

Étude d'un système de récupération de l'énergie utilisant deux volants d'inertie

*Comment utiliser au mieux un volant
d'inertie pour récupérer de l'énergie
cinétique par contact ?*

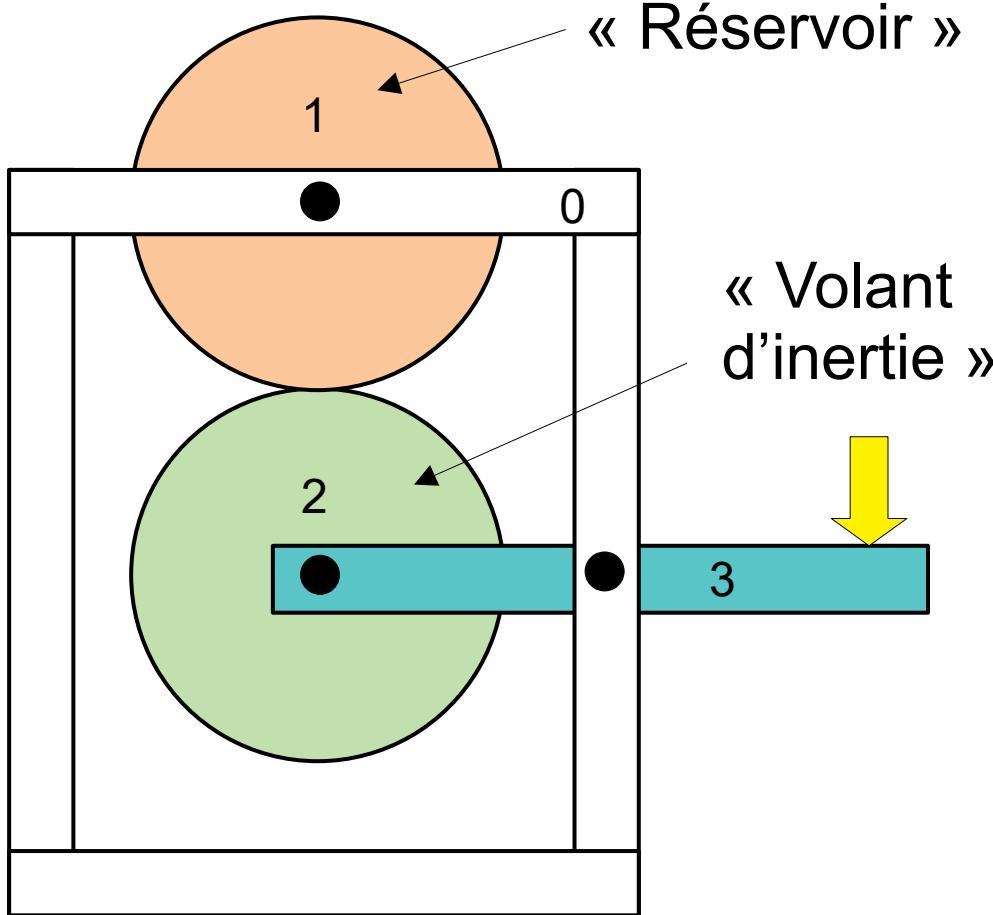
Tristan Chevreau - N° Candidat : 22848

Sommaire de la présentation

- I - En groupe : conception et modélisation de la maquette
- II – Conception, fabrication et implémentation des capteurs de vitesse
- III – Expériences sur la maquette
- IV – Synthèse, points à améliorer et conclusion

I - En groupe : conception et modélisation de la maquette

En groupe : conception de la maquette



Modèle retenu

Attentes :

- lancement à la main
- couplage / découplage
- ajuster la force normale au contact (⬇)
- capteurs de vitesse
- vitesses raisonnables

Contraintes :

- Usinage
- Matériaux disponibles

Fabrication de la maquette

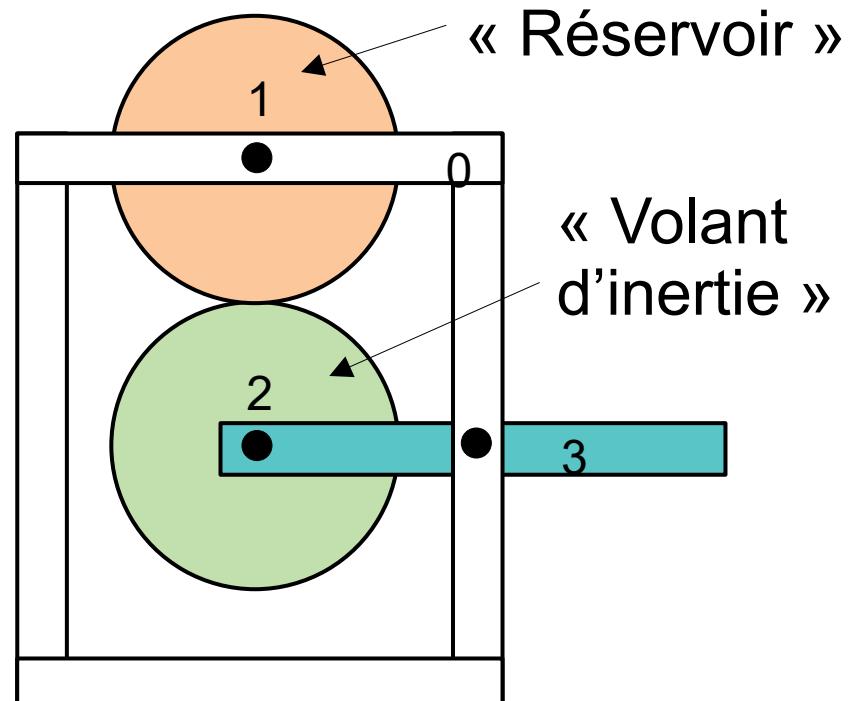


Usinage des disques



Système de fixation des disques

- Disques : bois massif
- Armature : aggloméré
- Fixations : tige filetée, écrous, rondelles



Modèle retenu

Résultat



Maquette en phase de contact



Maquette en rotation libre (deux disques)

