Première partie

Rappels théoriques et exercices

Chapitre 1

Signaux et systèmes : représentation et opérations

1.1 Rappel théorique

Signaux et Systèmes

En toute généralité,

- un *signal* $u\{\cdot\}$ est une application associant des valeurs $u\{q\} = m$, prises dans un ensemble M (l'image de u), aux valeurs de la variable indépendante q variant dans un ensemble $\mathcal{Q}$ (le domaine de u).

- un *système* $\mathcal{S}$ est un processus associant un ou plusieurs signaux de sortie à un ou plusieurs signaux d'entrée. Ces signaux peuvent être de types tout à fait différents. Il se peut également que différentes sorties soient possibles pour une même entrée ; dans ce cas, on dit que le système n'est pas univoque.

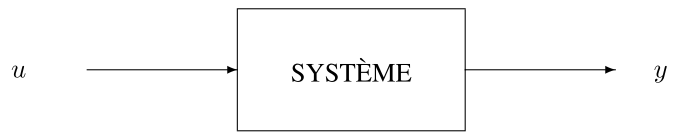
Dans ce cours, la variable indépendante q est toujours unidimensionnelle et ordonnée. On abordera deux types de systèmes.

1. Systèmes dans des espaces vectoriels :
   - les *signaux* prennent leurs valeurs (image) dans un espace vectoriel ($\mathbb{R}$ ou $\mathbb{C}$ - *cfr.* cours). La variable indépendante peut être
     – discrète, lorsque le domaine de u est $\mathbb{Z}$ (on écrira $u[\cdot]$) ou
     – continue, lorsque le domaine de u est $\mathbb{R}$ (on écrira $u(\cdot)$).
     On parlera respectivement de signal en temps discret ou en temps continu, même lorsque la variable indépendante n'est pas le temps.

   - les *systèmes* étudiés agissent sur les signaux définis ci-dessus. Nous n'étudierons en détail que les systèmes SISO (Single Output - Single Input), admettant un signal d'entrée scalaire u et un signal de sortie scalaire y, bien que la généralisation aux systèmes MIMO (Multiple Input - Multiple Output) soit assez immédiate.

2. Automates finis :
   - les signaux ont pour domaine les naturels $0, 1, 2, ...$ (on écrira $u[n]$) et prennent des valeurs dans des *ensembles finis de symboles abstraits* communément appelés *alphabets*. Les alphabets d'entrée et de sortie d'un système peuvent être différents.
   - les systèmes (plus souvent appelés *automates*) possèdent une *variable interne* x, aussi appelée *variable d'état* ou *état*, prenant des valeurs dans un ensemble fini de symboles. L'évolution de la

variable d'état définit un *signal d'état*. A chaque itération n, en fonction du symbole fourni par le signal d'entrée $u[n]$ et de son état actuel $x[n]$, l'automate produit le symbole du signal de sortie $y[n]$ et son nouvel état interne $x[n+1]$. Comme pour tout modèle d'état (cfr. chapitres 2 et 4), il est nécessaire de préciser également *l'état initial*, i.e. le symbole que prend la variable interne x pour $n = 0$, pour pouvoir prédire l'évolution du système.



La plupart des outils développés ne sont applicables qu'aux systèmes *linéaires et invariants*, notés LTI (Linear and Time Invariant), qui ne sont définis que pour les systèmes dans des espaces vectoriels (*cfr.* chapitre 3). Il est cependant important de se remémorer que les systèmes LTI ne sont qu'une classe particulière de systèmes, résultant souvent d'une idéalisation de la réalité (par exemple par linéarisation dans le voisinage d'un comportement de référence, *cfr.* chapitre 4). Il est donc naturel d'utiliser des moyens de représentation mathématique convenant à des classes plus larges de systèmes (*cfr.* chapitre 2).

