

Système masse-ressort et accéléromètre

Prenez en note tout élément pouvant figurer dans un compte-rendu de TP : mesures, calculs d'incertitude, observations (schémas) et interprétations, méthodes expérimentales...

Objectifs :

- Capturer un mouvement puis l'analyser d'un point-de-vue cinématique et dynamique ;
- Mettre en œuvre un protocole expérimental permettant d'étudier une loi de force.
- Mettre en œuvre un protocole expérimental de mesure de frottements fluides.
- Vérifier expérimentalement la loi de Hooke sur un ressort.
- Étudier les oscillations libres ou forcées d'un oscillateur mécanique amorti. Caractériser un phénomène de résonance.

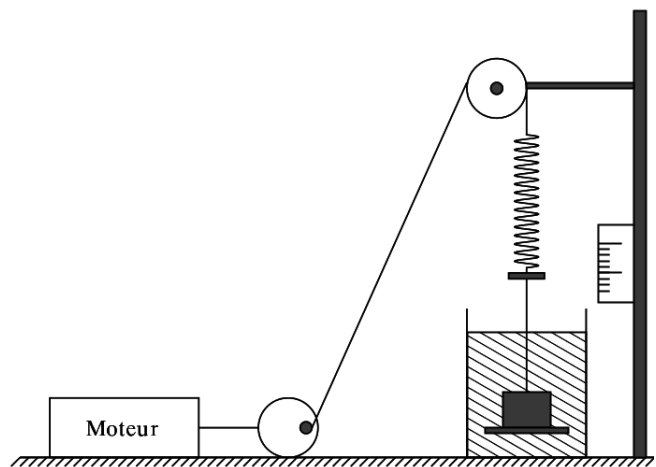
Après avoir caractérisé le ressort par sa raideur de différentes manières, on s'attache à l'étude des oscillations en régime libre, sans ou avec frottements fluides, puis en régime forcé. La capture du mouvement se fait via un accéléromètre connecté à une carte ARDUINO pilotée par PYTHON, qui permettra de mesurer la période et l'amplitude des oscillations.

Compétences expérimentales exigibles :

- Utiliser une balance de précision.
- Mettre en œuvre un accéléromètre, par exemple avec l'aide d'un microcontrôleur.

On s'intéresse au dispositif ci-contre, constitué d'un système masse-ressort vertical relié à un moteur permettant de fonctionner en Régime Sinusoïdal Forcé (RSF), c'est-à-dire d'imposer un mouvement sinusoïdal à une fréquence f fixée par le moteur.

La masse est fixée à l'extrémité d'une tige rigide, et est ajustable. On peut la plonger dans l'eau pour lui appliquer un amortissement par frottement fluide, dont on peut moduler la force en ajoutant une rondelle de diamètre variable. La tige est équipée d'un accéléromètre relié à une carte ARDUINO par une connexion flexible, permettant de capturer le mouvement. On se reportera à l'annexe pour trouver toutes les informations utiles sur son utilisation, et au précédent TP "Traction" pour ce qui concerne le pilotage de l'acquisition par PYTHON.



Avant de s'intéresser aux mouvements en régime libre (c'est-à-dire sans moteur), et en régime forcé (RSF), on réalise une étude statique (à l'équilibre).

I. Etude statique

On rappelle¹ qu'un ressort exerce une force dite *de rappel* qui obéit à la loi de HOOKE, à savoir qu'elle est proportionnelle à l'*allongement* $\ell - \ell_0$ du ressort, différence entre la longueur ℓ et la longueur à vide ℓ_0 (qu'aurait le ressort si on le détachait). On souhaite déterminer la *raideur* k du ressort, qui est le facteur de proportionnalité de cette force. Pour cela on place la masse hors de l'eau et on cherche la position d'équilibre z_{eq} .

1. cf cours de dynamique

• **MANIP 1 : Mesure de la raideur du ressort**

A l'aide d'une série de mesures pour des masses m différentes, mesurer la raideur k du ressort.