En groupe : modélisation du système

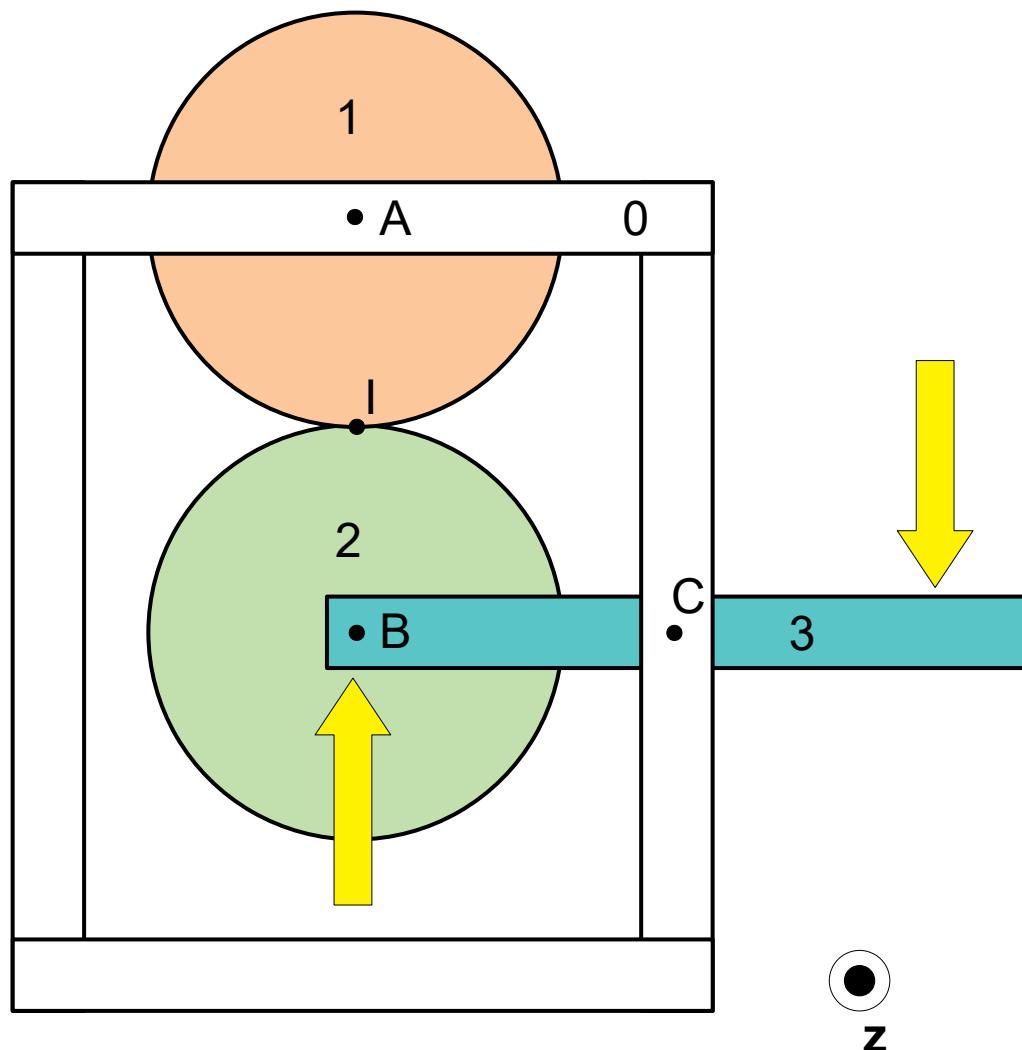
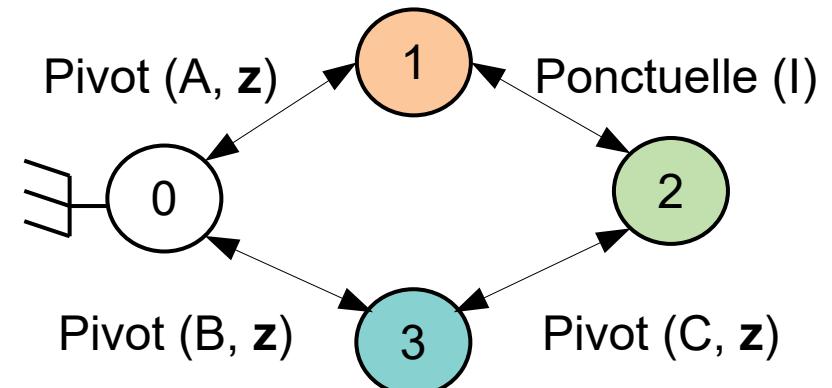


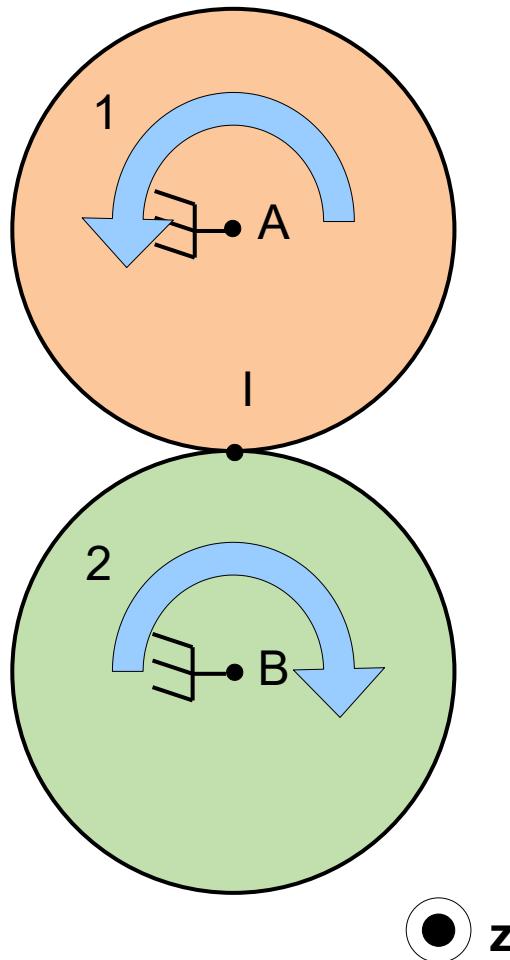
Schéma du système

- Problème plan (symétries)
- Liaisons pivot 01 et 02 imparfaites
- Pivot C : simplifiable



Graphe de structure du système

Données à mesurer



Basse vitesse : Couple de frottement sec des roulements à billes en A et B

Frottement dynamique et statique de Coulomb en I

Dispositif de mesure de la vitesse angulaire

Autres mesures
(masse, rayon \Rightarrow inertie ...)

II – Conception, fabrication et implémentation des capteurs de vitesse

Éléments disponibles

Effet Hall

Module ST054

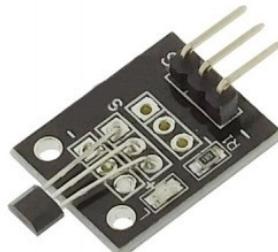
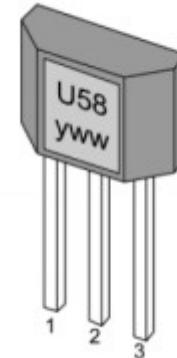


Photo du
module
(d'après
gotronic.fr)

Utilisable en l'état

Composant US5881



Représentation du
composant
(d'après datasheet
constructeur)

