

Boussole dans un champ magnétique

1. Expérience

Ce document présente l'étude expérimentale du mouvement de rotation d'une aiguille aimantée de boussole dans un champ magnétique. L'aiguille est posée sur une pointe verticale, ce qui lui permet de tourner autour d'un axe vertical avec très peu de frottement. Le champ magnétique utilisé dans un premier temps est le champ magnétique terrestre.

On note m le moment magnétique de l'aiguille et J son moment d'inertie. Si on suppose qu'il n'y a pas de frottement, l'équation du mouvement angulaire est l'équation du pendule :

$$\frac{d^2\theta}{dt^2} + \frac{mB}{J} \sin \theta = 0 \quad (1)$$

En réalité, il y a bien sûr un frottement qui réduit progressivement l'amplitude des oscillations.

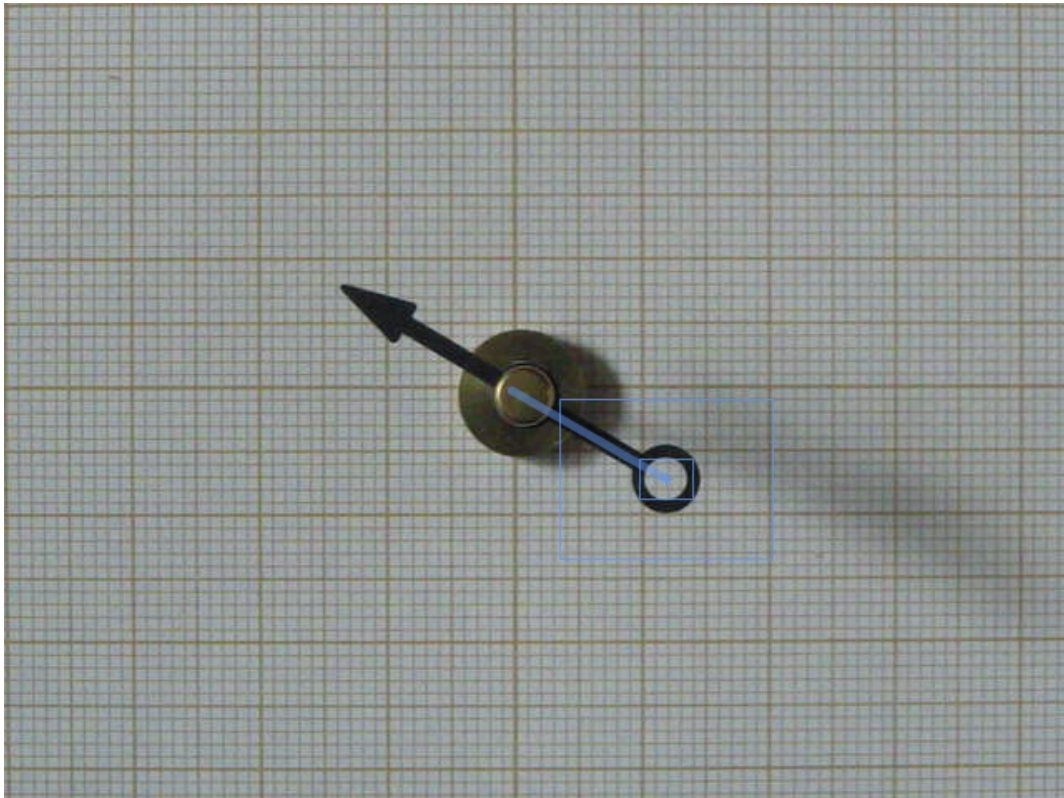
Une caméra est fixée au dessus de la boussole, son axe optique confondu avec l'axe de rotation. La vidéo ci-dessous est prise à une cadence de 120 images par secondes à la résolution de 640 par 480. Lorsque la vidéo est lue à vitesse normale, on voit le mouvement ralenti d'un facteur 4. Le champ magnétique terrestre est colinéaire à l'axe horizontal du papier millimétré, dirigé de la gauche vers la droite. Le grand bord de l'image a été aligné avec cette direction. La durée réelle de la vidéo est 79 secondes, ce qui fait une lecture de 5 minutes et 19 secondes en vitesse normale.

Votre navigateur ne gère pas l'élément video

2. Acquisition du mouvement

Le logiciel [Kinovea](#) est utilisé pour extraire de la vidéo le mouvement de la boussole.

