

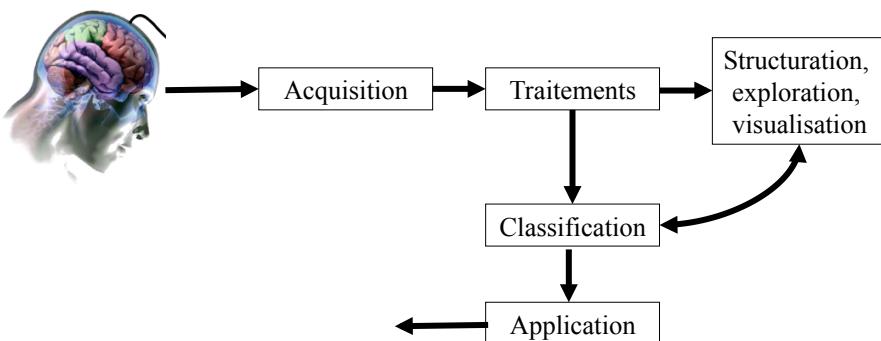
Traitement Numérique des Données

M1 – INF 2163

AIDN: Applications Interactives et
Données Numériques
Sylvie Gibet

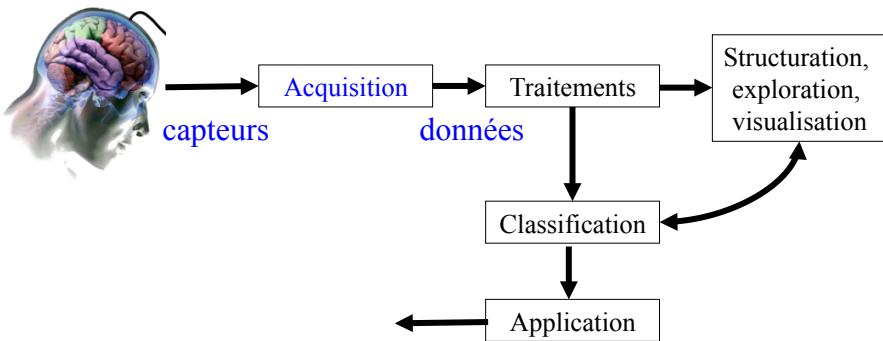
1

Objectifs du cours INF 2163 : traitement numérique des données



2

Objectifs du cours INF 2163 : traitement numérique des données



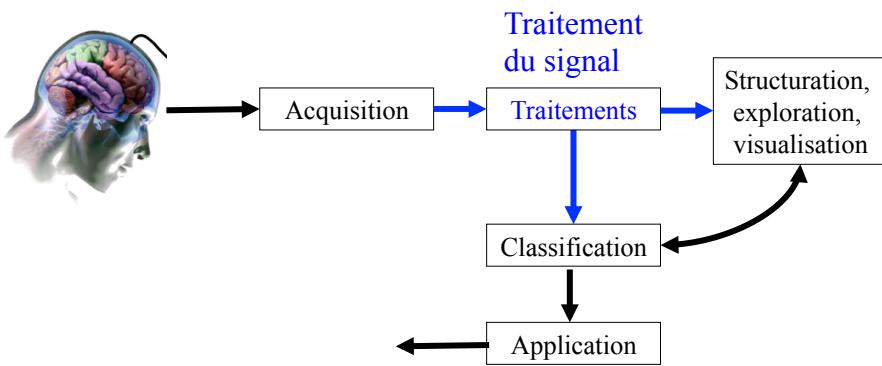
3

Objectifs du cours INF 2163 : traitement numérique des données

- **Données numériques : comment sont-elles capturées, numérisées, enregistrées**
 - Des capteurs vers les données : conversion données analogiques (continues) -> données numériques
 - **Théorie de l'échantillonnage**
 - Fréquence d'échantillonnage (Shannon/Nyquist)
 - Représentation, analyse
 - Ré-échantillonnage (multi-sources)

4

Objectifs du cours INF 2163 : traitement numérique des données



5

Objectifs du cours INF 2163 : traitement numérique des données

■ Traitement du signal :

Représenter, analyser, traiter, extraire de l'information à partir de données

- Changement d'espace de représentation

Représentation de Fourier (séries de Fourier, Transformée de Fourier) : son, image, etc.

- Réduction de dimension (analyse de données)

Linéaire : analyse en Composantes Principales (PCA)