À transformer en module
utilisable avec Arduino

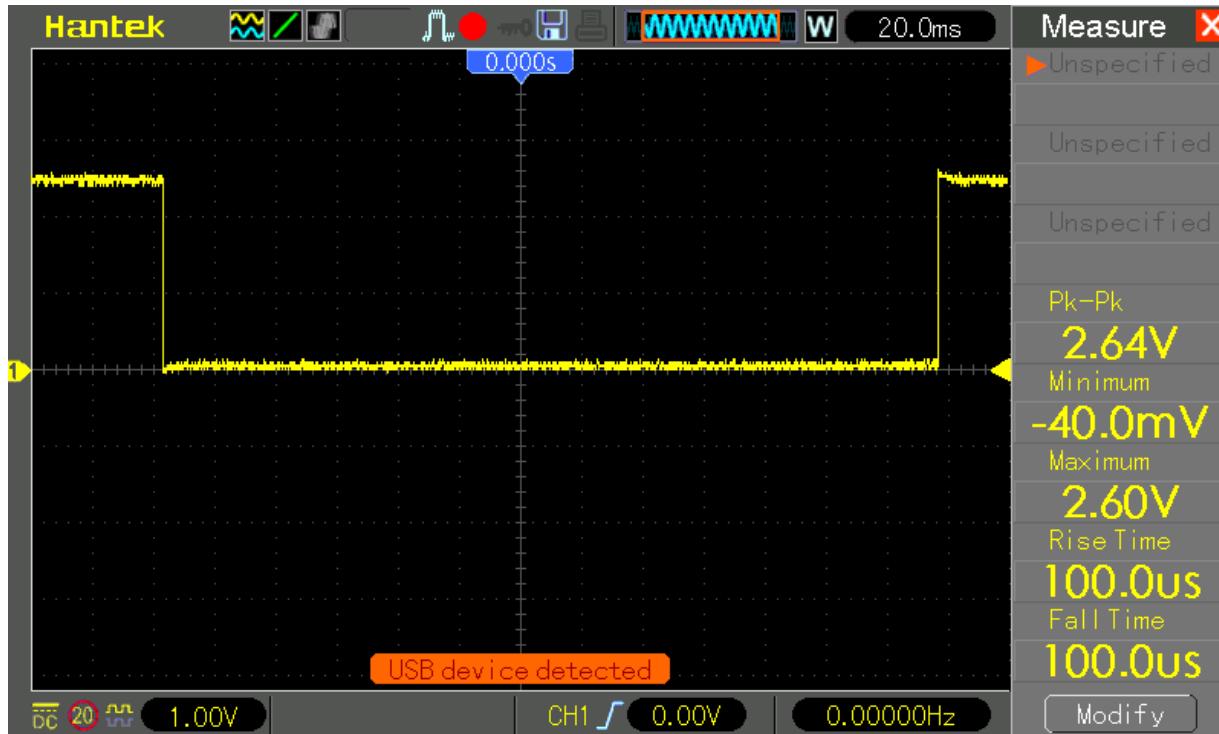
Analyse des performances du module ST054 à "répliquer"

Conditions de l'expérience :

- Alimentation stabilisée de 5V avec ampèremètre



Photo du module (d'après gotronic.fr)



Oscilloscope et mesures du module ST054

MESURES

- *Temps de montée et descente au minimum mesurable*
- $V_{HIGH} = 2,64 \text{ V} ; V_{LOW} = 0 \text{ V}$
- $I_{HIGH} = 2,6 \text{ mA} ; I_{LOW} = 6,0 \text{ mA}$
- $L_{min} = 4 \text{ cm}$

Cahier des charges du module à fabriquer

N°	Description	Valeur
1	Se baser sur le composant US5881-UA et fonctionner avec une alimentation de 5V (port USB)	$I_{\max} = 15 \text{ mA}$ $V_{\text{fonct}} = 5 \text{ V}$
2	Répliquer la tension de sortie en mode HIGH et LOW du ST054	$V_{\text{HIGH}} = 2,64 \pm 0,5 \text{ V}$ $V_{\text{LOW}} = 0 \pm 0,5 \text{ V}$
3	Indiquer la présence d'un aimant sans nécessiter de contact	$L_{\min} = 3 \text{ cm}$
4	Basculer d'un état à l'autre rapidement	$T \leq 1 \text{ ms}$

Conception et fabrication du second module

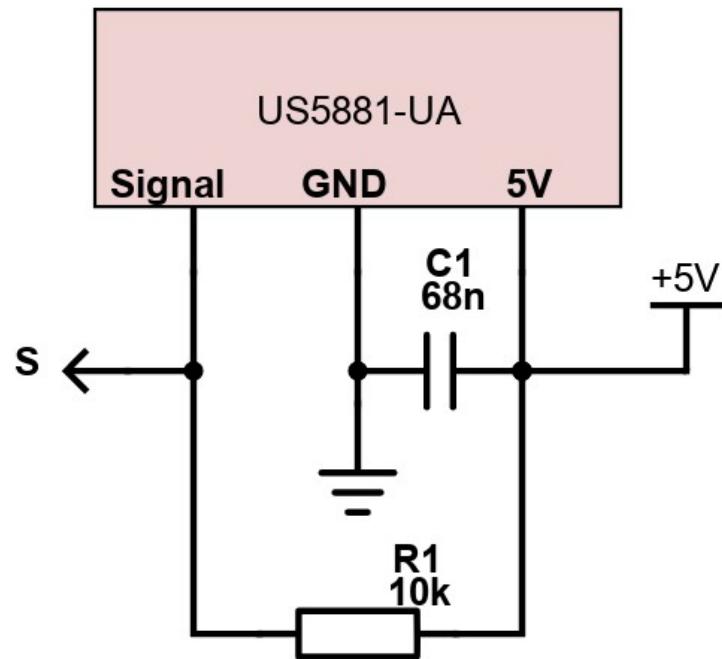


Schéma électrique retenu

- Notes et schémas d'application de la datasheet
- LED inutile
- Environnement non bruité

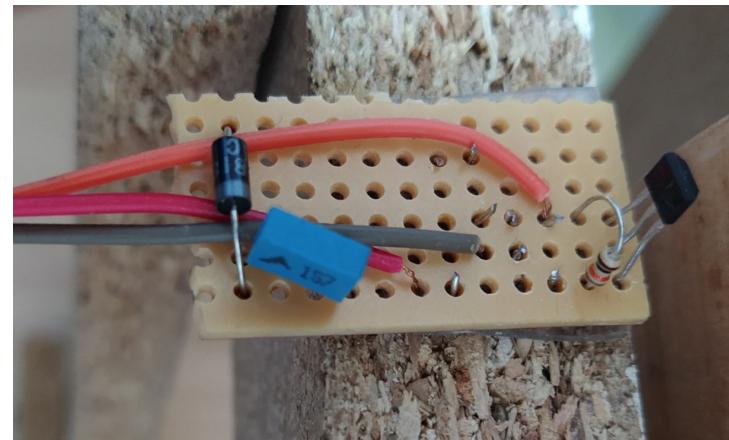
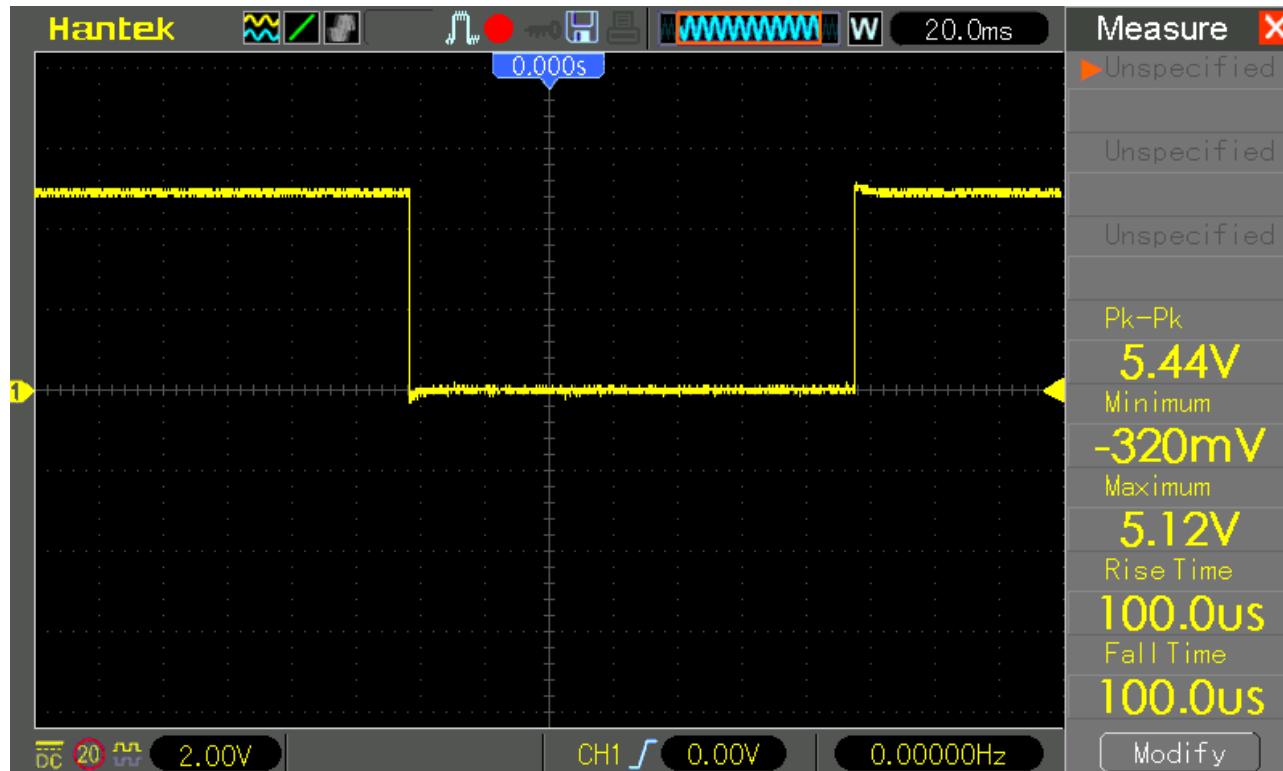


