60 Interférences sonores : simulateur Python

Programme à compléter

L'objectif de cet exercice est de simuler les signaux sonores reçus par un microphone M placé en un point du segment reliant deux haut-parleurs H1 et H2 placés face à face. La situation est décrite dans l'exercice **59** p. 482.

Fichiers Python
Programme à compléter
Fiche d'accompagnement
hatier-clic.fr/pct482

Prérequis théoriques

- Interférences
- Différence de marche
- Conditions d'interférences constructives et destructives

Le programme est à modifier et à compléter à la question **a** avec :
– une ligne de commande permettant d'entrer la valeur de la distance x qui sépare le microphone du premier haut-parleur H_1 ;
– une ligne de calcul de la valeur de la différence de marche δ.
Le programme trace ensuite les courbes représentant les signaux issus de H_1, de H_2 ainsi que leur superposition.

À la question **b**, on utilise le programme afin de vérifier, par analyse du graphique du signal de superposition, la nature des interférences en M pour les différentes valeurs de x de la question **d** de l'exercice **59** : x = 86,25 cm ; x = 63,5 cm ; x = 107 cm.

Programme à compléter

```
1  from math import *
2  import matplotlib.pyplot as plt
3  import numpy as np
4
5  print("")
6  print("**************************************")
7  print("*Interférences sonores                *")
8  print("*   Exercice 60 p. 482                *")
9  print("************************Hatier 2020**")
10
11 print("")
12 print("Attention : le séparateur décimal est le point")
13 print("")
14
15 # Saisie des paramètres des signaux
16 freq=float(input('fréquence du signal (Hz) = '))
17 ampl1=float(input('amplitude du signal 1 = '))
18 ampl2=float(input('amplitude du signal 2 = '))
19 cel=float(input('célérité (m/s) = '))
20 # Début des lignes à modifier
21 delta=float(input('différence de marche (m) = '))
22 #
23 # Fin des lignes à modifier
24
25 # Calcul du déphasage
26 deph=2*pi*freq*delta/cel
27 print("déphasage = ",deph," = ",deph/(2*pi),"* 2 pi")
28
29 # Construction des listes
30 listet=np.linspace(0,2/freq,250)
31 listesign1=np.linspace(0,2/freq,250)
32 listesign2=np.linspace(0,2/freq,250)
33 listesuperpo=np.linspace(0,2/freq,250)
34
35 for i in range(0,250) :
36     listesign1[i]=ampl1*sin(2*pi*freq*listet[i])
37     listesign2[i]=ampl2*sin(2*pi*freq*listet[i]+deph)
38     listesuperpo[i]=listesign1[i]+listesign2[i]
39
40 # Tracé des graphiques
41 plt.plot(listet,listesign1,color="blue")
42 plt.plot(listet,listesign2,color="green")
43 plt.plot(listet,listesuperpo,color="red")
44 plt.title("signal 1 en bleu, 2 en vert, somme en rouge")
45 plt.show()
```

Modules importés

Importation des modules de tracé de courbe et de calcul numérique.

Paramètres initiaux

Les valeurs numériques sont à entrer dans la console python une fois le programme lancé.

À modifier : valeur de x et calcul de la différence de marche

À la question **a**, la ligne 20 est à modifier pour permettre d'entrer la valeur numérique de l'abscisse x.

La ligne 21 est à compéter pour calculer la différence de marche à partir de la valeur de x : $H_1M = x$, $H_2M = d - x = 1{,}20 - x$.

Calcul du déphasage

Cette ligne permet de calculer le déphasage entre les deux ondes sonores.

Cette valeur est affichée dans la console grâce à la commande `print`.

Création des listes

Les listes contenant les points permettant de tracer les deux signaux sonores et leur superposition sont construites grâce à une boucle `for`.

Tracé du graphique

Le programme permet ici de tracer les trois signaux : celui issu de H_1 en bleu, celui de H_2 en vert et leur superposition en rouge.