On choisit de suivre la trajectoire de l'anneau situé sur le pôle sud de la boussole. Le premier point sélectionné sert d'origine des coordonnées. On commence donc par sélectionner le centre de la boussole pour que l'origine coïncide avec l'axe de rotation. On fait défiler deux ou trois images puis on déplace la sélection vers l'anneau. On obtient alors la représentation suivante :



Une fois l'anneau sélectionné, on lance la lecture de la vidéo en continu pour faire le suivi de la trajectoire.

Les données sont exportées sous forme de fichier texte : [CIMG5043.txt](#). Le temps est en millisecondes, les coordonnées en pixels.

3. Traitement des données

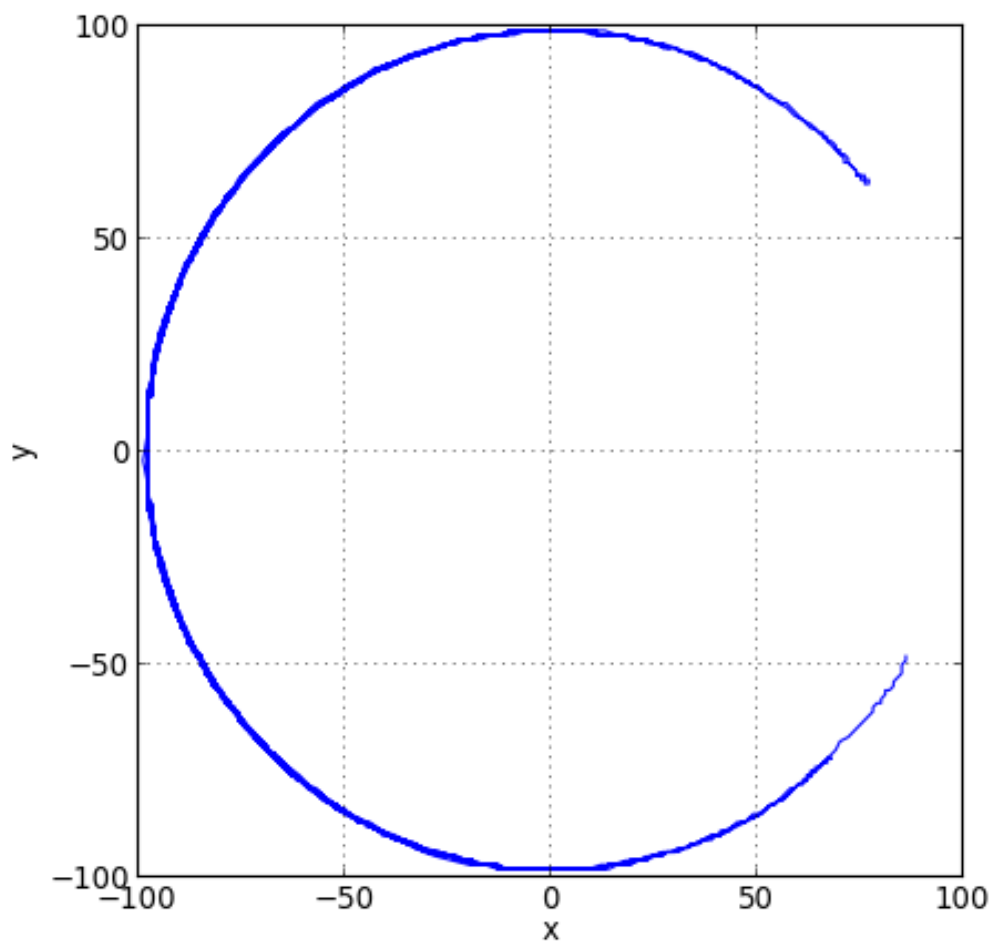
```
from matplotlib.pyplot import *
import numpy
import math
import cmath
import numpy.fft
import scipy.signal
import StringIO
```

3.a. Calcul de l'angle

On commence par lire les données dans le fichier texte. Les virgules doivent être converties en points. On saute les deux premières lignes du fichier. Pour chaque colonne, on enlève les 5 premières valeurs, qui correspondent à un marqueur situé au centre de la boussole.

```
s = open('CIMG5043.txt').read().replace(',','.')
data = numpy.loadtxt(StringIO.StringIO(s),skiprows=2,unpack=True)
debut = range(5)
t=numpy.delete(data[0],debut)
```