Photo du résultat

Performances du second module

Conditions des expériences :

- Alimentation stabilisée de 5V avec ampèremètre



Oscillogramme sans LED

Consommation :

$$I_{\text{LOW}} = 2,3 \text{ mA}$$
$$I_{\text{HIGH}} = 1,9 \text{ mA}$$

Exigences vérifiées :

- 1 (Alimentation)
- 3 (Distance, hors écran)
- 4 (Basculement)

Exigences non vérifiées :

- 2 (Tensions, mieux)

Installation des aimants et des capteurs

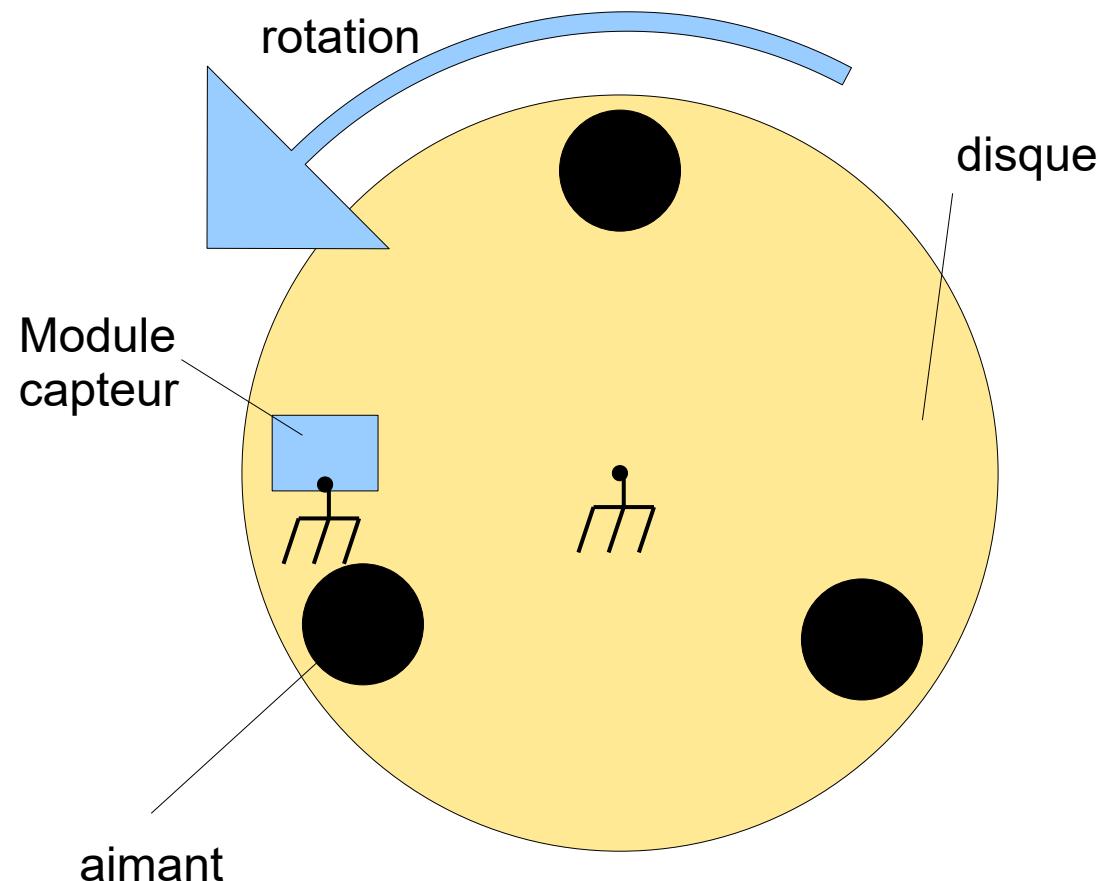
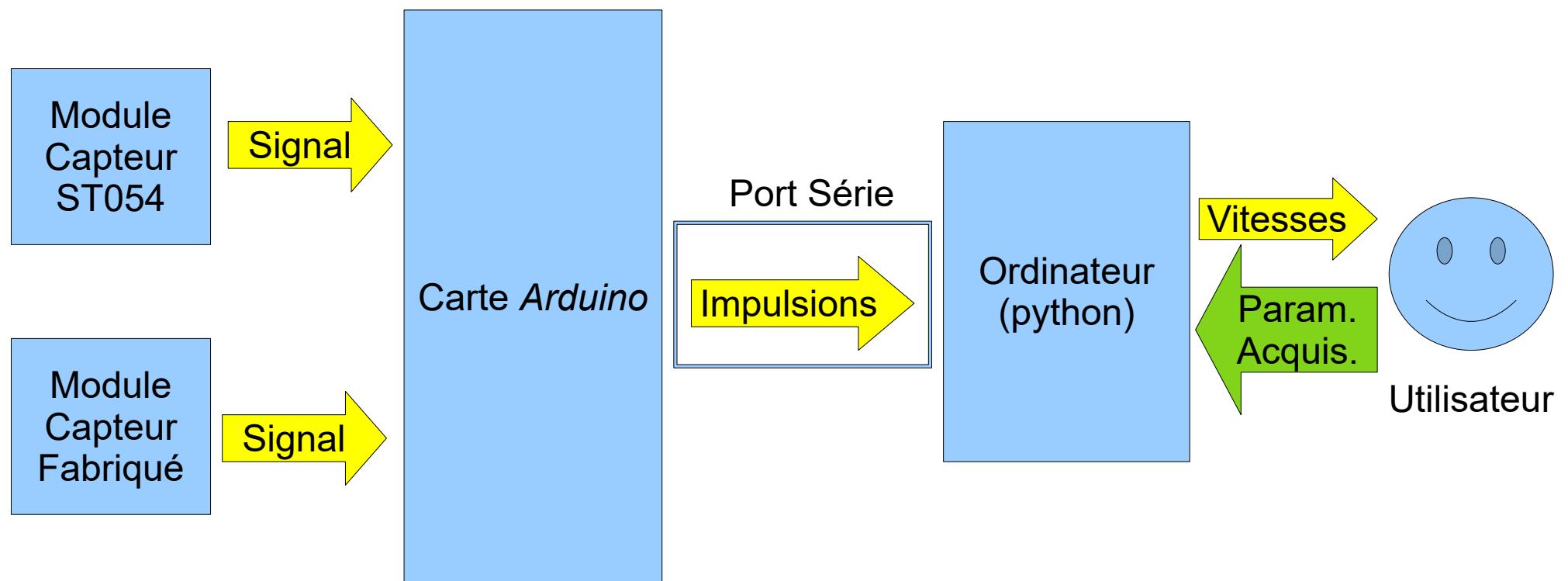
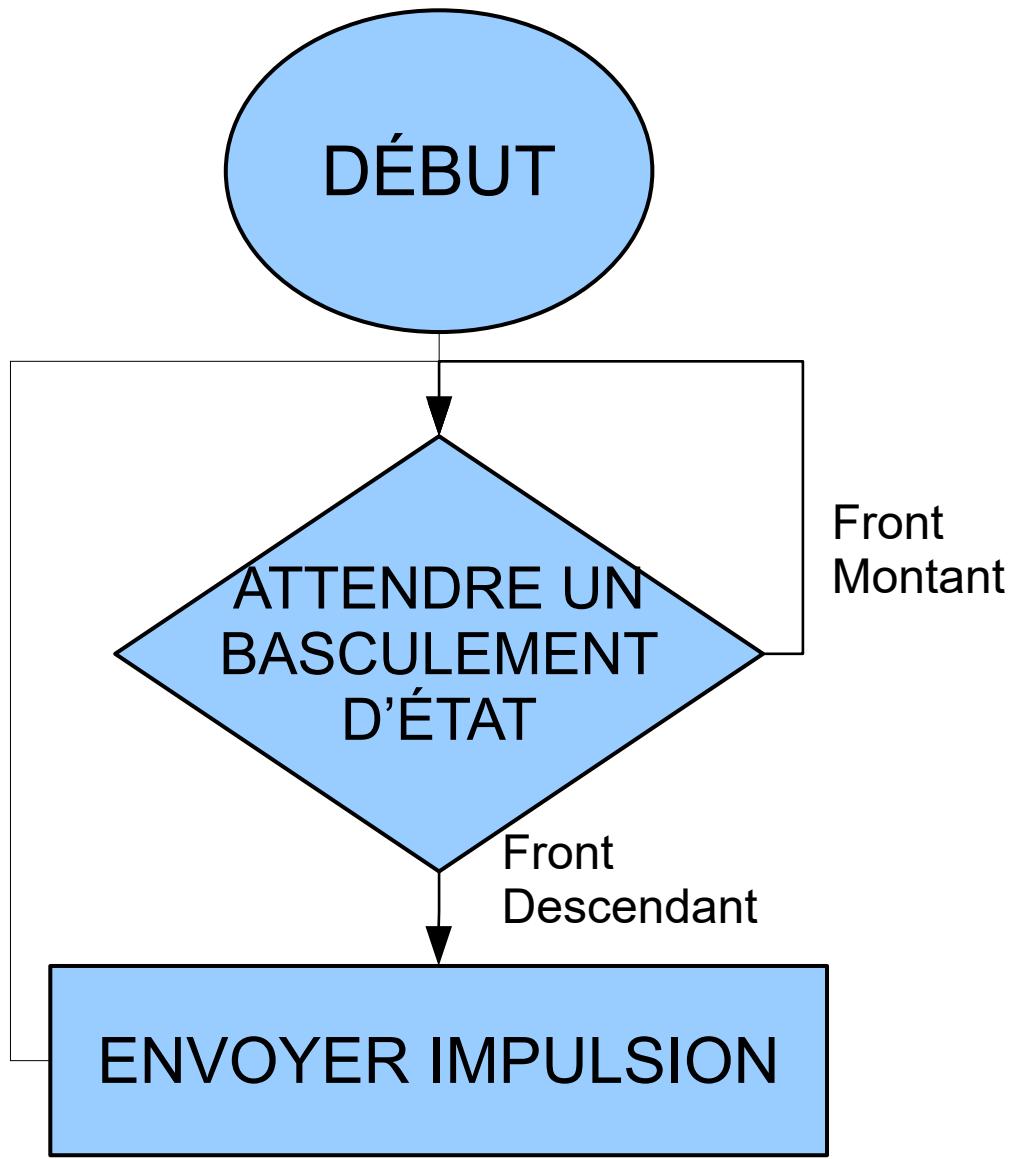