Non linéaire

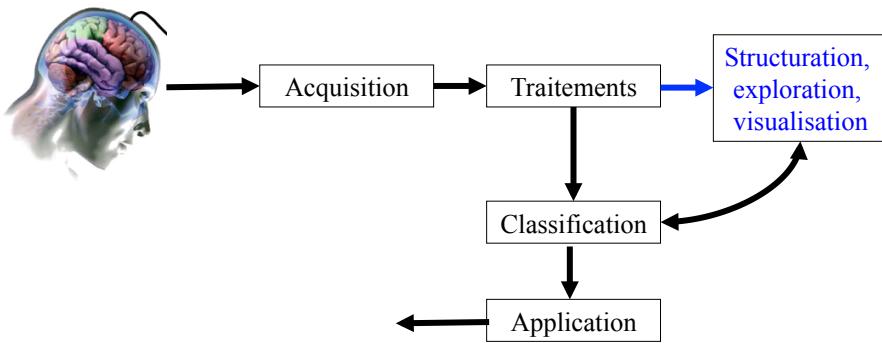
- Filtrage

Filtres linéaires, non linéaires

Filtres spatiaux, temporels

6

Objectifs du cours INF 2163 : traitement numérique des données



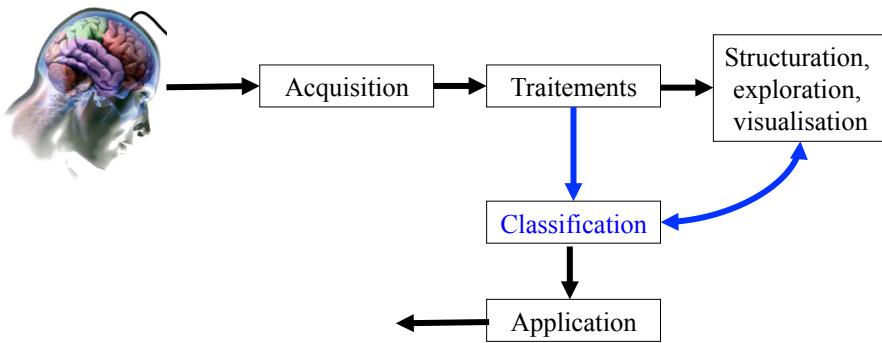
7

Objectifs du cours INF 2163 : traitement numérique des données

- ▣ **Structuration, exploration, visualisation**
 - Accéder aux données
 - Extraire, fusionner des données
 - Faire des tests statistiques sur les données
 - Visualiser les données

8

Objectifs du cours INF 2163 : traitement numérique des données



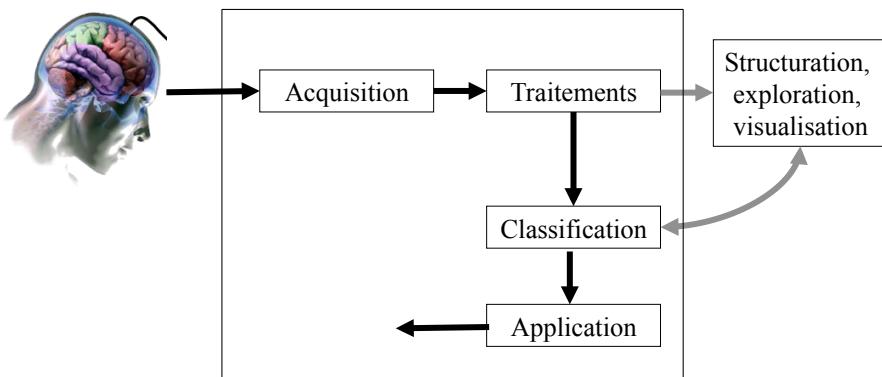
9

Traitement Numérique des Données

- ▣ **Reconnaissance de formes, apprentissage automatique** : prédire, faire émerger des patterns (formes), classifier, faire des recommandations
 - Apprendre à partir des données
 - Prédiction (régression linéaire, ...)
 - Classification supervisée
 - Classification non supervisée : clustering
 - Apprentissage par renforcement

10

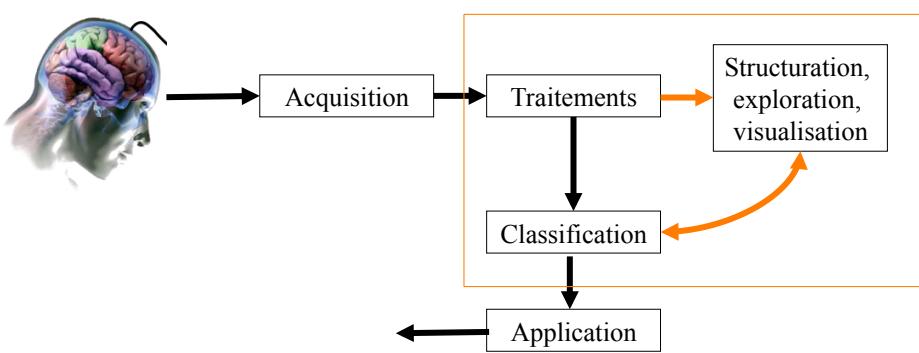
Objectifs du cours INF 2163 : traitement numérique des données