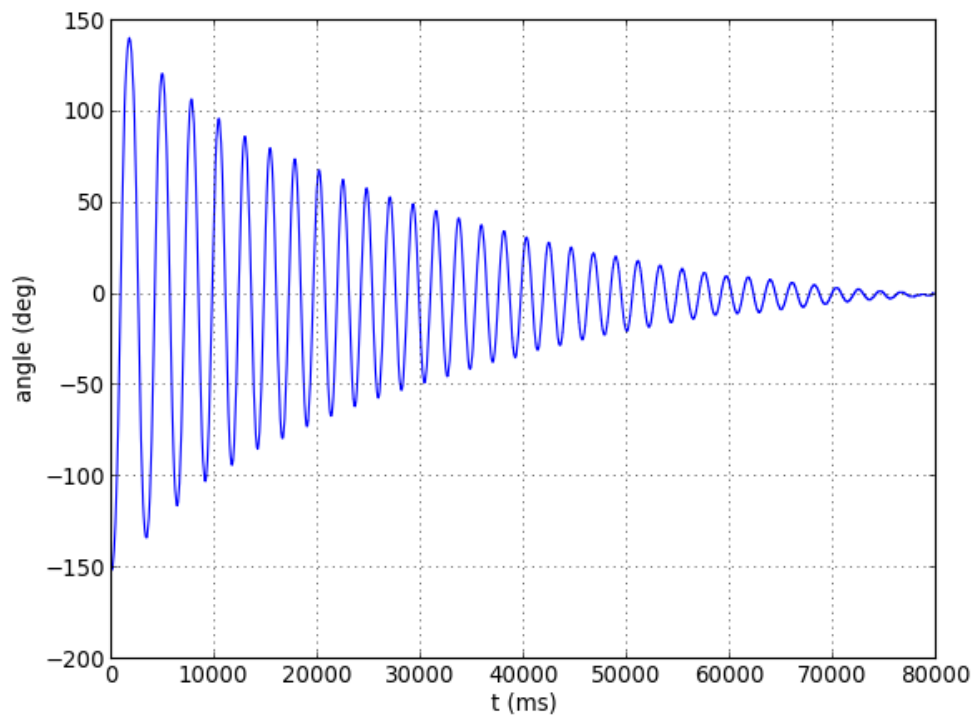
```
x=numpy.delete(data[1],debut)
y=numpy.delete(data[2],debut)
figure(figsize=(6,6))
plot(x,y)
xlabel("x")
ylabel("y")
grid()
```



Pour obtenir l'angle, le plus simple est de créer un tableau avec des nombres complexes et d'en extraire l'argument. Comme la position d'équilibre de l'anneau se trouve à gauche de l'image, on change le signe de x .

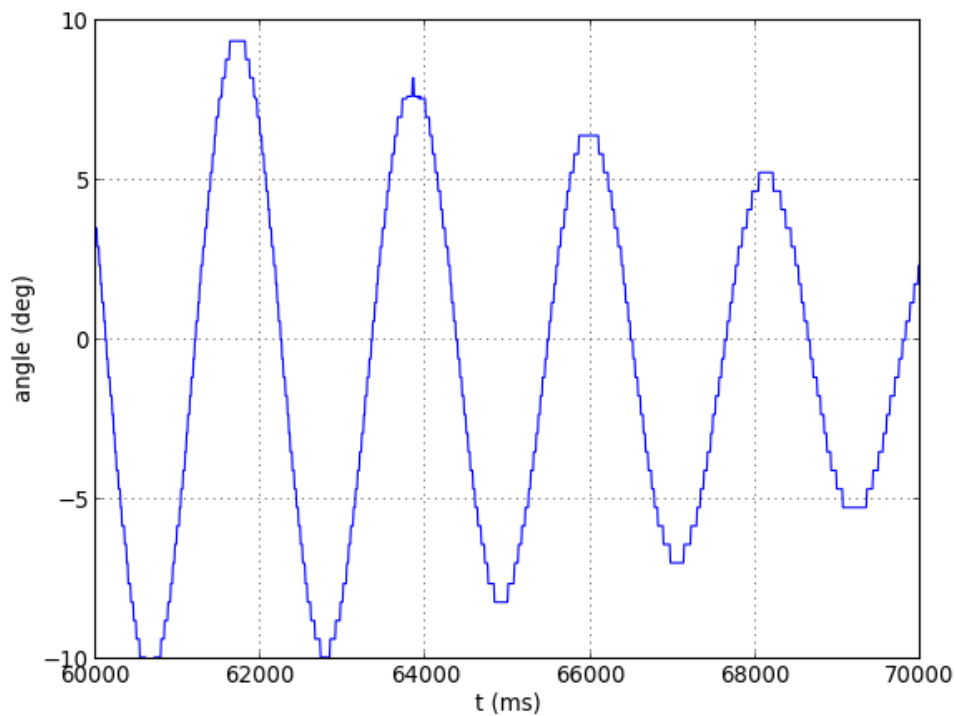
```
angle = numpy.angle(-x+1j*y)
```

```
figure()
plot(t,angle*180.0/math.pi)
xlabel("t (ms)")
ylabel("angle (deg)")
grid()
```