Schéma et photographie de l'installation

Les flux



Le code : Arduino



- rapidité d'exécution
- fronts descendants
- symétrie pour les deux capteurs
- port série

A screenshot of a serial monitor window titled "COM4". The window shows a list of received messages in a table format:

Time	Message
18:49:01.783	→ A
18:49:03.669	→ A
18:49:04.451	→ A
18:49:06.848	→ B
18:49:08.094	→ B
18:49:08.973	→ B

Moniteur série lisant les impulsions envoyées par la carte Arduino

Le code : Python

```
>>>
== RESTART: C:\Users\Tristan\Google Drive\Scol
Chemin ? Test
COM ? 5
Duree acquisition ? (s) 42
Vitesse de depart divisee par pi ? (rad/s) 6
Ouverture du port serie ...
OK
3
2
1
OK
Attente d'atteinte de la vitesse initiale ...
OK
Acquisition ...
Termine
>>> |
```

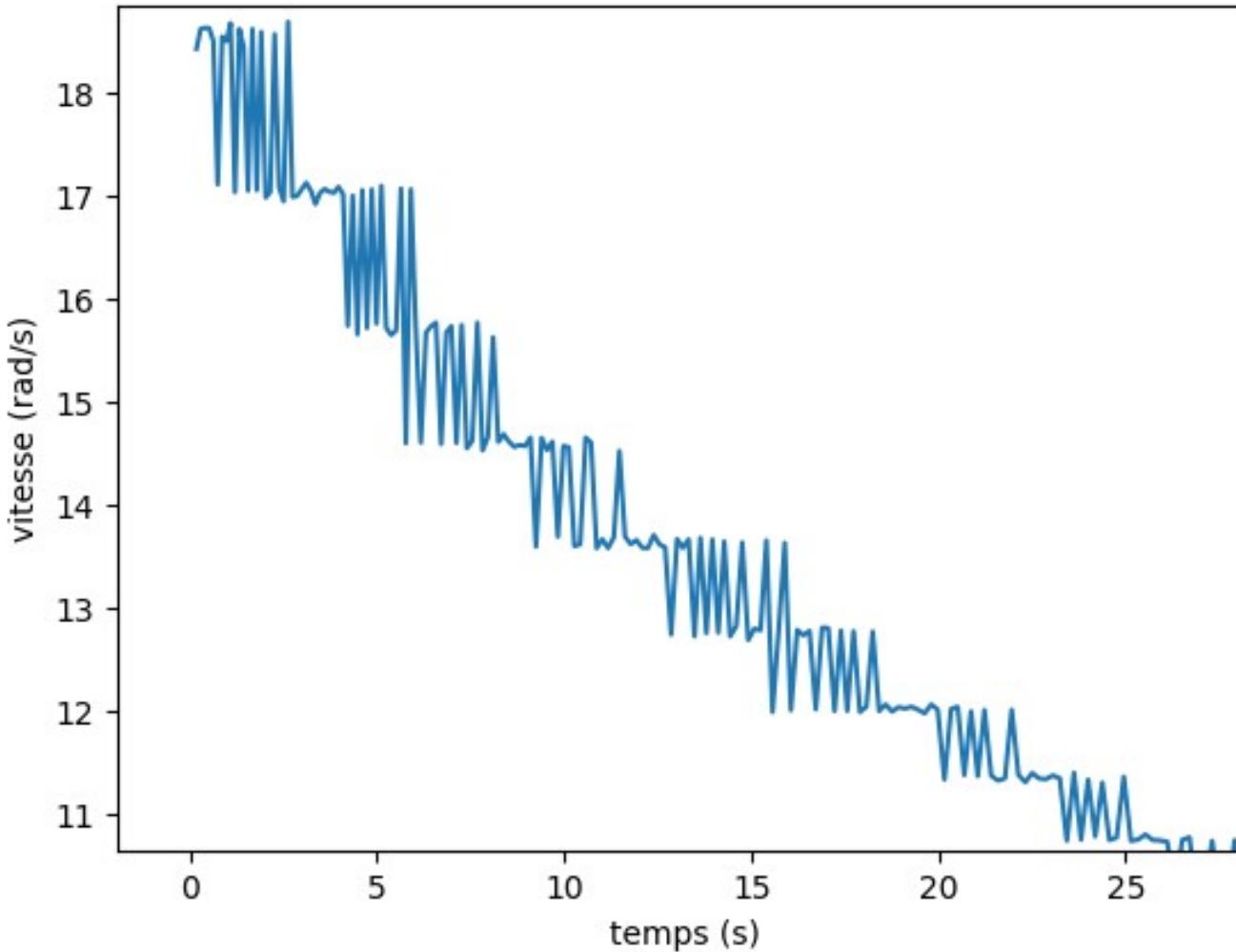
Exemple d'interaction

- Objectif : facilité d'utilisation
- Plusieurs phases :
 - Acquisition
 - Enregistrement (données brutes)
 - Calcul

$$\omega = \frac{2\pi}{N} \text{moyenne}(\Delta t_1, \dots, \Delta t_n)$$

- Plusieurs fonctions à combiner

Performances des mesures



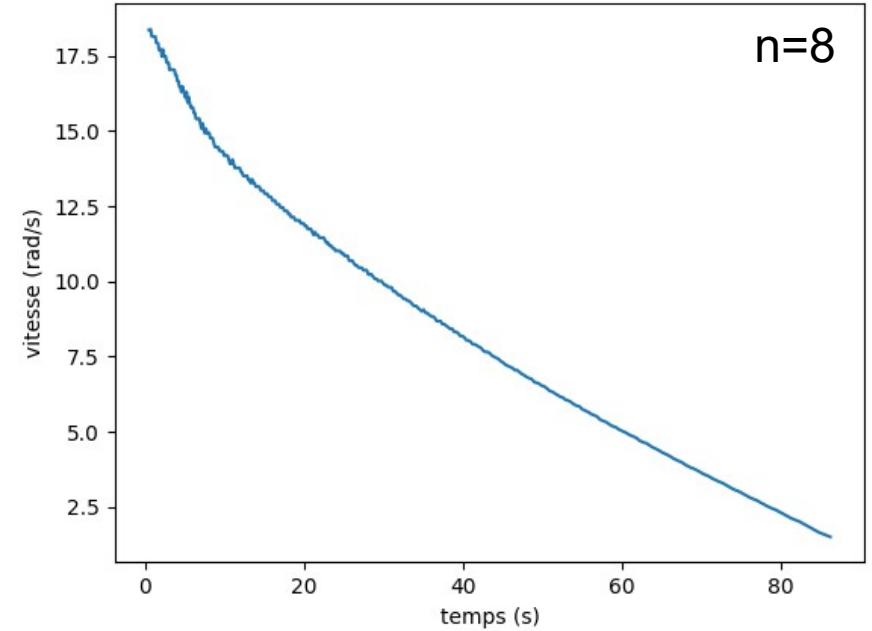
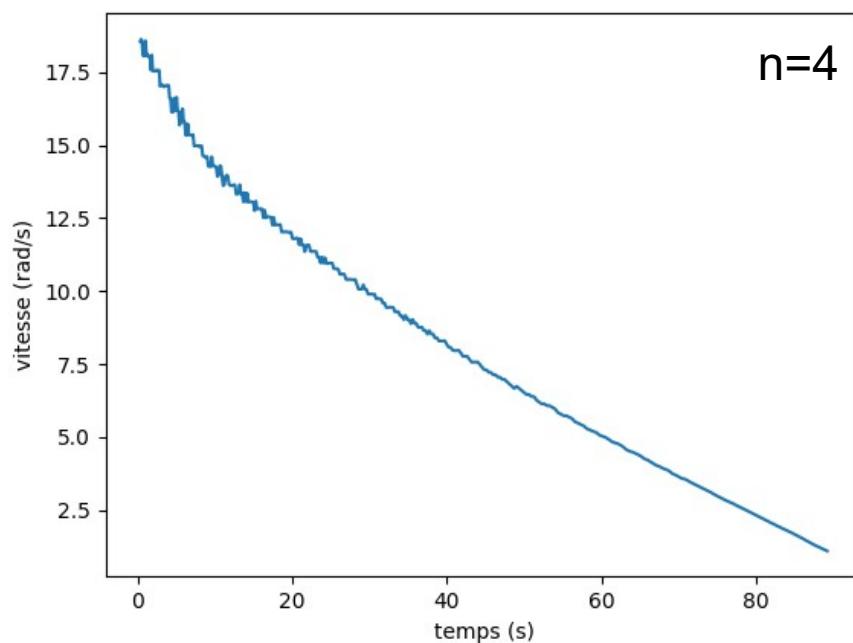
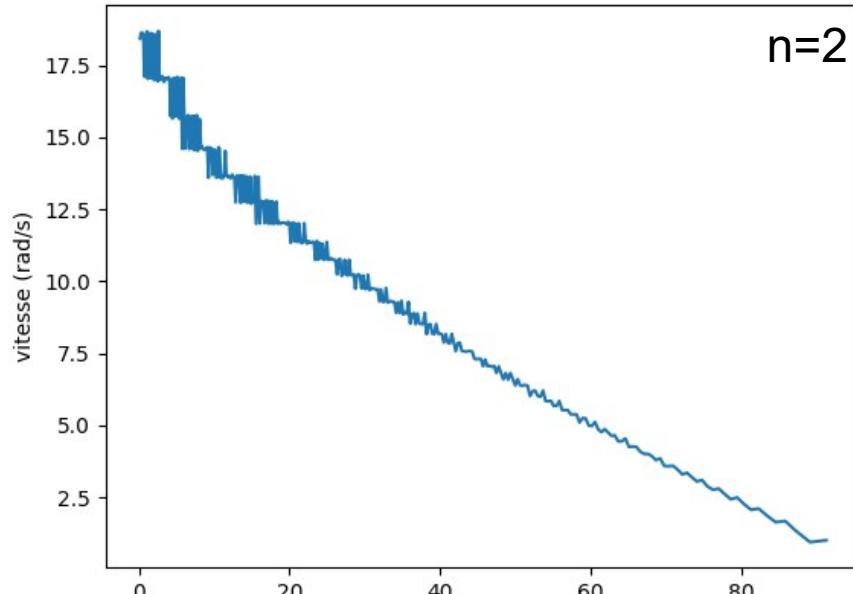
Pour une moyenne
sur deux « ticks »

