.

Dessiner les deux fonctions et leur produit à l'aide du code de visualisation déjà écrit dans la boucle `for`. Visualiser les cinq cas de supports disjoints, partiellement superposés (trois cas différents !) et entièrement superposés. (On considère seulement $t \geq 0$ grâce à la parité des trapèzes.)

Dessiner le résultat de la convolution.

3.4.4 *Bonus* : Deux trapèzes en fréquence

Écrire la transformée de Fourier de la convolution entre les deux trapèzes. Utiliser toujours le théorème de la transformée d'une convolution !

Utiliser la dernière partie du fichier `TP3_2_Convolution.m` pour calculer la transformée inverse et la comparer sur le même graphe avec le résultat de la convolution calculé à la question précédente 3.4.3.

4 Échantillonnage

4.1 Échantillonnage idéal d'un signal

On veut échantillonner le signal suivant :

$$f(t) = \sin(2\pi\nu_1 t) + 0.3\cos(2\pi\nu_2 t) \tag{11}$$

avec $\nu_1 = 11$ et $\nu_1 = 3$, dans l'intervalle $[0, 2]$.

Déterminer selon le théorème de Shannon une valeur de période d'échantillonage T_{ech} qui permet une reconstruction parfaite du signal, et une valeur de période qui ne permet pas de reconstruire le signal.

4.1.1 Signal bien échantillonné et sa reconstruction

1. Visualiser le signal temps-continu $f(t)$ et le signal échantillonné avec le code utilisé dans l'Exercice 2.3.
2. Calculer le spectre du signal échantillonné $\hat{f}_{\text{ech}}(\nu)$ avec la fonction Matlab `fft`, qui calcule la transformée de Fourier pour les signaux échantillonnés. Visualiser le spectre du signal échantillonné et commenter. Reconnaissez-vous les fréquences du signal $f(t)$ dans ce spectre ?
3. On veut maintenant reconstruire le signal. Filtrer le spectre du signal échantillonné pour éliminer les répliques dues à l'échantillonnage. Expliquer quel type de filtrage est nécessaire et visualiser le spectre filtré $\hat{g}(\nu) = \mathcal{F}\left\{\hat{f}_{\text{ech}}(\nu)\right\}$.
4. Utiliser la fonction Matlab `ifft` pour inverser le spectre filtré $\hat{g}(\nu)$. La sortie de cette fonction est l'ensemble des échantillons $g(nT_{\text{ech}})$ du signal reconstruit $g(t)$. Lorsque vous utilisez la `ifft` il faut diviser le résultat par T_{ech}.
   Visualiser les échantillons sur le signal originaire.
5. A partir des échantillons $g(nT_{\text{ech}})$, reconstruire le signal $g(t)$ comme somme de fonctions sinc. Visualiser le signal $g(t)$ sur le graphe de la question précédente, avec les échantillons et le signal originaire $f(t)$. Si vous avez bien calculé la période d'échantillonage, $g = f$.
6. Visualiser ainsi les premiers trois termes du développement en sinc sur le même graphe $f(t)$ sur une échelle adaptée.

4.1.2 Signal sous-échantillonné et sa reconstruction

Refaire l'Exercice 4.1.1 avec une période d'échantillonage qui ne respecte pas le théorème de Shannon.

4.2 *Bonus* : Échantillonage réel d'un signal

Compléter le code dans le fichier `peigne_portes.m` pour définir une fonction échantillonnée $f_\delta(t)$ à l'aide d'un peigne de portes au lieu d'un peigne de Dirac.

Visualiser les signaux $f(t)$ et $f_\delta(t)$ sur le même graphe.

Attention : Écrire cette dernière partie du code dans un fichier séparé comme les fonctions utilisées pendant les séances précédentes (`Heaviside_step.m`, `porte.m`, `trapeze.m`, `Fourier_transform.m`). Coller le code aussi dans le fichier principale en commentaire, afin qu'il soit inclus dans le compte rendu.