Voici un détail vers la fin du mouvement :

```
figure()
plot(t,angle*180.0/math.pi)
xlabel("t (ms)")
ylabel("angle (deg)")
grid()
axis([6e4,7e4,-10,10])
```



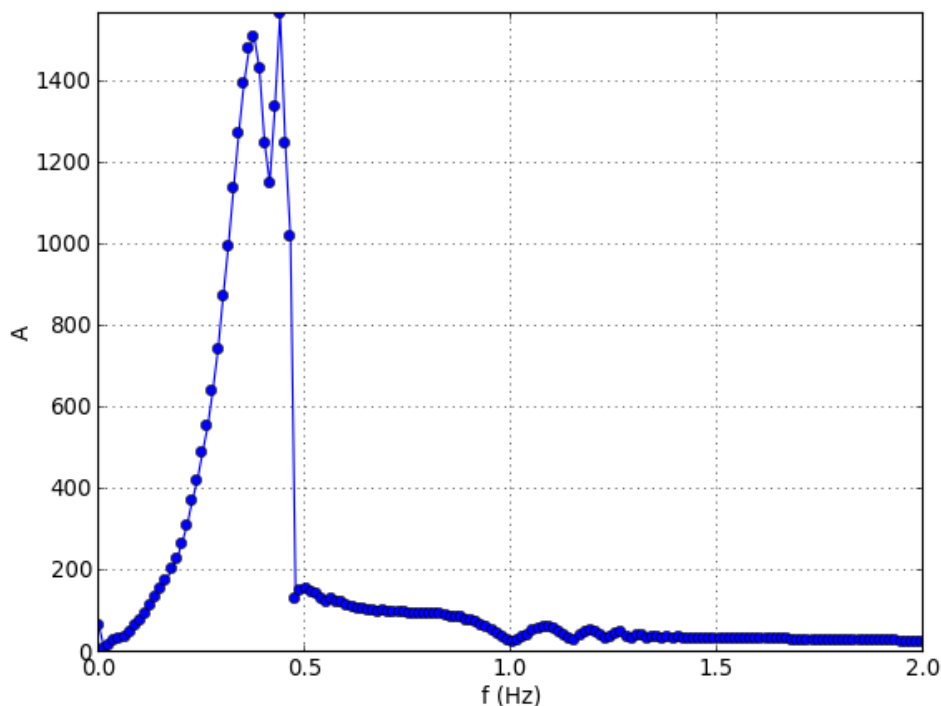
On voit sur ce tracé la quantification de l'angle, qui vient de celui des coordonnées (x, y) qui varient par saut de 1 pixel. Par rapport à la courbe idéale qui devrait être obtenue en signal analogique, la numérisation a introduit un *bruit de quantification*.

3.b. Analyse spectrale

On commence par une analyse spectrale globale, sur toute la durée de la trajectoire.

```
fe=120.0
te=1.0/fe
N=y.size
T=1.0/fe*N
f = numpy.arange(angle.size)*1.0/T
spectre = numpy.abs(numpy.fft.fft(angle))
```

```
figure()
plot(f,spectre,'-o')
xlabel('f (Hz)')
ylabel('A')
grid()
axis([0,2,0,spectre.max()])
```