Transformations de signaux par la variable indépendante

Dans cette section, nous revoyons rapidement quels sont les effets d'une transformation de la variable indépendante sur le graphe d'un signal, *i.e.* comment le graphe de $y(t) = u(\tau(t))$ (respectivement $y[n] = u[\tau[n]]$) est lié à celui de $u(t)$ (respectivement $u[n]$). Nous nous attacherons plus particulièrement au cas des transformations affines

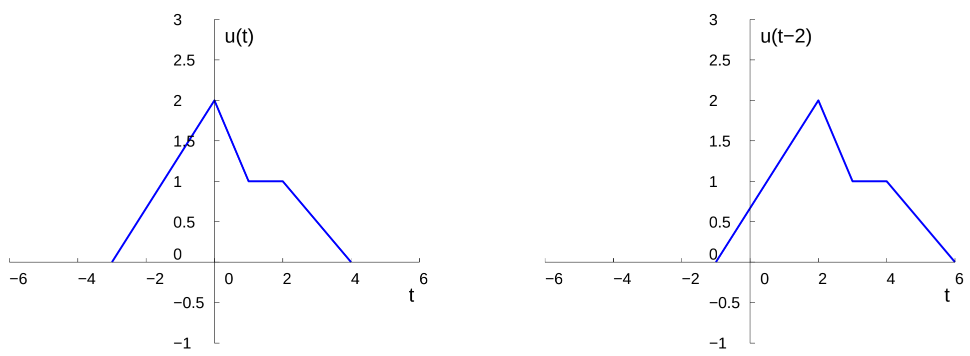
$$\begin{array}{ll} \tau(t) = \frac{t}{a} - \sigma & \text{ou} \\ \tau[n] = \frac{n}{a} - \sigma & . \end{array}$$

Décalage temporel

Il s'agit du cas $a = 1$. Nous avons donc les transformations

$$\begin{array}{lcll} u(t) & \longrightarrow & u(t-\sigma) & , \sigma \in \mathbb{R} \\ u[n] & \longrightarrow & u[n-\sigma] & , \sigma \in \mathbb{Z} \end{array}$$

Le signal transformé est égal au signal d'origine, décalé de σ vers la DROITE.



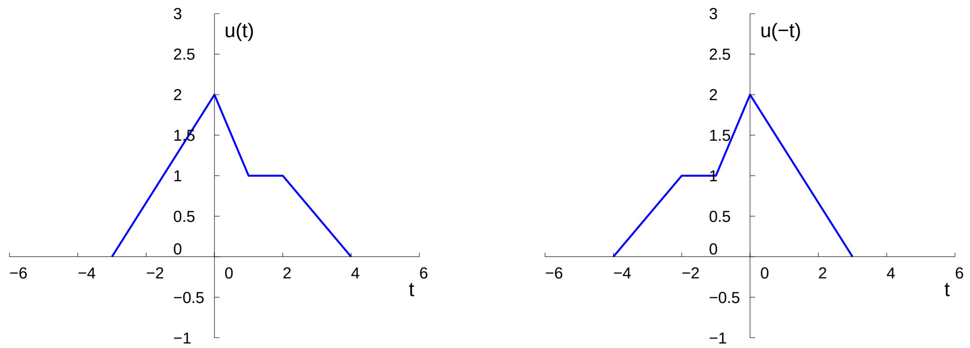
$\sigma > 0 \Rightarrow$ retard pur décalage vers la droite.
$\sigma < 0 \Rightarrow$ avance pure décalage vers la gauche.

Inversion temporelle

Il s'agit du cas $\sigma = 0$, $a = -1$. Nous avons donc les transformations

$$\begin{array}{lcl} u(t) & \longrightarrow & u(-t) \\ u[n] & \longrightarrow & u[-n] \end{array}$$

Le signal transformé est simplement le symétrique du signal d'origine par rapport à l'axe $t = 0$ (respectivement $n = 0$).

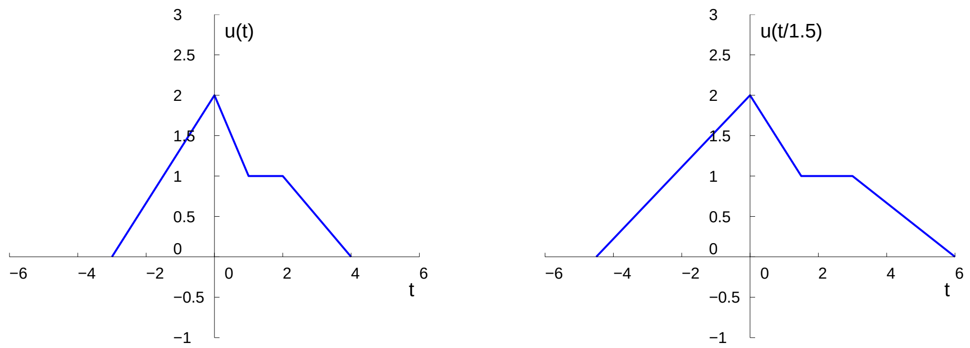


Dilatation ou contraction temporelle

Il s'agit du cas $\sigma = 0$ et $a > 0$. Nous avons alors les transformations

$$\begin{array}{lcll} u(t) & \longrightarrow & u\left(\frac{t}{a}\right) & a \in \mathbb{R}_{>0} \\ u[n] & \longrightarrow & u\left[\frac{n}{a}\right] & a \in (\mathbb{R}_{>0}) \end{array}$$