Chaîne de traitement

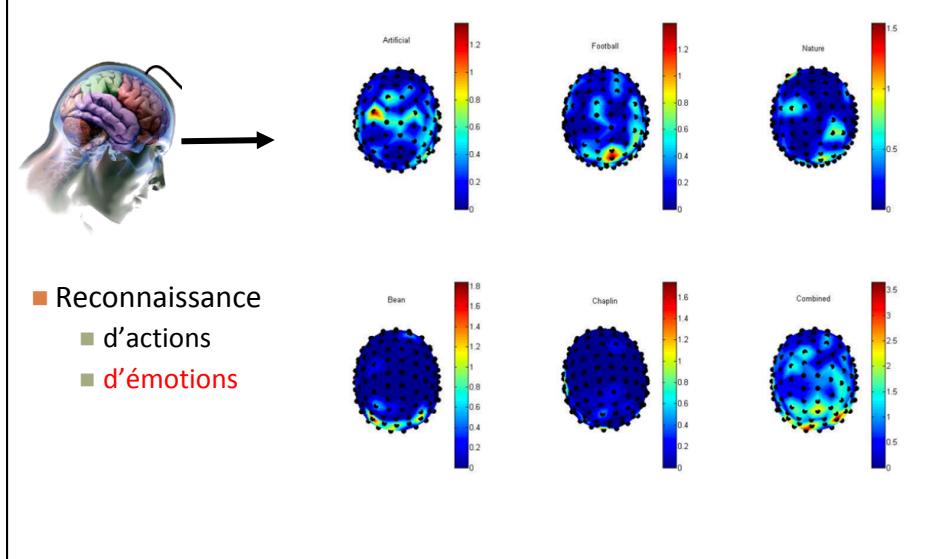
11

Objectifs du cours INF 2163 : traitement numérique des données

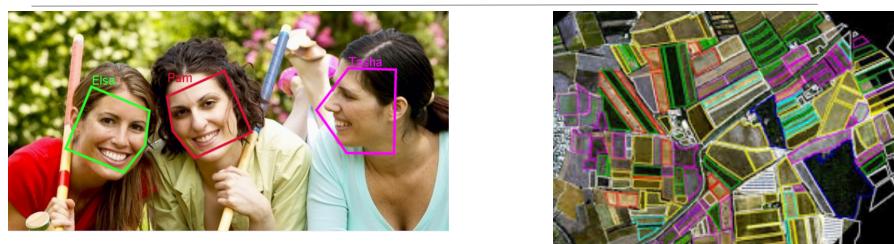


12

Exemple d'application : « mind-reading challenge »



Exemple d'application : détection d'objets dans des images

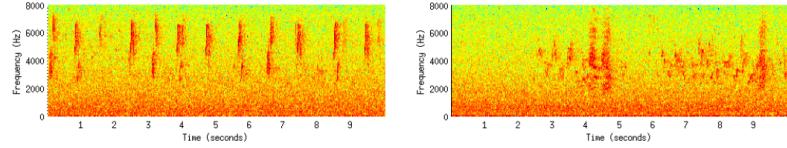


■ Étapes de résolution

- Sélectionner les données permettant de discriminer la/les classes données, les régions
- Capturer, enregistrer, débruiter, corriger les données
- Analyser les données (changement d'espace de représentation)
- Apprendre à partir des données
- Modéliser la frontière de discrimination à partir des données

14

Exemple d'application : classification de sons



■ Reconnaître des sons d'oiseaux enregistrés (IEEE MLSP 2013: Birds competition)

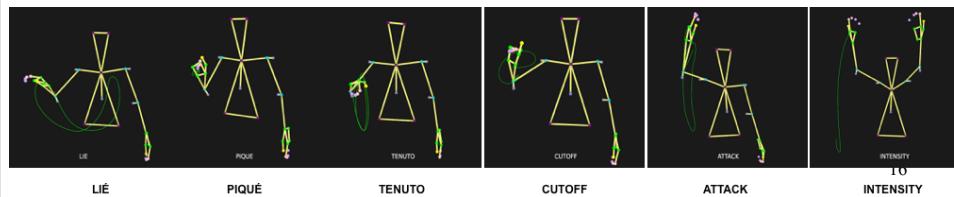
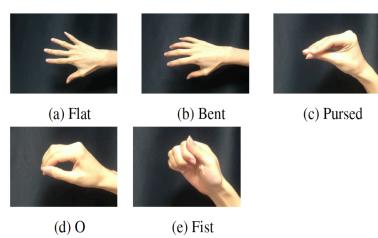
- Clips audio de 19 espèces, les sons se recouvrent (645 10s clips)
- Représentation spectrale (transformée de Fourier)
- Classification :
 - Moitié des exemples : servent à l'apprentissage
 - Challenge : prédire l'autre moitié

15

Exemple d'application : reconnaissance de gestes, mouvement – cours M2 : Mouvement et IA

■ Gestes de chefs d'orchestre

- Détermination des gestes par rapport au contrôle du son
- Extraction des caractéristiques : formes de la main (5), mouvements (6)
- Apprendre à la fois le geste et l'expressivité
- Application : contrôle gestuel de la synthèse sonore



Objectifs du cours INF 2163 - TND

- **Traitement numérique des données**
 - **Données numériques** : théorie de l'échantillonnage
 - **Traitement du signal** (transformée de Fourier discrète) : applications
 - **Structuration, exploration, visualisation** (pandas)
 - Reconnaissance de formes : un aperçu du machine learning et de ses applications
 - Classification, clustering