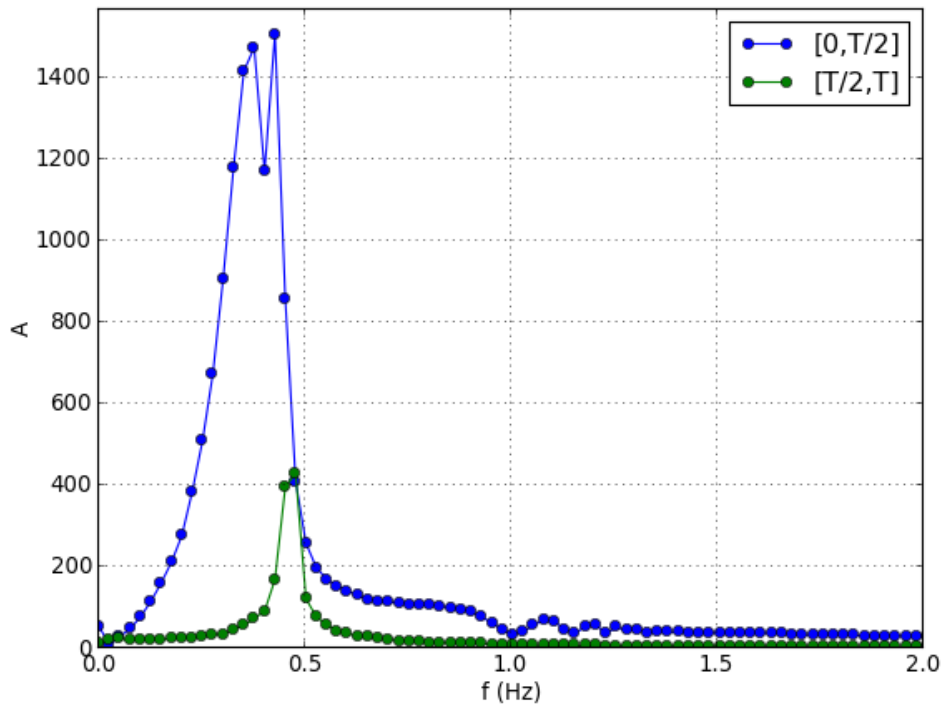


Le spectre est assez complexe car d'une part le mouvement est pseudo-périodique (périodique amorti), d'autre part la pseudo-période évolue au cours du mouvement (elle diminue lorsque l'amplitude diminue).

Pour y voir plus clair, on fait deux analyses spectrales, chacune sur une moitié de la trajectoire.

```
N1 = int(N/2)
spectre1 = numpy.abs(numpy.fft.fft(angle[0:N1]))
f1 = numpy.arange(spectre1.size)*1.0/(N1*te)
spectre2 = numpy.abs(numpy.fft.fft(angle[N1:N]))
f2 = numpy.arange(spectre2.size)*1.0/((N-N1)*te)
```

```
figure()
plot(f1,spectre1,'-o',label='[0,T/2]')
plot(f2,spectre2,'-o',label='[T/2,T]')
xlabel('f (Hz)')
ylabel('A')
grid()
legend(loc='upper right')
axis([0,2,0,spectre.max()])
```



Les spectres sont deux fois moins précis puisque la durée a été divisée par deux. Sur le spectre de la seconde moitié de la trajectoire, on voit le maximum correspondant aux oscillations de petite amplitude, pour un angle inférieur à 30 degrés environ.

On en déduit la fréquence des oscillations de petite amplitude :

$$f_0 = 0,475 \pm 0,025 \text{ Hz} \quad (2)$$

D'après l'équation du mouvement, cette fréquence est :

$$f_0 = \frac{1}{2\pi} \sqrt{\frac{mB}{J}} \quad (3)$$

Au lieu où l'expérience est réalisée, le champ magnétique terrestre peut être estimé ([Earth magnetic field](#)) à

$$B = 47,5 \pm 0,5 \text{ } \mu\text{T} \quad (4)$$

On en déduit :

$$\frac{m}{J} = (1,87 \pm 0,14) \cdot 10^5 \text{ s}^2 \cdot \text{T}^{-1} \quad (5)$$