On pourra utiliser un tableur et effectuer une régression linéaire pour des variables bien choisies, pour améliorer la précision et éliminer des erreurs éventuelles.

On pourra aussi en profiter pour effectuer en même temps les mesures de la MANIP 3 ci-dessous.

On souhaite maintenant plonger la masse dans l'eau, qui est un fluide 1000 fois plus dense que l'air. En plus de la force de frottement fluide, l'eau va exercer une poussée d'Archimède sur la masse immergée. On souhaite savoir si cela doit être pris en compte sous la forme d'une pesanteur effective ou si c'est négligeable.

• **MANIP 2 : Immersion de la masse - poussée d'Archimède**

- En contrôlant l'évolution de la position d'équilibre (sans eau puis avec eau), déterminer si la force d'Archimède est perceptible.
- En contrôlant l'évolution du niveau d'eau avant et après immersion, déterminer le volume de la partie immergée.
- Connaissant la masse volumique de l'eau $\rho \approx 1000 \text{ kg}\cdot\text{m}^{-3}$, peut-on conclure que le principe d'Archimède est vérifié ?

À partir de maintenant, l'étude des mouvements de la masse se fera prioritairement par le biais du signal de l'accéléromètre selon l'axe du mouvement.

II. Régime libre

II.1. Dans l'air

On place de nouveau la masse en dehors de l'eau. On souhaite savoir si l'on peut négliger la masse propre du ressort dans la modélisation théorique du mouvement.

Q1. Quel est l'ordre de grandeur des accélérations maximales à mesurer ? On pourra s'aider d'une mesure rapide de période et d'amplitude. En déduire le choix de la gamme de mesure de l'accéléromètre, sachant que l'on doit prendre en compte aussi la pesanteur.

• **MANIP 3 : Mise en place de l'accéléromètre**

- Brancher l'accéléromètre sur la carte ARDUINO ;
- Faire une première acquisition de quelques secondes pour mesurer le champ de pesanteur avec capteur statique, sur la direction du capteur concernée ;
- Après retournement du capteur, refaire une mesure et en déduire la correction à apporter pour régler le zéro de l'accéléromètre ;
- Mesurer g précisément et en déduire la correction à apporter sur les accélérations pour ne percevoir que celles liées au mouvement de la masse.
- En cas de signal très bruité, on pourra lisser le signal en appliquant une *moyenne glissante* dans le programme Python :

Principe d'une moyenne glissante

Un lissage par moyenne glissante consiste à remplacer (si possible) chaque valeur a_k correspondant à l'instant t_k par une moyenne des valeurs a_i au voisinage de l'instante t_k . Le nombre de termes choisis détermine la force du lissage. Le caractère centré de la fenêtre permet d'éviter la création d'un déphasage entre signal brut et signal lissé.

• MANIP 4 : Période propre

- Réaliser une première acquisition de mouvement d'oscillations libres dans l'air. Mesurer la période propre T_0 , et comparer avec une mesure directe au chronomètre.
- Pour différentes masses accrochées, mesurer la période propre T_0 de l'oscillateur.
- Vérifier par une régression linéaire pour des variables bien choisies si la loi $T_0 = 2\pi \sqrt{m/k}$ est bien vérifiée.
- En déduire une seconde estimation de la raideur k
- En déduire si le ressort peut-être considéré sans masse. Si oui à quelle condition ?

II.2. Dans l'eau

Dans la suite, on fonctionne avec la masse immergée, à laquelle on ajoute une rondelle pour accentuer le frottement fluide. Pour ne pas avoir de problème lié à une résonance trop forte en RSF, on prendra trois masses associées à la rondelle de taille moyenne. Pour l'approche théorique, on supposera que la force de frottement est linéaire en la vitesse : $\vec{F} = -\alpha\vec{v}$, où α est le *coefficient de frottement*. On admettra qu'il n'est pas nécessaire de considérer l'effet de la force d'Archimède pour l'étude des régimes variables.