17

Contrôle des connaissances

- Contrôles
 - Note théorique T1
 - Note pratique P : TP
- Note finale de cette matière (50%) : $M1 = (T1 + P)/2$
- Note de l'UE 1^{ère} session : $(M1+M2)/2$
 - M1 : 50% Traitement des données
 - M2 : 50% Codage
- 2^{ème} session :
 - T'1 remplace T1
 - vous conservez P

18

Programmation en Python (scientifique)

- <http://www.python.org>
- Ipython (interactive shell)
- **Python 3 for scientific programming**
<http://www.courspython.com/>
- **Tutoriel NumPy:**
<https://www.tutorialspoint.com/numpy/index.htm>
- **Doc Numpy:** <http://www.numpy.org/>
- **Doc scipy:** <https://docs.scipy.org/doc/>
- **Doc matplotlib:**
<https://matplotlib.org/tutorials/introductory/pyplot.html>

19

NumPy: www.numpy.org

[Scipy.org](#)

NumPy

NumPy is the fundamental package for scientific computing with Python. It contains among other things:

- a powerful N-dimensional array object
- sophisticated (broadcasting) functions
- tools for integrating C/C++ and Fortran code
- useful linear algebra, Fourier transform, and random number capabilities

Besides its obvious scientific uses, NumPy can also be used as an efficient multi-dimensional container of generic data. Arbitrary data-types can be defined. This allows NumPy to seamlessly and speedily integrate with a wide variety of databases.

NumPy is licensed under the [BSD license](#), enabling reuse with few restrictions.

Getting Started

- [Getting NumPy](#)
- [Installing the SciPy Stack](#)
- [NumPy and SciPy documentation page](#)
- [NumPy Tutorial](#)
- [NumPy for MATLAB® Users](#)
- [NumPy functions by category](#)
- [NumPy Mailing List](#)

For more information on the SciPy Stack (for which NumPy provides the fundamental array data structure), see [scipy.org](#).

[About NumPy](#)

[License](#)

[Old array packages](#)

20

10

www.matplotlib.org

matplotlib

The screenshot shows the official Matplotlib website. At the top is the URL "www.matplotlib.org" and the large "matplotlib" logo with a circular icon. Below the logo is a navigation bar with links: "home | examples | tutorials | pyplot | docs »". The main content area starts with a section titled "Introduction" which includes a brief description of what Matplotlib is and some sample plots. Below this are sections for "Installation" (with a link to installation instructions) and "Documentation" (with a link to the documentation for version 2.1.0). A sidebar on the right contains the number "21".

www.scipy.org

The screenshot shows the official SciPy website. At the top is the URL "www.scipy.org" and the "SciPy.org" logo. Below the logo is a blue banner with the text "Sponsored by ENTHOUGHT". The main content area features five circular icons with arrows pointing to them: "Install", "Getting Started", "Documentation", "Report Bugs", and "Blogs". Below these icons is a paragraph about SciPy being a Python-based ecosystem of open-source software for mathematics, science, and engineering. It lists several core packages with their logos and descriptions: NumPy (Base N-dimensional array package), SciPy library (Fundamental library for scientific computing), Matplotlib (Comprehensive 2D Plotting), IPython (Enhanced Interactive Console), Sympy (Symbolic mathematics), and pandas (Data structures & analysis). At the bottom is a blue button labeled "More information...".

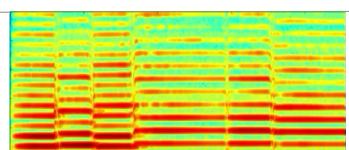
I. Théorie de l'échantillonnage

23

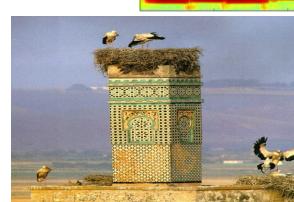
Qu'est-ce qu'un signal ?

- Onde acoustique
 - Son : spectrogramme
- Onde lumineuse
 - Image

2D

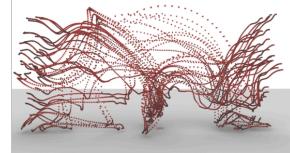
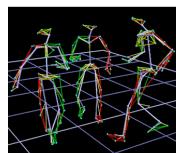


1D



- Capteurs position
 - Mouvement

nd



24

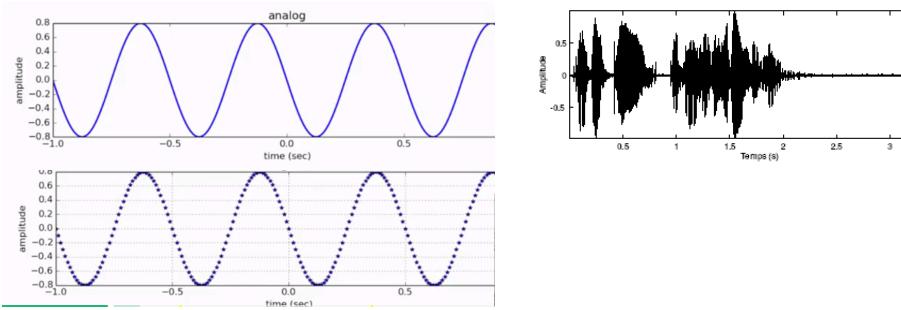
Plan du cours de TDN

- Qu'est-ce qu'un signal ?
 - Signal à temps discret : les capteurs délivrent de l'information au cours du temps, soit à des instants réguliers, soit de manière irrégulière
 - Signal multidimensionnel discret : 2D, 3D, nd
 - Théorie de l'échantillonnage : comprendre les aspects propres à la numérisation du signal
- Programmes en Python
 - Programmer en Python, NumPy, matplotlib
 - TP signal (scipy.signal)