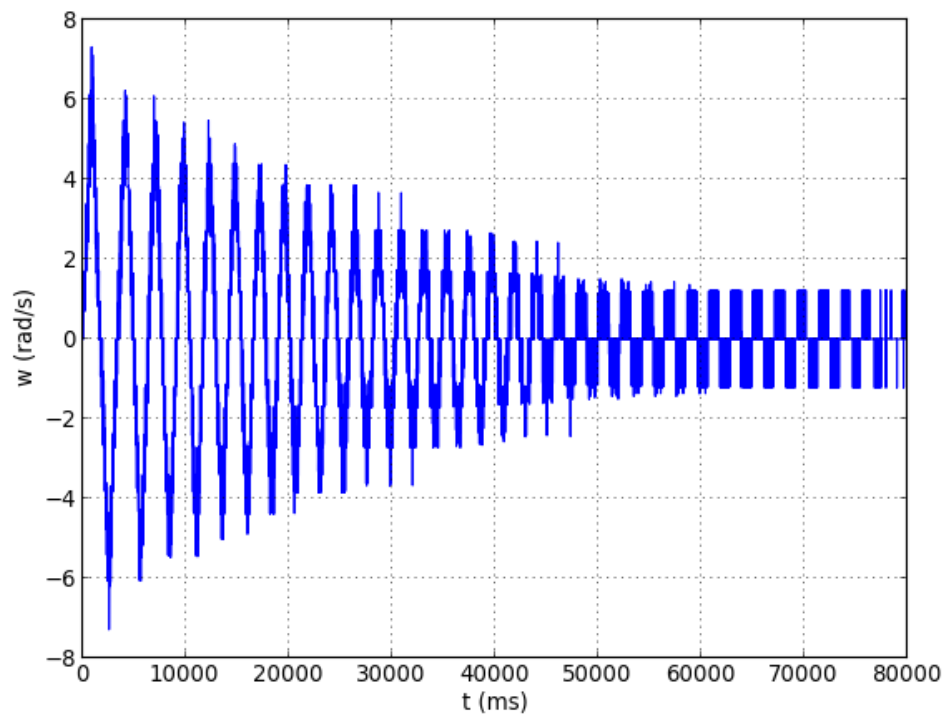
On peut alors utiliser cette valeur pour déterminer l'intensité d'un champ magnétique (par exemple celui d'un aimant), en faisant osciller cette boussole dans le champ.

3.c. Vitesse angulaire

La vitesse angulaire est obtenue avec un filtre dérivateur, c'est-à-dire un filtre RIF dont la réponse impulsionnelle est $(1, -1)$:

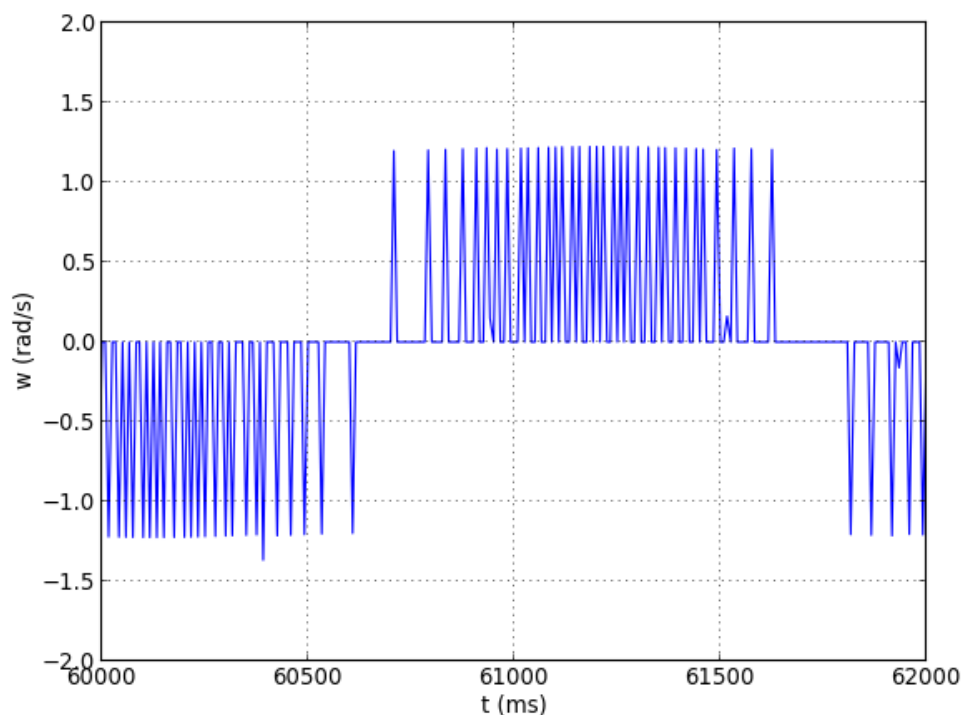
```
a=[te]
b=[1,-1]
omega = scipy.signal.lfilter(b,a,angle)
omega[0] = omega[1]
```

```
figure()
plot(t,omega)
xlabel("t (ms)")
ylabel("w (rad/s)")
grid()
```



Voyons un détail :

