Signaux avec Matlab

Durée: 4 périodes (1 séance de laboratoire)
Travail **individuel**

1. Objectifs

L'exercice a pour objectifs la compréhension et la mise en œuvre des concepts suivants :

- Utilisation de Matlab en général et de fonctions spécifiques en particulier.
- Construction et affichage de signaux élémentaires.
- Opérations sur les signaux.

2. Environnement

La donnée et les fichiers additionnels éventuels sont disponibles sous Moodle, cours *Signaux & Systèmes*, section *Exercices dirigés*, dossier *ExMatlab1*.

Le travail à réaliser sera fait entièrement dans l'environnement de Matlab à l'aide d'un ou de plusieurs script(s).

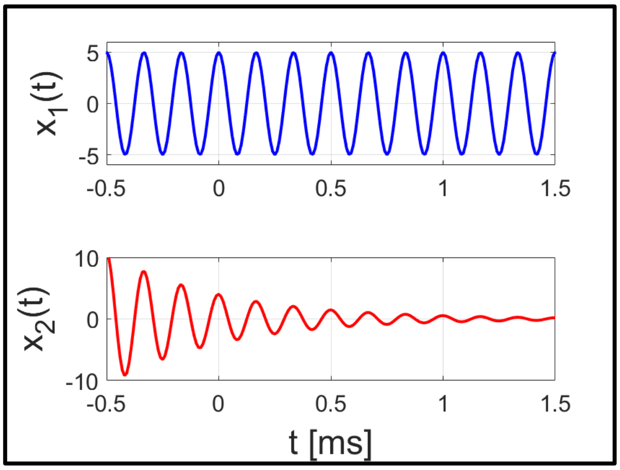
3. Travail à réaliser

Créez un nouveau script Matlab et sauvez-le en le nommant de la manière suivante:

ex1_<votre nom>.m

3.1 Affichage de signaux

Afin de se familiariser avec la construction des signaux et leur affichage dans Matlab. Générez les signaux définis ci-dessous et développez le script pour que l'affichage corresponde à celui-ci-contre.



$$\left.\begin{aligned} x_1(t) &= 5\cos(2\pi f\cdot t) \\ x_2(t) &= 4\,e^{-\frac{t}{\tau}}\cos(2\pi f\cdot t) \end{aligned}\right\} \quad \text{où: } \begin{aligned} f &= 6kHz \\ \tau &= 0.5ms \end{aligned}$$

pour $-1[s] < t < 1[s]$ avec une résolution de $10\mu s$.

Pour l'affichage, intéressez-vous notamment aux fonctions plot(), subplot(), x/ylim(), x/ylabel() et grid on de Matlab.

3.2 Signaux discrets:

1) Créez un nouveau script delta.m et sauvez-le au même endroit que votre script principal. Puis, développez-y **une fonction** ayant un vecteur n en paramètre d'entrée et retournant l'impulsion unité discrète en sortie.

> $Impulsion\ unité\ discrète$:
> $$\begin{cases}\delta[n] = 1\ si\ n = 0 \\ \delta[n] = 0\ si\ n \neq 0\end{cases}$$

2) Créez un vecteur n de -5 à 5 et utilisez votre fonction delta() pour générer le signal $\delta[n+1]$. A l'aide de stem(), affichez correctement le signal créé.

3) Depuis le signal $\delta[n+1]$ et en utilisant la fonction cumsum(), créez le signal $u[n+1]$. Affichez $u[n+1]$ sur le même graphique que $\delta[n+1]$, mais avec une couleur différente.

4) Avec le signal $u[n+1]$, créez le signal $r[n+1]$ et affichez-le également sur le même graphique avec un couleur différente.

3.3 Signaux continus:

1) En utilisant et modifiant le code ci-contre qui définit et affiche le saut unité, créez et affichez correctement la fonction $u(t-0.5)$ pour $-4[s] < t < 4[s]$ avec une résolution de 1ms.

   Qu'est-ce que permet la commande syms? et la fonction eval()?

