

FILTRAGE NUMÉRIQUE

Niveau concerné:	BTS Systèmes Numériques
Objectifs pédagogiques:	<ul style="list-style-type: none"> - Étudier les filtres numériques: récursifs ou non, stabilité. - Modéliser une sinusoïde échantillonnée. - Mettre en place un filtrage numérique.
Compétences mobilisées:	<p><i>Analyser/Raisonner:</i></p> <ul style="list-style-type: none"> - Choisir un modèle ou des lois pertinentes <p><i>Réaliser:</i></p> <ul style="list-style-type: none"> - Mettre en œuvre les étapes d'une démarche. - Effectuer des procédures courantes (calculs). <p><i>Valider:</i></p> <ul style="list-style-type: none"> - Confronter un modèle à des résultats expérimentaux. - Procéder à des tests de vraisemblances. <p><i>Communiquer:</i></p> <ul style="list-style-type: none"> - Présenter oralement une démarche de manière argumentée, synthétique et cohérente. - Rédiger un compte-rendu en utilisant des outils numériques adaptés. <p><i>Compétences numériques (CRCN):</i></p> <ul style="list-style-type: none"> - Adopter une pensée algorithmique et informatique.
Outils numériques utilisés¹:	<ul style="list-style-type: none"> - Python 3.6.7 https://www.python.org/downloads/ - Environnement de développement Geany https://www.geany.org/Download/Releases
Fichiers fournis:	<ul style="list-style-type: none"> - « filtre1.py »: fichier à fournir aux élèves. - « filtre2.py »: fichier à fournir aux élèves. - « filtre_corrige.py »: fichier corrigé. - « son.wav »: fichier audio à traiter. - « fiche_reponse_eleve.odt »: fichier réponse.

¹Tous les logiciels employés sont des logiciels Open Source, gratuits, librement téléchargeables et installables sur toutes les plateformes (Windows, mac OS ou Linux).

<p>Contexte pédagogique et déroulement de la séance:</p>	<p>Durée de la séance 2h. En demi-groupe. Première séance d'introduction sur le chapitre concernant le traitement numérique du signal. Aucune notion sur les filtres numériques n'a été vue pour le moment. Chaque élève travaille seul sur un ordinateur. Ils récupèrent le dossier de travail qui contient tous les documents (programmes à compléter, sujet, document réponse), ils le renomment avec leur nom et tous les fichiers seront enregistrés dans ce dossier (ramassé à la fin de la séance). La durée de 2h semble adaptée pour cette activité. Les plus rapides ont pu finir sans problèmes. Les étudiants connaissent très peu le langage python, mais comme il s'agit d'informaticiens, ils n'ont pas de difficultés à s'adapter. Au contraire, ils sont plutôt intéressés par le fait de pratiquer un autre langage que celui qu'ils utilisent tous les jours dans le cadre de leur formation (le C++).</p> <p>Profil de la classe: Étudiants de deuxième année de BTS, environ 50% sont originaires de bac pro SN (Systèmes Numériques), 40% de STI2D et 10% de TS. Ils bénéficient d'un enseignement de physique-chimie (anciennement physique appliquée) à hauteur de 2h de cours et 2h de TP par semaine. Il ne s'agit pas d'une matière à fort coefficient pour l'obtention du BTS et la plupart des étudiants présentent des difficultés importantes dans la matière. Les difficultés se concentrent principalement sur le passage à l'écrit et sur le calcul, qui sont deux notions qui peuvent bloquer complètement certains étudiants.</p> <p>Pré-requis des élèves: En première année, ils ont étudié l'échantillonnage et la chaîne de numérisation du signal ainsi que les filtres analogiques et les notions de fréquence d'échantillonnage, période d'échantillonnage, filtrage analogique (filtre passe-haut, passe-bas, fréquence de coupure, diagramme de Bode...etc.).</p>
<p>Retours d'expérience, améliorations et développements envisagés:</p>	<p><i>L'utilisation du codage a été une source de motivation importante pour tous les élèves. La simplicité de la syntaxe python, la possibilité de visualiser les graphiques de manière simple permet aux élèves de se concentrer sur le cœur du problème: la programmation de l'équation de récurrence. Les savoirs semblent bien avoir été renforcés après cette activité, même pour les élèves les plus en difficultés.</i></p> <p>Les difficultés: Les manipulations et la programmation ne posent pas vraiment de problèmes, avec un peu d'aide tous finissent par y arriver. La difficulté principale de l'activité est la question Q4 concernant l'échantillonnage, aucun étudiant n'a réussi à répondre à la question sans aide. La plus grosse difficulté, qui est récurrente sur toutes les activités tout au long de l'année, est le passage à l'écrit à la rédaction du compte-rendu.</p> <p>L'évaluation: Avec la grille de compétence initiale, la moyenne est de 10,5/20 avec des notes allant de 6 à 18. Ce qui, bien que parfaitement cohérent avec toutes les autres évaluations de la classe, me paraît un peu faible pour un TP qui a globalement bien été réussi (pour l'aspect pratique). Ce qui m'amène à penser à réexaminer la grille de compétences afin de ne pas trop pénaliser le passage à l'écrit.</p> <p>Amélioration envisagée: Ajouter une façon d'évaluer l'oral en plus de l'écrit (demander certaines réponses à l'oral) afin de moins pénaliser les élèves qui ont des difficultés à l'écrit (même pour un compte rendu informatique des difficultés subsistent pour certains).</p> <p>Il existe peut-être une solution plus simple pour afficher le diagramme de bode, d'après la documentation (non testé à ce jour):</p> <pre>scipy.signal.bode(system, w=None, n=100)[source]</pre> <p><i>Calculate Bode magnitude and phase data of a continuous-time system.</i></p>