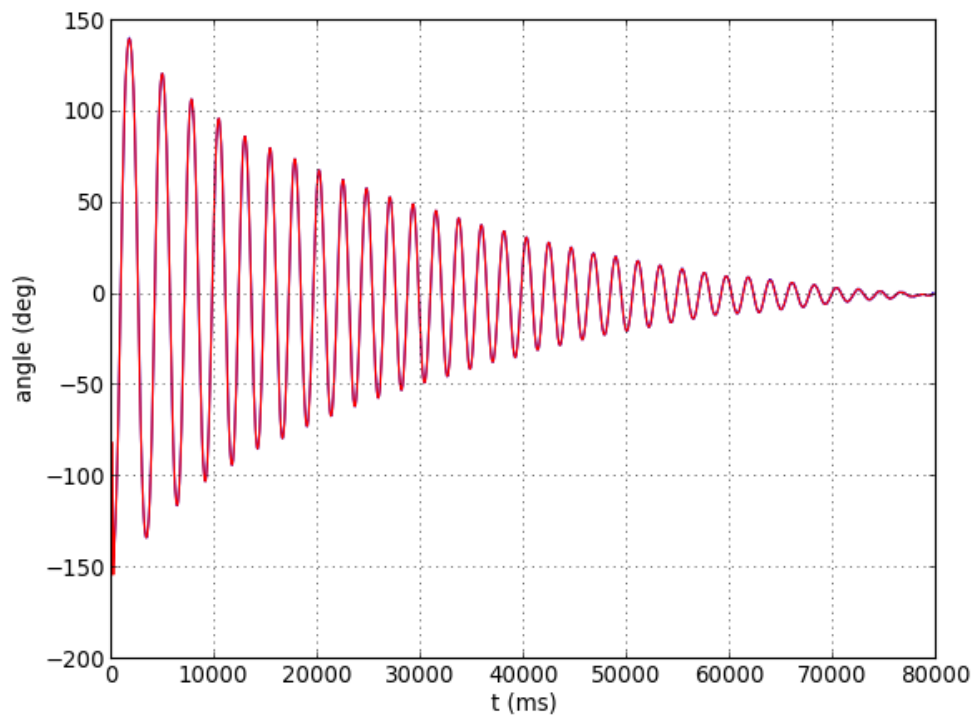
```
figure()
plot(t,omega)
xlabel("t (ms)")
ylabel("w (rad/s)")
axis([6e4,6.2e4,-2,2])
grid()
```



La vitesse n'a que trois valeurs possibles dans cet intervalle. Cela vient de la quantification de l'angle, qui varie par sauts. On voit donc que le bruit de quantification est très amplifié par la dérivation. D'une manière générale, la dérivation amplifie le bruit. Le signal obtenu ici est pratiquement inutilisable.

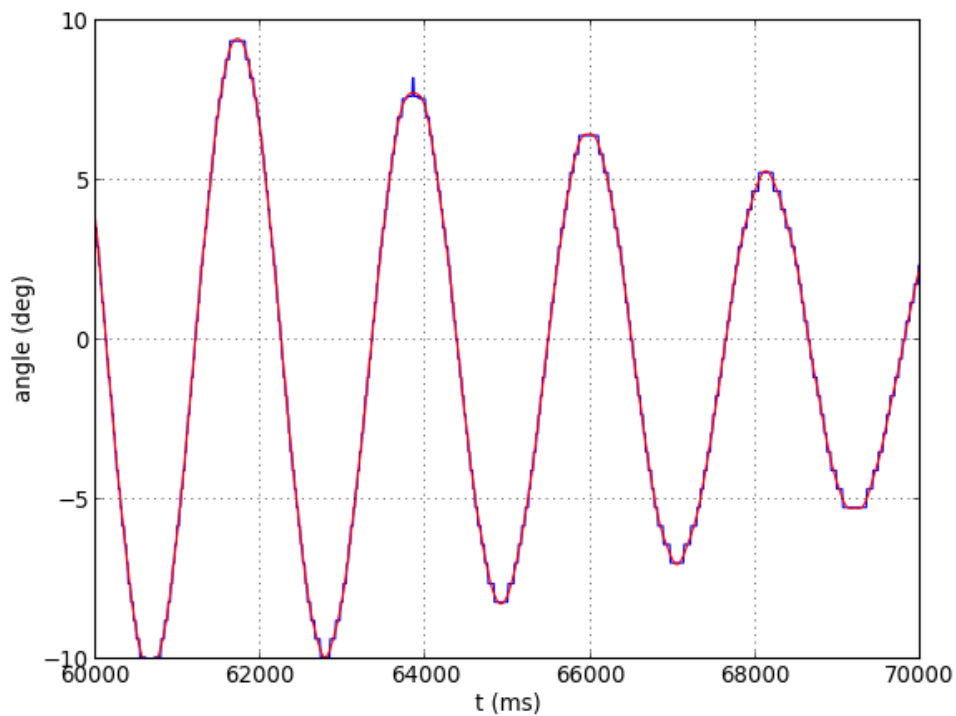
La solution est de lisser le signal avant d'effectuer la dérivation. Pour cela, on peut utiliser un filtre RIF passe-bas. Dans le cas présent, la fréquence d'échantillonnage (120 Hz) est très grande devant la fréquence du signal (environ $0,5\text{ Hz}$). On peut donc utiliser un filtre passe-bas avec un indice de troncature assez élevé, ce qui donnera un très bon filtrage. Voici le résultat avec une fréquence de coupure de 5 Hz . On superpose le signal filtré avec le signal d'origine pour vérifier que l'amplitude n'est pas modifiée :

```
P=30
fc=5.0/120
h = scipy.signal.firwin(numtaps=2*P+1,cutoff=[fc],nyq=0.5,window='hamming')
angle_filtre = scipy.signal.convolve(angle,h,mode='same')
figure()
plot(t,angle*180.0/math.pi,'b')
plot(t,angle_filtre*180.0/math.pi,'r')
xlabel("t (ms)")
ylabel("angle (deg)")
grid()
```