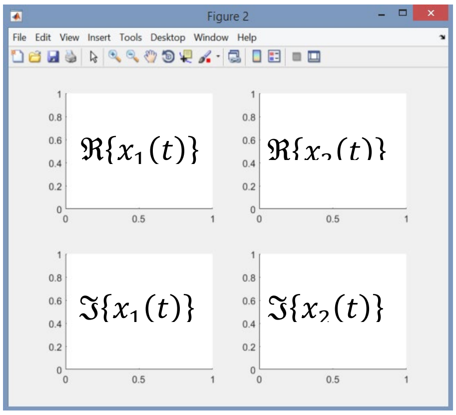
```
% Saut unité
syms t;
u=heaviside(t);
t=(-5:1e-3:5);
plot(t,eval(u));
```

2) Créez le signal $\delta(t-0.5)$ depuis $u(t-0.5)$ en utilisant la fonction diff() et affichez-le sur le même graphique en couleur différente. En zoomant sur le graphique, regardez qu'elle est la valeur affichée de $\delta(t-0.5)$ pour $t = 0.5$. Qu'en déduisez-vous?

3) Depuis le signal $u(t-0.5)$ et en utilisant la fonction int(), créez le signal $r(t-0.5)$. Affichez-le aussi sur le même graphique avec une couleur différente.

Est-ce que ces signaux affichés sont continus ou discrets?

4. Exponentielle complexe, real() et imag()



Générez les signaux suivant:

$$x_1(t) = \mathrm{A} \cdot e^{(\sigma + j\omega)t}$$

$$x_2(t) = A \cdot \left(e^{(\sigma + j\omega)t} + e^{(\sigma - j\omega)t}\right)$$

où $\mathrm{A} = 4$, $\sigma = -100$, $\omega = 2 \cdot \pi \cdot 200Hz$
et $-50ms < t < 100ms$ avec une résolution de $100\mu s$.

En utilisant les fonctions subplot(), real() et imag() affichez ces signaux selon le canevas ci-contre. Pour l'affichage, limitez l'axe du temps de $0ms$ à $+30ms$.

Pourquoi les amplitudes de $\Re\{x_1(t)\}$ et de $\Re\{x_2(t)\}$ sont différentes alors que A est identique? Pourquoi la partie imaginaire du signal complexe $x_2(t)$ est à 0? Que se passe-t-il lorsque l'on varie la valeur de σ ou de ω?

5. Optionnel

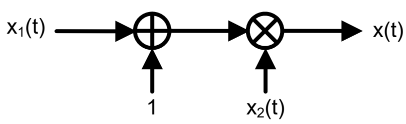
Application – modulation AM

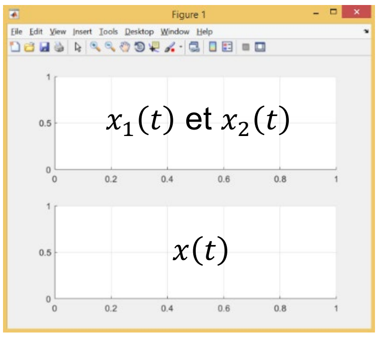
1) Générez les signaux suivant:

$$x_1(t) = 0.4 \cdot cos(2\pi 250t) + 0.2 \cdot \cos(2\pi 500t + 1)$$

$$x_2(t) = 0.05 \cdot cos(2\pi 10'000t)$$

pour $0[s] < t < 5[s]$ avec une résolution de $5\mu s$. Puis, implémentez la fonction définie par le schéma fonctionnel ci-dessous pour générer $\mathrm{x}(t)$:





2) En utilisant la fonction subplot(), affichez $x_1(t)$ et $x_2(t)$ sur le graphique du haut et $x(t)$ sur le graphique du bas d'une seule fenêtre (cf. ci-contre). Liez les axes des abscisses à l'aide de la fonction linkaxes().

   Utilisez le zoom pour observer les différents signaux.

   Que constatez-vous sur le signal $x(t)$ par rapport à $x_1(t)$?

6. Références

Notes du cours Signaux & systèmes 1, chapitre “Signaux”
Aide en ligne de Matlab : https://ch.mathworks.com/matlabcentral/