	<p>Apport du codage: Concernant les enseignements et les acquis des élèves concernant les notions sur le filtrage numérique, un petit questionnaire sera distribué aux étudiants au début de la séance suivante afin de voir s'ils ont retenu les notions découvertes lors de cette activité. Ce qui est certain, c'est que l'utilisation du codage a été une source de motivation importante pour tous les élèves, ce qui est déjà un bon point. La simplicité de la syntaxe python, la possibilité de visualiser les graphiques de manière simple permet aux élèves de se concentrer sur le cœur du problème: la programmation de l'équation de récurrence. Pour avoir testé différentes activités sur ce chapitre les années précédentes, la plupart mettant en jeu l'utilisation d'un tableur, il y a indéniablement un net avantage à utiliser le langage python. Le tableur, moins attractif pour les élèves semble plus compliqué et leur parle beaucoup moins que la programmation en python. D'autre part on peut faire beaucoup plus de choses, tout en étant très proche des nombreux programmes utilisant des données numérisés (l'accroche avec la musique assisté par ordinateur prend également tout son sens et parle bien à certains élèves qui ont déjà été amenés à manipuler ce genre de logiciels).</p>
--	---

FILTRAGE NUMÉRIQUE

ÉVALUATION					
ANALYSER/RAISONNER (ANA)		A	B	C	D
Q6	Choisir un modèle ou des lois pertinentes.				
RÉALISER (RÉA)		A	B	C	D
Q2	Mettre en œuvre les étapes d'une démarche.				
Q3, Q4					
Q1, Q7	Effectuer des procédures courantes (calculs).				
VALIDER (VAL)		A	B	C	D
Q2, Q3	Confronter un modèle à des résultats expérimentaux				
COMPÉTENCES NUMÉRIQUES (NUM)		A	B	C	D
Q3, Q5	Adopter une pensée algorithmique et informatique.				
COMMUNIQUER (COMM)		A	B	C	D
	Présenter oralement une démarche de manière argumentée, synthétique et cohérente.				
	Rédiger un compte-rendu en utilisant des outils numériques adaptés.				

NOM:

Prénom:

FILTRAGE NUMÉRIQUE

1 Mise en situation et documents utiles



Filtrage audio et MAO (Musique assistée par Ordinateur)

Dans le cadre du traitement numérique du son, les logiciels de MAO (Musique Assistée par Ordinateur) mettent en jeu de nombreux filtres numériques.



Observer la vidéo à l'adresse:

https://drive.google.com/open?id=1qS5IbGBtYQkB1WbRrLmo_B7Pr_UecK1I^a

Le but de cette activité est de mettre en place différents filtres de ce type et de comprendre comment ils fonctionnent.

^aTitre: Pode Ser; Auteur: Andreia Dacal; Source: <https://www.facebook.com/andreiadacalmusica/> ; Licence: <http://creativecommons.org/licenses/by-nc-sa/2.0/fr/>



Doc. 1: Filtre numérique et équation de récurrence

Le filtrage numérique est au cœur de la chaîne de traitement numérique. Le signal électrique, numérisé, a été transformé en une suite de nombres qu'on peut traiter grâce à un algorithme (équation de récurrence) produisant ainsi des échantillons de sortie différents. On peut représenter le filtre numérique de cette façon:



Les échantillons de sortie y_n sont obtenus grâce à l'équation de récurrence qui peut s'écrire de manière générale:

$$y_n = a_1 \cdot y_{n-1} + a_2 \cdot y_{n-2} + a_3 \cdot y_{n-3} + \dots + b_0 \cdot x_n + b_1 \cdot x_{n-1} + b_2 \cdot x_{n-2} + \dots$$

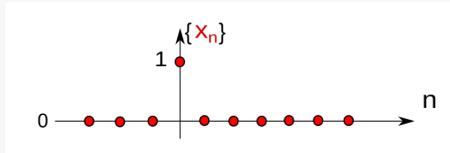
Les coefficients $a_1, a_2, a_3, \dots, b_0, b_1, b_2, \dots$ sont caractéristiques de chaque filtre numérique.