Voici le même détail que plus haut :

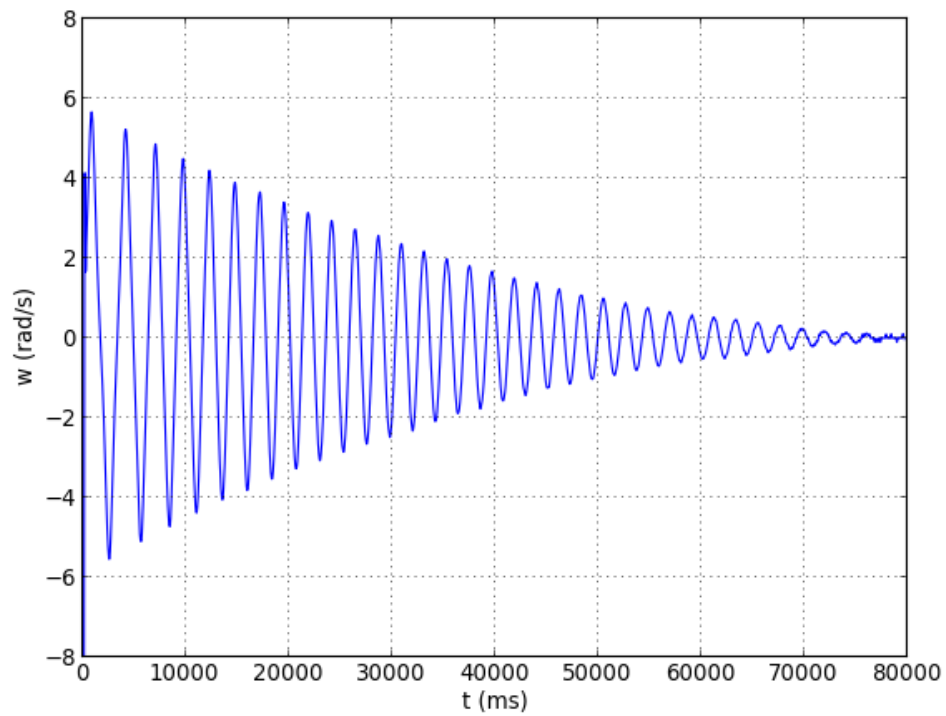
```
figure()
plot(t,angle*180.0/math.pi,'b')
plot(t,angle_filtre*180.0/math.pi,'r')
xlabel("t (ms)")
ylabel("angle (deg)")
grid()
axis([6e4,7e4,-10,10])
```



Le lissage a bien fait disparaître le bruit de quantification. On peut à présent effectuer la dérivation :

```
a=[te]
b=[1,-1]
omega_filtre = scipy.signal.lfilter(b,a,angle_filtre)

figure()
plot(t,omega_filtre)
xlabel("t (ms)")
ylabel("w (rad/s)")
axis([0,8e4,-8,8])
grid()
```



3.d. Portrait de phase

Pour finir, on trace la trajectoire dans le plan de phase.

```
figure()
debut=P
plot(angle_filtre[debut:N],omega_filtre[debut:N])
xlabel("angle (rad)")
ylabel("omega (rad/s)")
grid()
```