Le signal transformé est égal au signal d'origine DILATÉ de a par rapport à l'axe $t = 0$ (respectivement $n = 0$)



$$\begin{array}{ll} a > 1 \Rightarrow & \text{dilatation.} \\ a < 1 \Rightarrow & \text{contraction.} \end{array}$$

NB : Dans le cas discret, on prend la convention $u[\tau] = 0$ lorsque τ n'est pas un entier. Ainsi, pour $a = 2$, le signal transformé $u[\tau[n]]$ sera nul pour n impair. Similairement, comme il n'est pas permis d'attribuer une valeur au signal $u[\tau[n]]$ pour n non-entier, toute information contenue dans $\{u[n] : \tau[n] \cap \mathbb{Z} = \emptyset\}$ sera perdue après transformée. Cela implique que

– pour a irrationnel, toute l'information est perdue : $u[\tau[n]] \equiv 0$;
– toute l'information est gardée ssi $a \in \mathbb{N}_{>0}$.

Transformation affine générale

Dans ce cas se pose la question de savoir dans quel ordre il faut combiner les opérations que nous venons d'examiner ; une première remarque évidente est que les opérations d'inversion temporelle et de dilatation / contraction commutent entre elles. En ce qui concerne le décalage, l'on s'aperçoit que le résultat respecte **l'ordre inverse des priorités usuelles des opérations** (parenthèses, multiplications/divisions, additions/soustractions). En effet, écrivant les deux alternatives

$$\begin{array}{llll} [u(t) \longrightarrow y_1(t) = u(t-\sigma) & ; \; y_1(t) \longrightarrow y_2(t) = y_1\left(\frac{t}{a}\right)] & \Rightarrow & y_2(t) = y_1\left(\frac{t}{a}\right) = u\left(\frac{t}{a} - \sigma\right) ; \\ [u(t) \longrightarrow y_1(t) = u\left(\frac{t}{a}\right) & ; \; y_1(t) \longrightarrow y_2(t) = y_1(t-\sigma)] & \Rightarrow & y_2(t) = y_1(t-\sigma) = u\left(\frac{t-\sigma}{a}\right) , \end{array}$$

on obtient
– pour le premier cas, la transformation $t \to t/a - \sigma$ en commençant par l'addition (et non la division) ;
– pour le second cas, la transformation $t \to (t-\sigma)/a$ en commençant par la division (et non la parenthèse).

1.2 Exercices 1 - 4.

Exercice 1 - (+ Ex 28) - Cherchez les différents signaux que vous pouvez associer à une image photographique. Dans chaque cas, identifiez le domaine et l'image du signal et discutez son adéquation avec les signaux traités dans le cours de système. Imaginez des systèmes agissant sur ces différents signaux ; spécifiez clairement les signaux d'entrée et de sortie.

a) - (Exemple) - Une image continue en niveaux de gris peut être représentée par le signal $u(x, z)$ où x et z désignent les coordonnées horizontale et verticale et où u décrit l'intensité.
Les variables x et z sont continues ($\Rightarrow$ signal en temps continu) et prennent leurs valeurs dans $[0, X_{max}]$ et $[0, Z_{max}]$ respectivement. Ce signal a donc un domaine de dimension 2 ; il faut se restreindre à une dimension (suivre une courbe sur l'image) afin de pouvoir le traiter avec les outils vus dans ce cours, à moins de généraliser ces outils à plusieurs dimensions.
Le signal u prend en général ses valeurs dans $[0, 1]$, qui n'est pas un espace vectoriel. Il est cependant possible, en excluant les points 0 et 1, d'effectuer un changement de variable $(0, 1) \to (-\infty, +\infty)$ par exemple à l'aide d'une fonction tangente.

b) Essayez de faire un raisonnement similaire pour une image couleur.

c) Que pouvez-vous dire si l'on considère une image composée d'un nombre fini de pixels (u prenant toujours ses valeurs dans un ensemble continu de $\mathbb{R}$) ? En particulier, essayez de vous ramener à un signal répondant aux hypothèses de ce cours (notamment, ne dépendant plus que d'une seule variable).

d) Qu'en est-il du cas d'une image stockée dans un ordinateur, *i.e.* comportant un nombre fini de pixels et un nombre fini de couleurs ?