Doc. 2: Définitions et propriétés

- **Réponse impulsionnelle:**

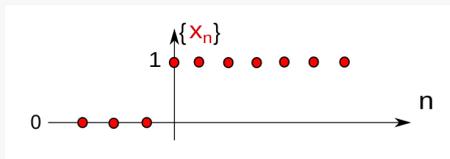
C'est la réponse à une impulsion en entrée:



$$x_n = \{1, 0, 0, 0, 0, 0, 0, \dots\}$$

- **Réponse indicielle:**

C'est la réponse à un échelon en entrée:



$$x_n = \{1, 1, 1, 1, 1, 1, 1, \dots\}$$

- Un filtre numérique est stable si sa réponse impulsionnelle tend vers zéro à l'infini.
- Les filtres **non-récurrents** ne tiennent pas compte des échantillons de sortie précédents (y_{n-1}, y_{n-2}, \dots) pour le calcul de y_n .
- Les filtres **récurrents** prennent en compte les échantillons de sortie précédents (y_{n-1}, y_{n-2}, \dots) pour le calcul de y_n .

2 Quelques exemples

- On considère le filtre dont l'équation de récurrence est la suivante:

$$y_n = 0,9 \times y_{n-1} + 0,1 \times x_n$$

Q1: Est-ce un filtre récurrent ou non-récurrent? Justifier. Compléter le tableau ci-dessous, en effectuant les calculs à la main.

n	-1	0	1	2	3
x_n impulsion					
y_n impulsion	0				

Q2: Compléter le programme "filtre1.py" afin d'afficher les réponses impulsionnelles et indicielles de ce filtre. Comparer avec les résultats précédents. Ce filtre est-il stable?

APPEL n ° 1: VÉRIFICATION DES RÉPONSES IMPULSIONNELLES ET INDICIELLES.

- On considère maintenant le filtre dont l'équation est la suivante:

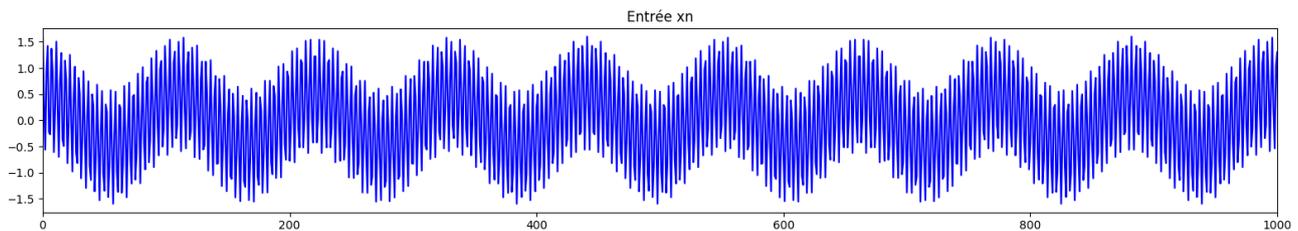
$$y_n = 0,9 \times y_{n-1} + 0,2 \times y_{n-2} + 0,2 \times x_n + 0,1 \times x_{n-1}$$

Q3: Mêmes questions que précédemment.

APPEL n ° 2: VÉRIFICATION DES RÉPONSES IMPULSIONNELLES ET INDICIELLES.

3 Filtrage d'un signal bruité

On dispose d'un signal sinusoïdal fortement bruité par une fréquence harmonique $f_2 = 30 \times f$:



On souhaite mettre en place un filtrage numérique afin de supprimer cette fréquence supérieure parasite.

Pour générer le signal bruité on additionne deux fonctions cosinus

$$x(t) = 0,5 \times \cos(2\pi f.t) + 1,1 \times \cos(30 \times 2\pi f.t)$$

Q4: Expliquer ce que devient le paramètre "t" avec la notion d'échantillonnage (faire intervenir les paramètres "n" et "Te"). Compléter la ligne du programme "filtre2.py" afin d'obtenir le signal bruité échantillonné $x[n]$.

Q5: Compléter la ligne 62 permettant de calculer les échantillons de sortie y_n

Tester le programme. Écouter les sons filtré et non-filtré générés dans le dossier de travail.

Q6: On souhaite éliminer la fréquence parasite de telle sorte à ce qu'elle soit atténuée d'au moins 30 dB. Parmi ceux proposés, quel filtre faut-il choisir?

Q7: Déterminer la (ou les) fréquence(s) de coupure de ce filtre.

APPEL n ° 3: VÉRIFICATION SIGNAL FILTRÉ ET DE LA FRÉQUENCE DE COUPURE.

4 Filtrage audio

Appliquer les filtres proposés à l'extrait audio à votre disposition en utilisant la fonction `wavf.read()`

```
fe, x = wavf.read('son.wav')
```

Écouter le résultat après avoir remplacé la partie qui génère la sinusoïde bruitée par la ligne ci-dessus.

APPEL n ° 4: VÉRIFICATION DU FILTRAGE DU SIGNAL AUDIO.