25

Signal : analogique vs. numérique

- son : pression de l'air en fonction du temps
 - Son pur : 1 seule fréquence (onde sinusoïdale)
 - Son quelconque : 1 ensemble de fréquences



26

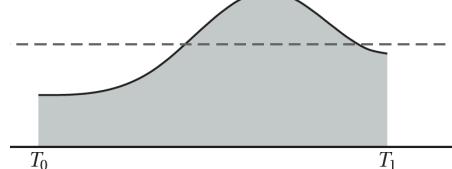
Théorème de l'échantillonage

- Le **théorème de Nyquist/Shannon** : à la base du passage continu \rightarrow discret des signaux (voir plus loin)
- **Echantillonnage (sampling)** : observer et écrire les valeurs d'une grandeur physique à des temps discrets

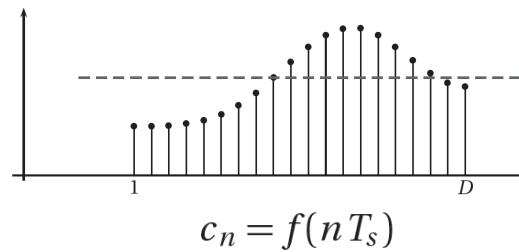
27

Temps discret vs temps continu

$$\bar{C} = \frac{1}{T_1 - T_0} \int_{T_0}^{T_1} f(t) dt$$



$$\hat{C} = \frac{1}{D} \sum_{n=1}^D c_n$$

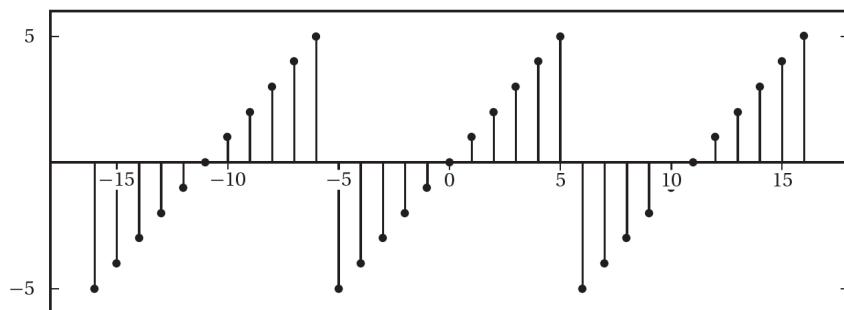


$$c_n = f(nT_s)$$

28

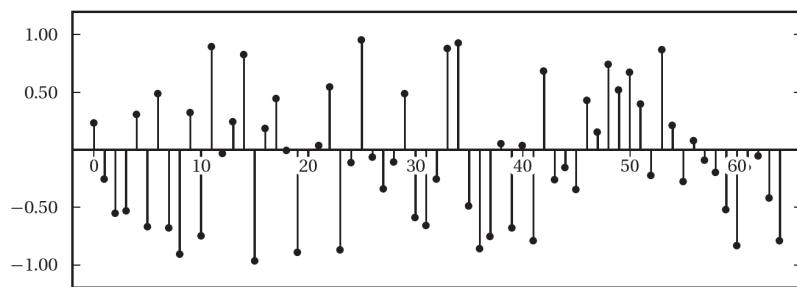
Signal à temps discret

- Définition : séquence de valeurs définie comme une fonction d'un entier d'index n , $n \in \mathbb{Z}$:
 - ▣ C'est donc une collection infinie de valeurs (n négatif et positif)
 - ▣ Exemple $x[n] = ((n+5) \bmod 11) - 5$



Signal à temps discret

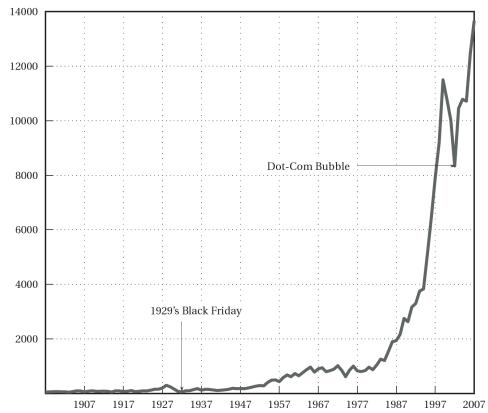
- Autre exemple
 - ▣ $x[n]$ = la n -ème sortie d'une source aléatoire $U(-1,1)$