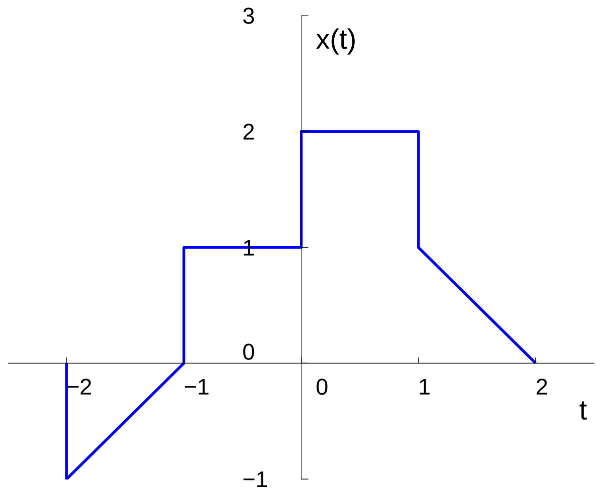
e) - (Exemple) - Un exemple de système agissant sur une image en niveaux de gris est un filtre d'intensité. Dans le cas du signal continu en (x, z) et avec $u \in (0, 1)$, nous pourrions imaginer de diviser l'intensité par 2 en chaque point ("effet lunettes solaires"). On aurait alors $y_1(x, z) = u(x, z)/2$. Ce système est SISO ; un changement de variable différent sera cependant nécessaire sur $y_1(x, z)$ pour qu'il prenne ses valeurs dans un espace vectoriel. Une autre possibilité serait $y_2(x, z) = (u(x, z))^2$, qui prend ses valeurs dans $(0, 1)$ comme le signal d'entrée ; dans ce cas, le même changement de variable peut être utilisé sur u et y_2.

f) Que dire de la faisabilité du système ci-dessus si l'on dispose d'un nombre fini de couleurs ?

g) Comment définir un système similaire sur une image couleur ?

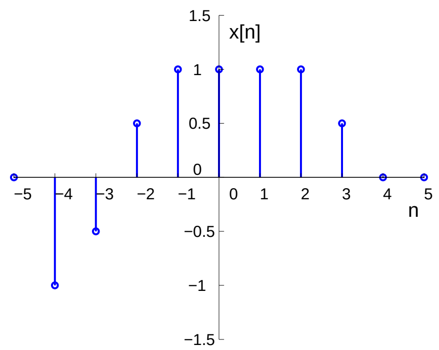
h) Considérant un signal du type de c), écrivez la relation entrée-sortie pour un système associant à chaque pixel la moyenne de ses voisins et analysez ces entrée et sortie.

Exercice 2 - Soit le signal continu $x(t)$ représenté ci-dessous.



Tracer les signaux suivants.
a) $x(t-1)$
b) $x(2-t)$
c) $x(2t+1)$
d) $x\left(4-\frac{t}{2}\right)$
e) $(x(t)+x(-t))\,\mathbf{I}_{+}(t)$
f) $x(t)\left(\delta(t+\frac{3}{2})-\delta(t-\frac{3}{2})\right)$

Exercice 3 - Soit le signal discret $x[n]$ représenté ci-dessous.



Tracer les signaux suivants.
a) $x[n-4]$

b) $x[3-n]$
c) $x[3n]$
d) $x[3n+1]$
e) $x[n]\mathbf{I}_+[2-n]$
f) $x[n-2]\delta[n-2]$
g) $\frac{1}{2}x[n] + \frac{1}{2}(-1)^n x[n]$
h) $x[(n-1)^2]$

Exercice 4 - Imaginer des situations réelles où un signal de sortie est obtenu par transformation affine de la variable indépendante du signal d'origine. Considérer les cas discret et continu, où t et n représentent le temps ou une autre variable.