Hypothèses :

- Port série arduino
- Vitesse de lecture

Artefact obtenu pour une vitesse calculée sur 2 points
(Zoomé)

Solution : moyennes glissantes

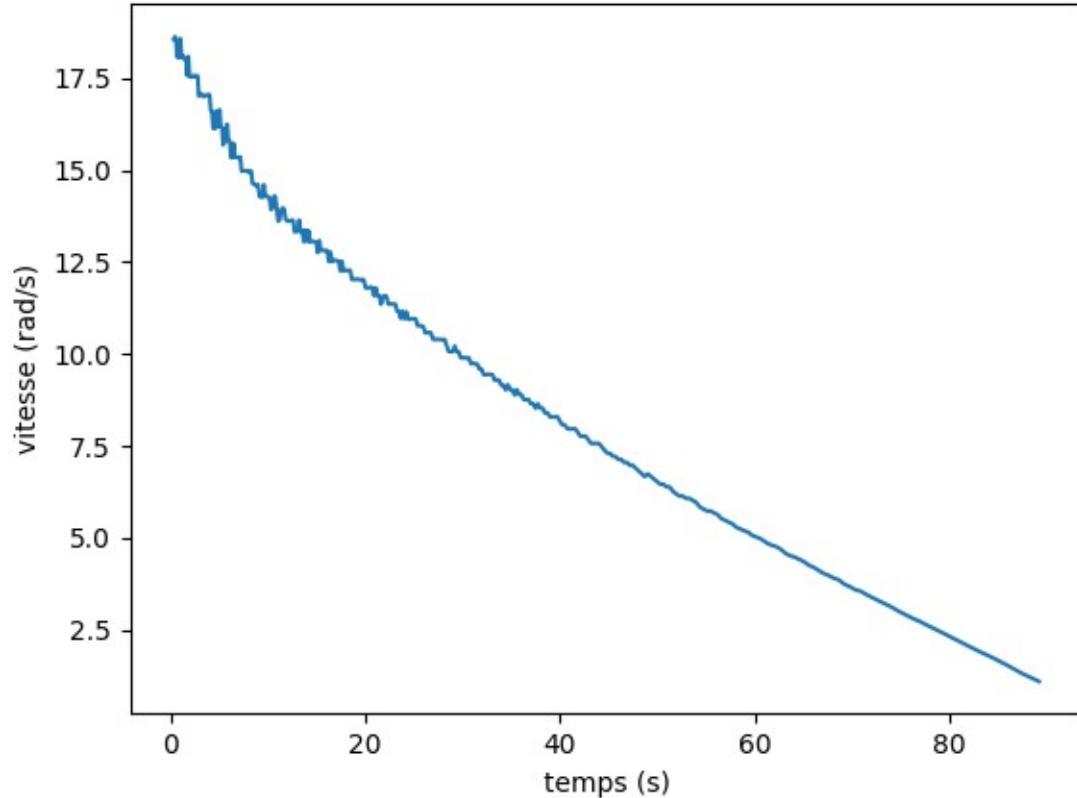


⇒ Meilleur compromis (1 tour)

III – Mesures et expériences sur la maquette : courbes, rendements

Couple de frottement sec des pivots

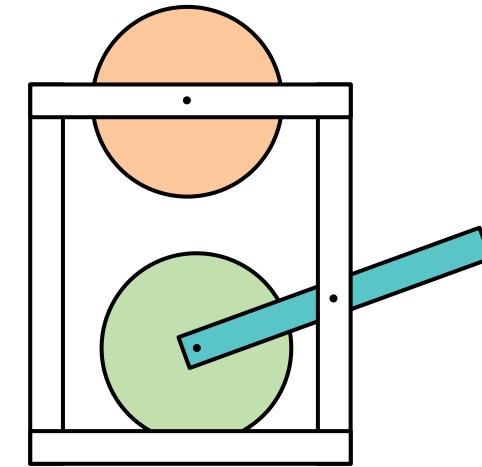
en A et B : Loïc Weislinger



Courbe pour un lancer d'un seul disque sans contact

$$C = -13 \cdot 10^{-3} N \cdot m$$

Lancer à vide d'un disque :



Décroissance linéaire
⇒ frottement sec : OK

$$\text{TEC : } C = \frac{d\omega}{dt} I$$

Frottement statique et dynamique entre les deux disques : Laurent Josseaume

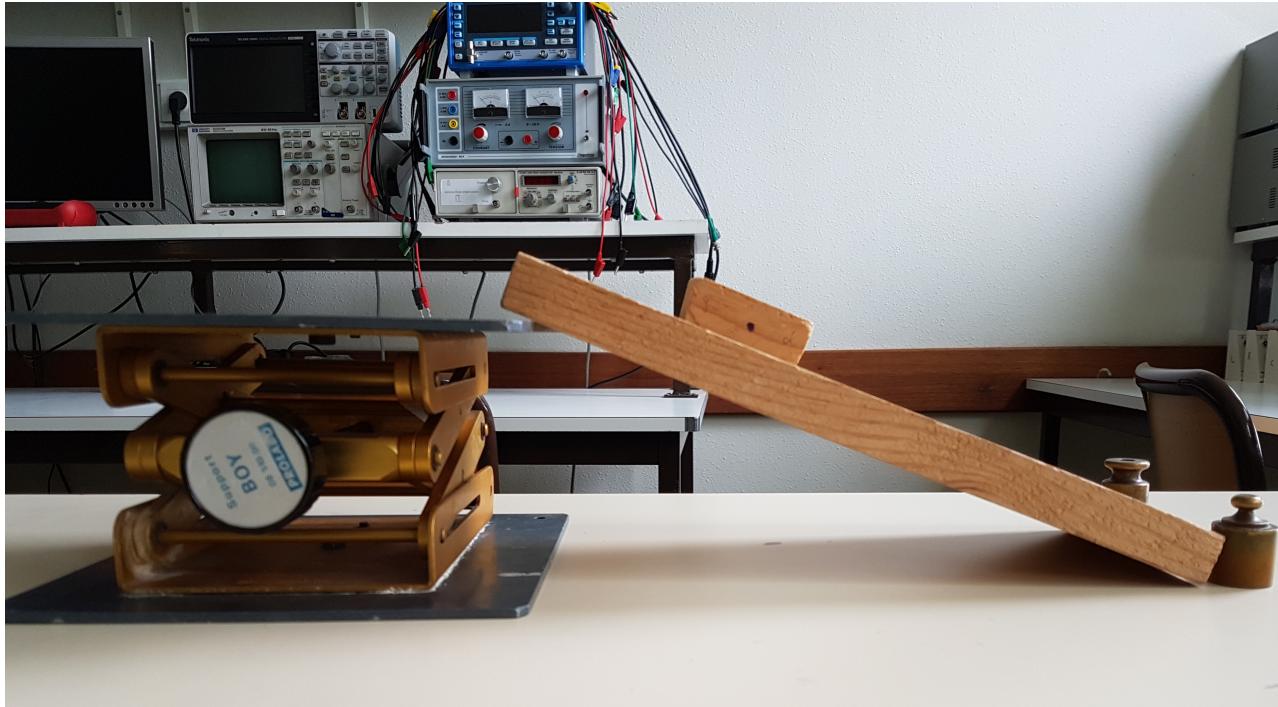


Photo du montage expérimental

$$f_s = 0,53$$

$$f_d = 0,42$$

Protocole

- Lancer le "réservoir"
- Attendre qu'il atteigne la vitesse de départ : 3 tours/s
- Mettre en contact le "réservoir" et le volant en **choisissant la force normale au contact**
- On calcule le rendement du "réservoir" au volant d'inertie