30

Signal à temps discret

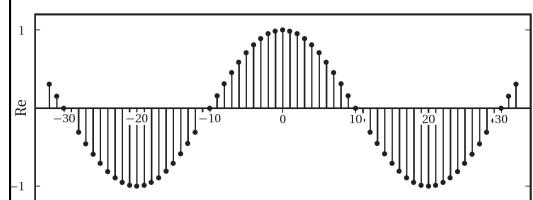
- Autre exemple
- $x[n] = \text{L'index de Dow-Jones moyen de l'année } n$



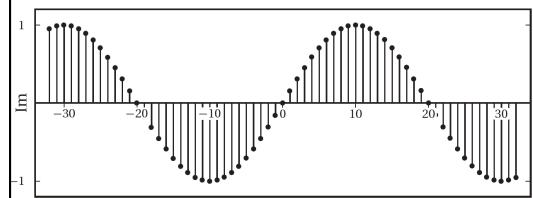
31

Signal à temps discret

- Autre exemple



$$x[n] = \cos\left(\frac{\pi}{20} \cdot n\right)$$



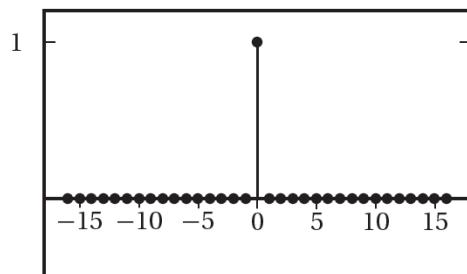
$$x[n] = \sin\left(\frac{\pi}{20} \cdot n\right)$$

32

Signaux basiques

□ Impulsion

$$\delta[n] = \begin{cases} 1 & n = 0 \\ 0 & n \neq 0 \end{cases}$$



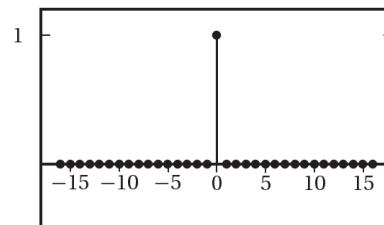
33

Exemple en Python

□ Synthèse d'une impulsion

```
import numpy as np  
  
def genSig1(N,k) :  
    r = np.zeros(N)  
    r[k]=1  
    return r
```

```
import matplotlib.pyplot as plt  
R = genSig1(20,5)  
plt.figure(1)  
plt.stem(R)
```

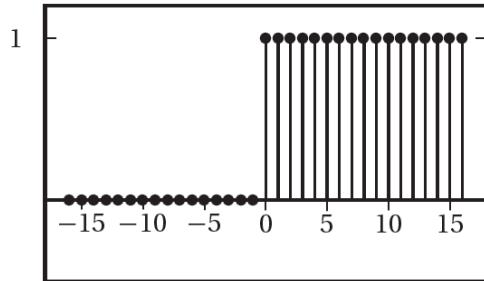


34

Signaux basiques

□ Marche unitaire

$$u[n] = \begin{cases} 1 & n \geq 0 \\ 0 & n < 0 \end{cases}$$



35

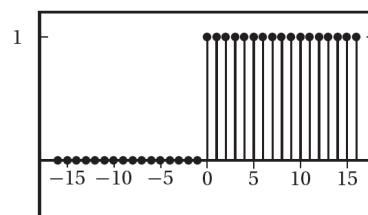
Exemple en Python

□ Synthèse d'une marche unitaire

```
import numpy as np
```

```
def genSig2(N,k):  
    r = np.zeros(N)  
    for i in range(k,N):  
        r[i]=1  
    return r
```

```
import matplotlib.pyplot as plt  
R = genSig2(20,5)  
plt.figure(1); plt.stem(R)
```

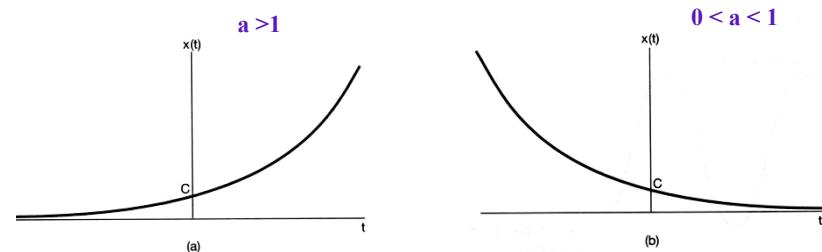


36

Signaux basiques

- Continu : signaux à exponentielle réelle

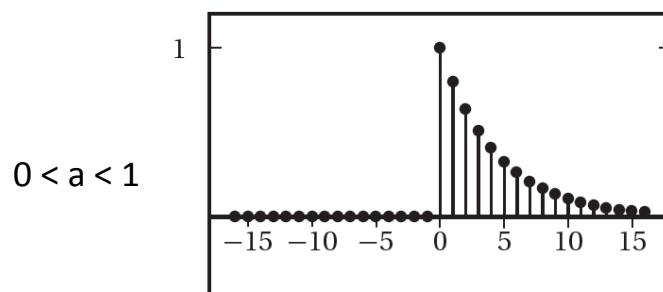
$$x(t) = Ce^{at} \quad \text{avec } C \text{ et } a \text{ réels}$$



