

Outils Numériques pour l'Ingénieur·e en Physique

2023-2024

5N-028-PHY / ONIP-1

Bloc 3 - Modulation AM (50%)

Concepts étudiés

- [PHYS] Modulation d'amplitude
- [MATH] Transformée de Fourier
- [NUM] Signaux numériques
- [NUM] Figures scientifiques

Mots clefs

Fichier CSV ; Graphique scientifique ;
Transformée de Fourier ; Modulation
d'Amplitude ; Démodulation

Sessions

- 0 Cours(s) - 1h30
- 0 TD(s) - 1h30
- 4 TD(s) Machine - 2h00
- 0 TP(s) - 4h30

Travail

Seul ou par binôme

Institut d'Optique
Graduate School, France
<https://www.institutoptique.fr>

GitHub - Digital Methods

<https://github.com/IOGS-Digital-Methods>

Afficher et traiter des données provenant d'instruments de mesure

Les **expériences scientifiques**, les **essais industriels** sur des systèmes ou bien encore des **résultats de simulation** produisent énormément de **données**. Ces données sont souvent sauvegardées sous forme de **fichiers formatés** (format normalisé ou interne aux entreprises/laboratoires).

Il est alors indispensable de pouvoir **afficher les données** contenues dans ce type de fichier de manière claire et sans ambiguïté, avant d'en **extraire des informations pertinentes** par un traitement adapté.

Vous traiterez dans cette séquence une **information modulée en amplitude**, acquise par un **oscilloscope numérique** et stockée dans un **fichier de type tableur**.

Acquis d'Apprentissage Visés

En résolvant ce problème, les étudiant·e·s seront capables de :

CÔTÉ NUMÉRIQUE

1. **Générer des signaux numériques** à partir de fonctions mathématiques
2. **Définir et documenter des fonctions** pour générer des signaux numériques
3. **Produire des figures** claires et légendées à partir de signaux numériques - incluant un titre, des axes, des légendes
4. [BONUS] **Construire des bibliothèques de fonctions**

CÔTÉ PHYSIQUE

1. **Analyser le contenu spectral** d'un signal électrique
2. **Déterminer les paramètres** d'une modulation d'amplitude
3. **Décoder** un signal modulé en amplitude

Livrables attendus

Pour valider cette session, vous devez fournir les **livrables suivants** :

1. **Fonctions commentées** (selon la norme PEP 257) pour générer des signaux numériques appropriés
2. **Graphiques légendés** incluant toutes les données nécessaires à la bonne compréhension des données présentées : signal initial, transformée de Fourier du signal initial, signaux générés pour démoduler le signal, transformées de Fourier intermédiaires, signal démodulé
3. **Analyse des figures** en insistant sur la démarche ayant amené à la démodulation du signal
4. **BONUS : Fichiers démodulés** contenant les différents signaux démodulés

Ces livrables pourront prendre la forme d'un **compte-rendu** incluant une introduction à la problématique, les figures demandées ainsi que leur analyse.

Ce compte-rendu sera accompagné des **fichiers** *main.py* et *signal_processing.py* contenant le programme principal permettant la génération des figures et de leurs légendes et les différentes fonctions commentées selon la norme PEP 257.

Données à traiter

Dans cette séquence, vous serez amenés à utiliser des données provenant d'un fichier de points issu d'un **oscilloscope**. Le fichier se nomme B3_DATA_01.CSV.

Le signal qu'il contient est un enregistrement d'une **transmission d'informations modulées en amplitude** par un signal porteur sinusoïdal.

Deux autres fichiers vous sont également proposés :

- B3_DATA_02.CSV contenant un signal sonore modulé en amplitude à déchiffrer...
- B3_DATA_03.CSV contenant un ensemble de signaux modulés en amplitude à l'aide de différentes porteuses.

Ressources

Cette séquence est basée sur le langage Python.

Vous pouvez utiliser l'environnement **Spyder 5** inclus dans *Anaconda 3*.

Des tutoriels Python (et sur les bibliothèques classiques : Numpy, Matplotlib or Scipy) sont disponibles à l'adresse : <http://lense.institutoptique.fr/python/>.

Outils Numériques

Fonctions et bibliothèques conseillées :

- **Numpy** gestion de matrices
 - `arange`
 - `linspace`
 - `logspace`
- **Matplotlib** affichage de données :
 - `plotly`
 - `figure, plot`
 - `subplot`
 - `legend, title`
 - `xlabel, ylabel`
 - `show`
- **Scipy** fonctions scientifiques :
 - `fftpack` sublibrary
 - `fft, ifft`
 - `fftshift`
 - `fftfreq`

Outils avancés :

- **rcParams** de Matplotlib pour l'amélioration de l'affichage de courbes