• MANIP 5 : Pseudo-période et facteur de qualité

- Mesurer la période propre T_0 du système nouvellement équipé dans l'air.
- Immerger la masse. Mesurer la *pseudo-période*^a T . En déduire la valeur du coefficient de frottement α de la force, sachant qu'en théorie^b

$$T_0 = T \sqrt{1 - \frac{\alpha^2}{4mk}}.$$

a. c'est-à-dire l'équivalent de la période mais pour un mouvement oscillatoire amorti.

b. cf cours sur les régimes libres en électrocinétique.

III. Régime sinusoïdal forcé

Grâce au moteur et à la poulie, la cote de l'extrémité supérieure du ressort suit une loi horaire sinusoïdale, c'est-à-dire du type

$$z_1(t) = a \cos(\omega t) + z_{10} \quad \text{avec} \quad \omega = 2\pi f,$$

où f est la *fréquence* de rotation du moteur, ω sa *pulsation*, et a l'*amplitude* du forçage. On rappelle que dans le référentiel du laboratoire, en prenant l'origine des cotes en la position d'équilibre, l'équation du mouvement de la masse s'écrit :

$$\ddot{z} + \frac{\alpha}{m} \dot{z} + \omega_0^2 z = \omega_0^2 a \cos(\omega t) \quad \text{avec} \quad \omega_0 = \sqrt{\frac{k}{m}}.$$

Q2. Comment s'écrit le lien entre amplitude du mouvement et amplitude de l'accélération lorsque l'on se place en régime sinusoïdal forcé ?

• MANIP 6

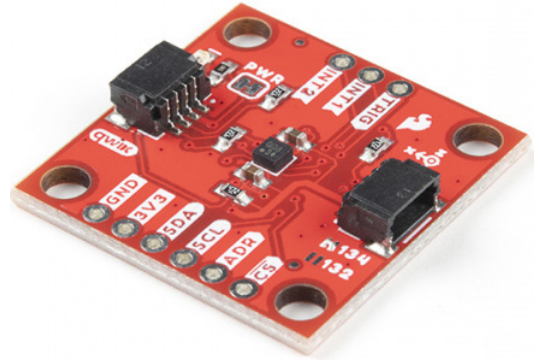
- Relever la courbe de réponse fréquentielle, donnant l'amplitude des oscillations en fonction de la fréquence.
- Tracer le graphe associé à l'aide du tableur. Mesurer l'amplitude du forçage a , puis la fréquence de *résonance* f_r et l'amplitude à la *résonance* $Z_{m_{\max}}$. De quel type de filtrage mécanique s'agit-il ?
- En déduire la valeur du coefficient de frottement α de la force.
- Que vaut le déphasage entre forçage et position de la masse lorsque la fréquence vaut la fréquence propre de l'oscillateur : $f = f_0$? et à "haute fréquence", c'est-à-dire pour $f \gg f_0$?

IV. ANNEXE - Utilisation d'un accéléromètre

IV.1. Présentation de l'accéléromètre KX132 (Kionix)

La platine Sparkfun KX132 est essentiellement constituée d'un accéléromètre numérique Kionix KX132 et d'une connectique Qwiic. Le KX132 est un accéléromètre à trois axes à faible consommation et résolution 16 bits avec quatre plages de mesure d'accélération sélectionnables par l'utilisateur de $\pm 2g/4g/8g/16g$ (avec $g = 9,8\text{m.s}^{-2}$) et a un débit de données de sortie jusqu'à 10 kHz (max).

Il fournit en outre des options de post-traitement des données en 3 étapes configurables par l'utilisateur avec filtre passe-bas, filtre passe-bas/passe-haut et moteur de calcul RMS. Enfin il comprend plusieurs fonctionnalités additionnelles, notamment la détection de chute libre, la détection d'impulsion directionnelle et de double impulsion (tapotements), la détection d'orientation d'inclinaison, etc.



Pour le transfert des données, le Qwiic KX132 peut s'interfacer avec des microcontrôleurs utilisant les protocoles de transfert numérique I2C ou SPI à haute vitesse. Nous utiliserons le protocole I2C.