37

Signaux basiques

- Discret : décroissance exponentielle



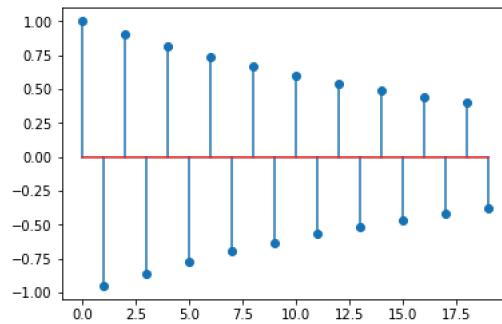
$$x(n) = a^n \cdot u(n)$$

38

Signaux basiques

- Discret : décroissance exponentielle

$$\begin{aligned} a < 0 \\ \text{abs}(a) < 1 \end{aligned}$$



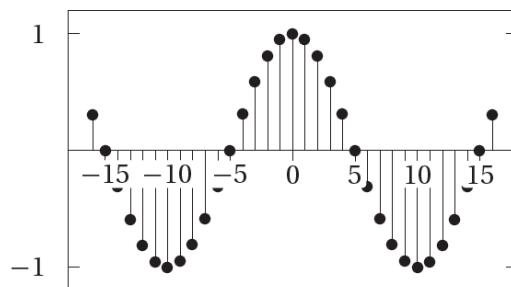
$$x(n) = a^n \cdot u(n)$$

39

Signaux basiques

- Sinusoïde (sin ou cos) : $x[n] = \cos(2\pi f_0 n)$

- f_0 = fréquence du signal
- f_s = fréquence d'échantillonnage (ici, $f_s = 20 * f_0$)

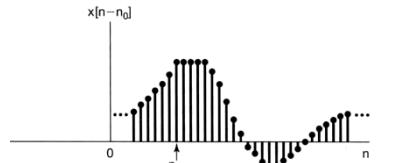
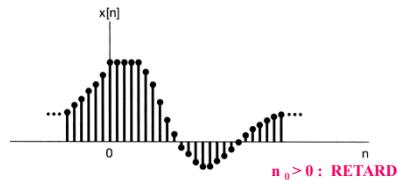
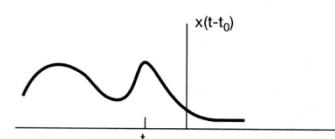
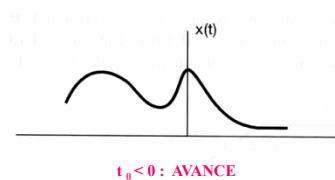


40

Principaux opérateurs

□ Décalage dans le temps : $y[n] = x[n - k]$

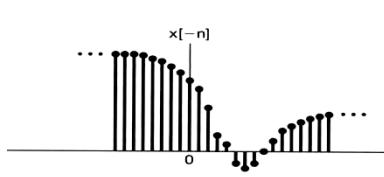
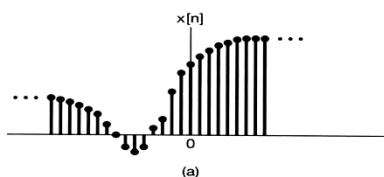
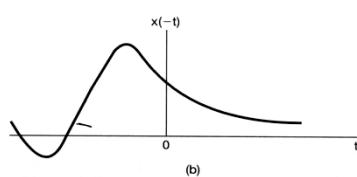
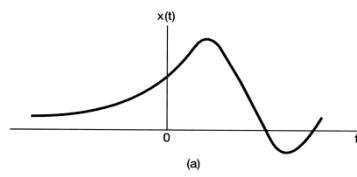
Si $n_0 > 0$, décalage à droite, si $n_0 < 0$, décalage à gauche



41

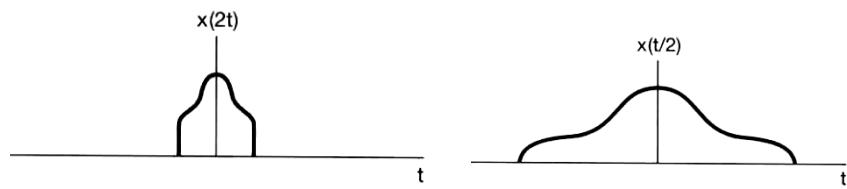
Principaux opérateurs

■ Inversion temporelle



Principaux opérateurs

- Amplification ou atténuation (scaling) : $y[n] = a \cdot x[n]$
- Changement d'échelle