1.3 Applications MATLAB©

MATLAB© est un logiciel permettant de faire des calculs mathématiques "semi-formels". MATLAB© connaît un grand nombre d'opérations ou de fonctions mathématiques : fonctions usuelles, calcul matriciel, fonctions plus spécifiques à des domaines techniques,... . Les outils usuels de l'algorithmique (boucles `for` ou `while`, conditions `if ... else`, définition de fonctions,...) peuvent également être utilisés avec de multiples variantes.

Les quelques séances qui vont suivre ont pour but de vous familiariser avec l'utilisation de MATLAB© dans le cadre des signaux et systèmes. Des compléments d'information peuvent être facilement obtenus grâce à l'aide incluse dans le logiciel. L'accès à l'aide s'obtient en tapant `help fonction`, où `fonction` représente le nom d'une fonction prédéfinie de MATLAB© . L'appel à `help` sans argument renvoie la liste des sujets pour lesquels l'aide est disponible, ensuite `help sujet` renvoie la liste des fonctions relatives au sujet. Dans les versions récentes de MATLAB© , il est également possible d'accéder à un menu d'aide interactif, via l'onglet `Help` proposé en entête de la fenêtre principale ou en cliquant sur un hyperlien `doc fonction` apparaissant dans la fenêtre de commande à la fin du commentaire renvoyé par `help fonction`. On peut avoir un aperçu des possibilités de MATLAB© en lançant la commande `demo`.

Les présentes lignes n'ont pas vocation d'exhaustivité en ce qu'elles se bornent aux concepts inhérents aux signaux et systèmes. Il est clair que les possibilités de MATLAB© sont nettement plus importantes que celles décrites dans cet ouvrage et que l'on devra faire appel à la documentation complète pour les utiliser.

Représentation d'une fonction

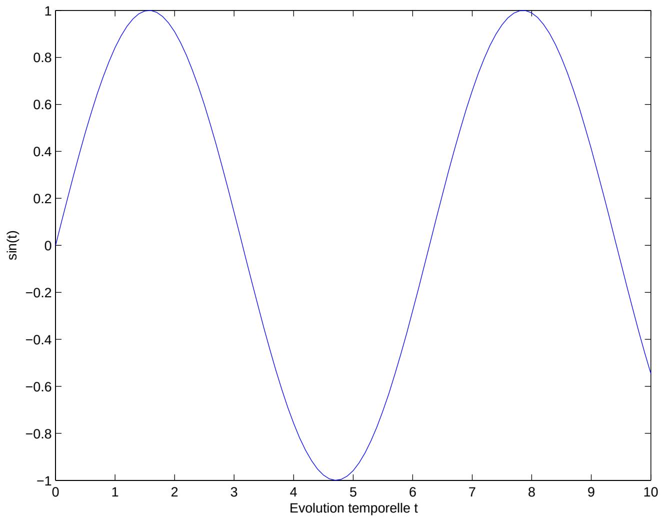
La représentation d'une fonction passe par le stockage des valeurs de cette fonction dans un vecteur ou dans une matrice.

Exemple :

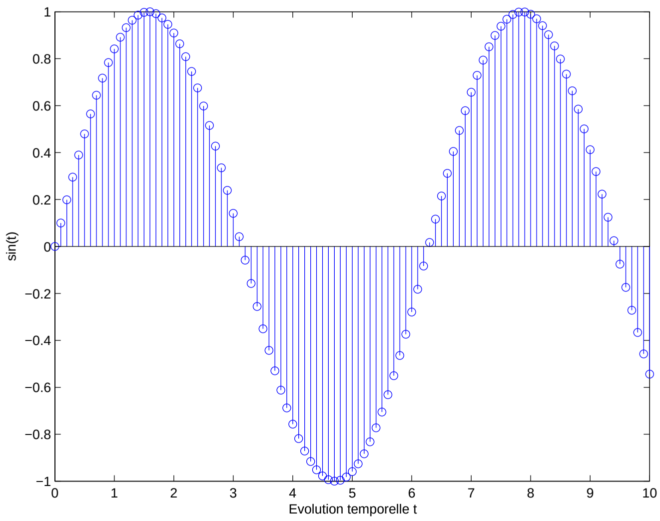
```
t = [0:0.1:10];
y = sin(t);
```

illustre la représentation en mémoire de la fonction $y = \sin t$ lorsque $t \in [0, 10]$. La représentation informatique étant par essence discrète, la précision de la représentation est liée à la fréquence d'échantillonnage, choisie à 10 dans l'exemple. Le choix d'un pas d'échantillonnage suffisamment petit est important en vue d'une approximation correcte d'un signal continu. Le *théorème de Shannon* affirme qu'un signal de largeur de bande limitée est échantillonné sans perte d'information si la fréquence d'échantillonnage est supérieure à 2 fois la largeur de bande. Les effets de l'échantillonnage sont étudiés de manière rigoureuse dans le Chapitre 12 du syllabus théorique.