Le bus I2C est une interface série à deux fils, qui contient un fil d'horloge (Serial Clock Line, SCL) et un fil de données (Serial Data line, SDA). Le connecteur Qwiic (Sparkfun) permet de connecter 4 fils au microcontrôleur (GND, VCC, SDA et SCL) à l'aide d'une seule fiche (même principe que les connecteur grove).

Pour plus d'information sur le principe de fonctionnement des accéléromètres de type MEMS, se reporter à l'approche documentaire dédiée.

IV.2. Sketch ARDUINO


Le *sketch* (programme) Arduino `accelero_3axes_I2C.ino` est téléchargeable sur le site, ainsi que le programme d'acquisition sous Python, `acquisition_capteur.py`, à adapter pour post-traitement.

Le sketch utilise la bibliothèque `wire.h` pour implémenter le protocole I2C, ainsi que la bibliothèque `SparkFun_Qwiic_KX13X.h` dédiée à ce capteur, qui rend l'écriture du code plus concise et plus facile.

La gamme de mesure choisie est, a priori $\pm 8g$, à modifier si besoin. Le délai entre 2 lectures doit être fixé égal à la période de transfert de données (ODR, Output Data Rate). L'ODR étant par défaut de 50 Hz, le délai est fixé à 20 ms.

IV.3. Extrait de documentation

Les informations techniques principales fournies par le fabricant sont reproduites dans les deux pages suivantes.

| | | |
|---|---|---|
|  | ± 2g / 4g / 8g / 16g Tri-axis Digital Accelerometer Specifications | PART NUMBER: KX132-1211 Rev. 1.0 31-Jul-2019 |
|---|---|---|

Product Specifications

Mechanical

(specifications are for operation at 2.5V and T = 25C unless stated otherwise)

| Parameters | | Units | Min | Typical | Max |
|---|-------------------------|----------|----------------------|-----------------------|-------|
| Operating Temperature Range | | °C | -40 | - | +105 |
| Zero-g Offset | | mg | | ±25 | ±90 |
| Zero-g Offset Variation from RT over Temp. | | mg/°C | | 0.25 | |
| Sensitivity ¹ (16 bit) | GSEL1=0, GSEL0=0 (±2g) | counts/g | 15401 | 16384 | 17367 |
| | GSEL1=0, GSEL0=1 (±4g) | | 7700 | 8192 | 8684 |
| | GSEL1=1, GSEL0=0 (±8g) | | 3850 | 4096 | 4342 |
| | GSEL1=1, GSEL0=1 (±16g) | | 1925 | 2048 | 2171 |
| Sensitivity (Buffer 8-bit mode) ^{1,2} | GSEL1=0, GSEL0=0 (±2g) | counts/g | | 64 | |
| | GSEL1=0, GSEL0=1 (±4g) | | | 32 | |
| | GSEL1=1, GSEL0=0 (±8g) | | | 16 | |
| | GSEL1=1, GSEL0=1 (±16g) | | | 8 | |
| Sensitivity Variation from RT over Temperature | | %/°C | | 0.01 (xy) 0.03 (z) | |
| Positive Self-Test Output change on Activation ³ | | g | 0.25 (xy) 0.2 (z) | 0.5 | 0.75 |
| Mechanical Signal Bandwidth (-3dB) ⁴ | | Hz | | 4200 (xy) 2900 (z) | |
| Non-Linearity | | % of FS | | 0.5 | |
| Cross Axis Sensitivity | | % | | 2 | |
| Noise ⁵ | RMS | mg | | 0.7 | |
| | Density | µg/√Hz | | 130 | |

Table 1: Mechanical Specifications

Notes:

- Resolution and acceleration ranges are user selectable via I²C or SPI
- Sensitivity is proportional to BRES in BUF_CNTL2.
- Requires changing of STPOL bit in INC1 register to 1 prior to performing self-test
- Signal bandwidth varies with Output Data Rate (ODR), and Low Pass Filter setting. Measured with ODR = 25600Hz, LPRO = 1 settings.
- Noise varies with ODR, power mode, and the Average Filter Control (AVC) settings. Measured with RES = 1, ODR = 50Hz, IIR_BYPASS = 0, LPRO = 1 settings.