43

Principaux opérateurs

- Somme de 2 séquences : $y[n] = x[n] + w[n]$
- Avec scaling : $\alpha \cdot (x[n] + w[n]) = \alpha \cdot x[n] + \alpha \cdot w[n]$
- Avec retard de k échantillons :

$$F_{\text{Retard}}(x[n] + w[n]) = x[n - k] + w[n - k]$$

44

Principaux opérateurs

- Multiplication : $y[n] = x[n] \cdot w[n]$

- Intégration

$$y[n] = \sum_{k=-\infty}^n x[k]$$

- Différentiation : approximation par une méthode aux différences finies au premier ordre :

$$y[n] = x[n] - x[n - 1]$$

45

Principaux opérateurs

- Énergie

$$E_x = \|x\|_2^2 = \sum_{n=-\infty}^{\infty} |x[n]|^2$$

- Puissance: énergie moyenne sur une période

$$P_x = \frac{1}{N} \sum_{n=0}^{N-1} |x[n]|^2$$

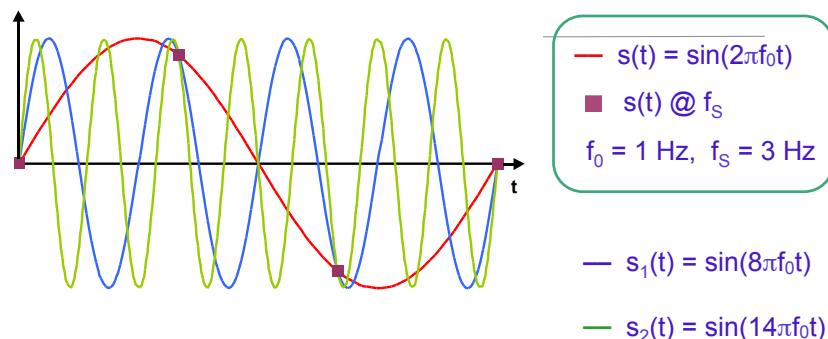
46

Échantillonnage

- Problème : à quelle vitesse doit on échantillonner un signal continu pour préserver l'information qu'il contient ?
- Ex : roues d'un train dans une vidéo
 - 25 images par seconde (fréquence d'échantillonnage)
 - Le train démarre : les roues tournent dans le sens des aiguilles d'une montre
 - Le train accélère : les roues tournent dans le sens inverse !

POURQUOI ?

Échantillonnage



Théorème de l'échantillonage ou de Nyquist-Shannon

- Le **théorème de Nyquist-Shannon** énonce que pour représenter correctement un signal numérisé, la fréquence d'échantillonnage d'un signal doit être égale ou supérieure au double de la fréquence maximale contenue dans ce signal, afin de convertir ce signal d'une forme continue à une forme discrète.
- Ce théorème est à la base du passage continu -> discret des signaux

49

Théorème de l'échantillonnage

Theo* Un signal $s(t)$ de fréquence max f_{MAX} peut être reconstitué s'il est échantillonné avec $f_s > 2 f_{MAX}$.

* Des auteurs multiples: Whittaker(s), Nyquist, Shannon, Kotel'nikov.

Exemple

$$s(t) = 3 \cdot \underbrace{\cos(50\pi t)}_{F_1} + 10 \cdot \underbrace{\sin(300\pi t)}_{F_2} - \underbrace{\cos(100\pi t)}_{F_3}$$

Condition sur f_s ?

$$F_1 = 25 \text{ Hz}, F_2 = 150 \text{ Hz}, F_3 = 50 \text{ Hz}$$

50

Théorème de l'échantillonnage

Theo* Un signal $s(t)$ de fréquence max f_{MAX} peut être reconstruit s'il est échantillé avec $f_s > 2 f_{MAX}$.

* Des auteurs multiples: Whittaker(s), Nyquist, Shannon, Kotelnikov.

Exemple

$$s(t) = \underbrace{3 \cdot \cos(50\pi t)}_{F_1} + \underbrace{10 \cdot \sin(300\pi t)}_{F_2} - \underbrace{\cos(100\pi t)}_{F_3}$$

$$F_1 = 25 \text{ Hz}, F_2 = 150 \text{ Hz}, F_3 = 50 \text{ Hz}$$

f_{MAX}

Condition sur f_s ?

$$f_s > 300 \text{ Hz}$$

51

Échantillonnage : exemple

□ Exemples

- CD audio échantillonnés : ?
- Téléphone :
- Systèmes HiFi : piètre qualité pour les animaux domestiques (les chats ou chiens entendent des ultra-sons à une fréquence d'environ 30 000 Hz !)
- Vision : limitée par le nombre de cellules visuelles dans la rétine

Échantillonnage

□ Point de vue perceptuel

- Perte d'information importante ? NON, car nos sens n'ont qu'une **précision limitée** et agissent comme des **filtres passe-bas**.
- Exemple : on ne peut pas percevoir des sons au delà de 20 000 Hz (limite inférieure des ultra-sons) -> pour l'oreille humaine, il suffit pour reconstruire des sons d'échantillonner le signal au-delà de 40 000 Hz

Traitement discret vs. continu

Traitement numérique (TNI)

Avantages

- Plus flexible.
- Souvent plus facile à mettre à jour.
- Données facilement enregistrées et stockées.
- Meilleurs contrôle des performances.
- Reproductibilité.

Limitations

- Vitesse des CA/D & processeurs de T.S. : les signaux à larges bandes spectrales toujours difficiles à traiter (systèmes temps-réels).
- Effet de la longueur finie des représentations digitales.
- Obsolescence (électronique analogique vieillit moins vite).