La fonction `plot(t,y)` permet de représenter les valeurs du vecteur y en fonction des indices du vecteur t, comme illustré sur la figure suivante pour la fonction définie ci-dessus. Cette fonction fournit une représentation continue des valeurs discrètes.



La fonction `stem(t,y)` est le pendant discret de la fonction `plot`. Elle représente, sous forme de barres verticales, la séquence de données du vecteur `y` aux valeurs spécifiées sur l'axe des abscisses par le vecteur `t`.



Résumé des commandes utiles pour le tracé de graphes

`plot(signal)`	Trace sous la forme d'une courbe les composantes du vecteur `signal` : l'axe des x correspond à l'indice de la composante dans le vecteur `signal`, l'axe des y à la valeur de la composante.
`stem(signal)`	Trace sous forme d'un échantillonnage discret les composantes du vecteur `signal`.
`subplot(a,b,c)`	Partage l'écran en une matrice de dimension $a \times b$ et affiche le graphe suivant cette commande dans la région c.
`mesh(A)`	Affiche en 3D les valeurs de la matrice A.
`title('Figure 1')`	Inscrit la chaîne de caractères 'Figure 1' en titre au-dessus du graphe courant.
`xlabel('Abscisses')`	Inscrit la chaîne de caractères 'Abscisses' en regard de l'axe des x.
`ylabel('Ordonnées')`	Inscrit la chaîne de caractères 'Ordonnées' en regard de l'axe des y.
`hold on`	Superpose les graphiques ultérieurs avec l'affichage courant.
`hold off`	Désactive la superposition des graphiques.

Applications

1. a) Créer dans MATLAB© un signal sinusoïdal de 31 échantillons, régi par l'équation $h(t) = g \circ f$ où $f(t) = \frac{t}{2} + 1$ et $g(x) = \sin(x)$.
   b) Représenter graphiquement le signal échantillonné $h[t]$.
   c) Représenter la fonction $h(t)$ pour $t \in [-5, 5]$.
2. a) Concevoir un code MATLAB© permettant d'obtenir le signal impulsion $\delta[n]$.
   b) Générer et tracer les séquences définies ci-dessous en vous servant du point a).
      (i) $x_1[n] = 1.5\delta[n-333], \quad 300 \leq n \leq 350$
      (ii) $x_2[n] = 4.5\delta[n+7], \quad -10 \leq n \leq 0$
3. Générer et tracer la fonction définie ci-dessous.

$$x[n] = \sin\left(\frac{\pi}{\sqrt{23}}n\right), \quad 0 \leq n \leq 50$$

   Le signal $x[n]$ est-il périodique ?
4. Un signal discrétisé en temps est souvent obtenu par échantillonnage d'un signal continu. Soit le signal continu

$$s(t) = A\cos(2\pi f_0 t + \phi) \ .$$

   Si on l'échantillonne à la fréquence $f_e = \frac{1}{T}$, on obtient

$$s[n] = s(t)|_{t=nT} = A\cos(2\pi f_0 nT + \phi) = A\cos\left(2\pi\frac{f_0}{f_e}n + \phi\right) \ .$$

   a) Ecrire une fonction MATLAB© qui génère les échantillons de $s(t)$ pour créer une séquence en temps discret de longueur finie. La fonction utilisera 6 arguments d'entrée : l'amplitude A, la fréquence f_0, la phase à l'origine ϕ, l'instant de début, l'instant de fin, et la fréquence d'échantillonnage f_e.
   b) Utiliser la fonction créée au point a) pour générer les échantillons définis de manière suivante.

Fréquence = 1200 Hz	Fréquence d'échantillonnage = 8000 Hz
Phase initiale = 45 deg	Instant initial = 0 s
Amplitude = 50	Instant final = 7 ms

   c) Faire deux tracés du signal résultant : l'un en fonction du temps t (en ms), et l'autre en fonction des indices d'échantillons n.
   d) Répéter les points b) et c) avec $f_e = 1500\ Hz$.