± 2g / 4g / 8g / 16g Tri-axis Digital Accelerometer Specifications

PART NUMBER:

KX132-1211

Rev. 1.0

31-Jul-2019

Electrical

(specifications are for operation at 2.5V and T = 25C unless stated otherwise)

| Parameters | | Units | Min | Typical | Max |
|--|---|-------|--------------|----------------|--------------|
| Supply Voltage (VDD) | Operating | V | 1.7 | 2.5 | 3.6 |
| I/O Pads Supply Voltage (IO_VDD) | SPI, I ² C (Fast/Standard mode) | V | 1.2 | | 3.6 |
| | I ² C (High Speed mode) | | 1.7 | | 3.6 |
| Current Consumption (Accelerometer Only) | Operating (High Performance with Wake-up Detection) ODR=400Hz | µA | | 148 | |
| | Operating (Low power with Wake-up detection) ODR=0.781Hz ¹ | | | 0.53 | |
| | Operating (Low Power with Wake-up detection plus Advanced Data Path) ODR=0.781Hz ⁷ | | | 0.67 | |
| Standby Current Consumption | | µA | | 0.5 | |
| Output Low Voltage (V _{OL}) ² | (IO_VDD < 2V) | V | - | - | 0.2 * IO_VDD |
| | (IO_VDD ≥ 2V) | V | - | - | 0.4 |
| Output High Voltage (V _{OH}) | | V | 0.8 * IO_VDD | - | - |
| Input Low Voltage (V _{IL}) | | V | - | - | 0.2 * IO_VDD |
| Input High Voltage (V _{IH}) | | V | 0.8 * IO_VDD | - | - |
| Start Up Time ³ | | ms | 2 | | 1300 |
| Power Up Time ⁴ | | ms | | 20 | 50 |
| I ² C Communication Rate | | MHz | | | 3.4 |
| I ² C Slave Address (7-bit) | | | | 0x1E / 0x1F | |
| WHO_AM_I register value | | | | 0x3D | |
| Output Data Rate (ODR) ⁵ | | Hz | 0.781 | 50 | 25600 |
| Output Signal Bandwidth (-3dB) ⁶ | | Hz | | ODR/9 or ODR/2 | |

Table 2: Electrical Specifications

Notes:

1. Current varies with Output Data Rate (ODR) as shown in Figure 2, types and number of enabled digital engines, the average filter control settings, and VDD. Measured with OWUF<2:0> = 0, OSA<3:0> = 0, AVC<2:0> = 1, LPSTPSEL = 1.
2. For I²C communication, this assumes a minimum 1.5kΩ pull-up resistor on SCL and SDA pins.
3. Start up time is from PC1 set to valid outputs. Time varies with ODR, Power Mode, and FSTUP bit setting (see Figure 1).
4. Power up time is from VDD valid to device boot completion.
5. Typical values. ODR is user-selectable via I²C or SPI. See ODCNTL register for details.
6. Refers to accelerometer's raw output data. Additional bandwidth control is available using the Advanced Data Path (ADP) engine.
7. Measured with RMS_AVC<2:0> = 1, OADP<3:0> = 0, LPSTPSEL = 0.

TP Masse-Ressort

Liste de matériel - 3 postes

Le TP dure 2H en parallèle du TP Chronophotographie (2H aussi) de sorte que les élèves peuvent permuter à mi-séance. Il y a donc besoin de 3 postes de chaque.

Oscillations forcées - Matériel pour 2 postes

- Un montage masse-ressort sur pied avec forçage à moteur+poulie + masses et rondelles ajustables + réservoir à eau pour frottement fluide ;
- Plaquette ARDUINO avec plaquette micro-fils + câble USB ;
- Un capteur accéléromètre SparkFun KX132 + son câble de connexion Qwiic + bouclier Sparkfun Qwiic pour Arduino Uno ;
- Un système de fixation de l'accéléromètre sur la tige rigide ;
- Un double décimètre ;
- Un chronomètre ;