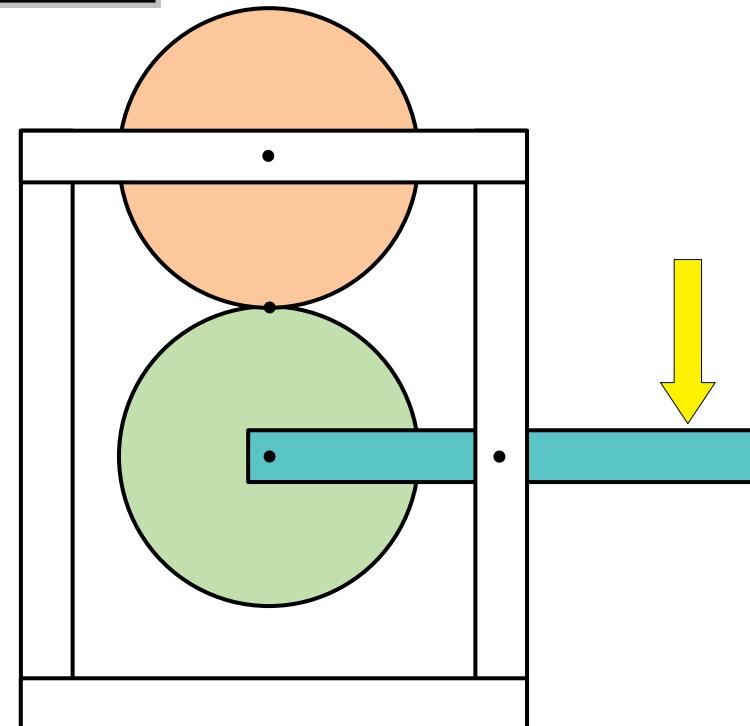
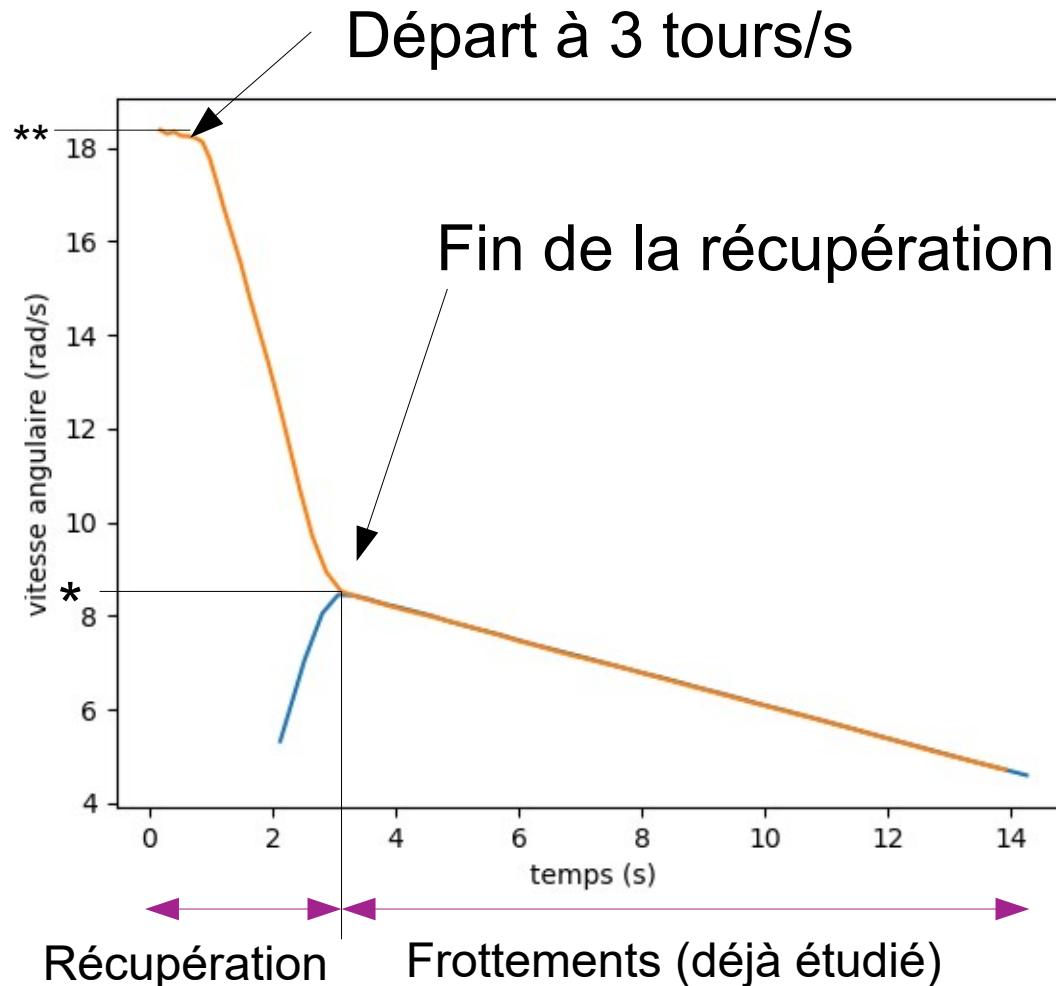


Schéma des disques en contact ; en jaune, la force de contact appliquée

$$\eta = \frac{E_{c, \text{volant, final}}}{E_{c, \text{reservoir, initial}}} \underset{\text{disques identiques}}{=} \frac{\omega_{c, \text{volant, final}}^2}{\omega_{c, \text{reservoir, initial}}^2} \leq 50\%$$

Les courbes obtenues

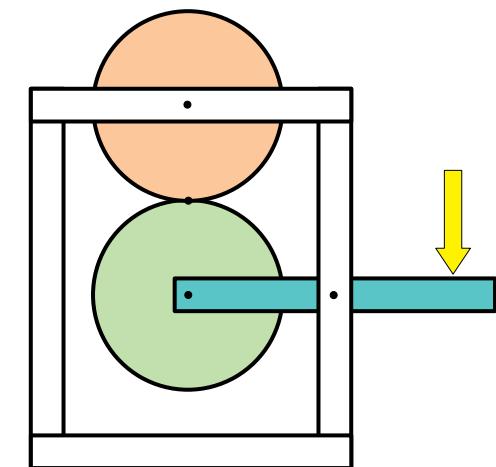
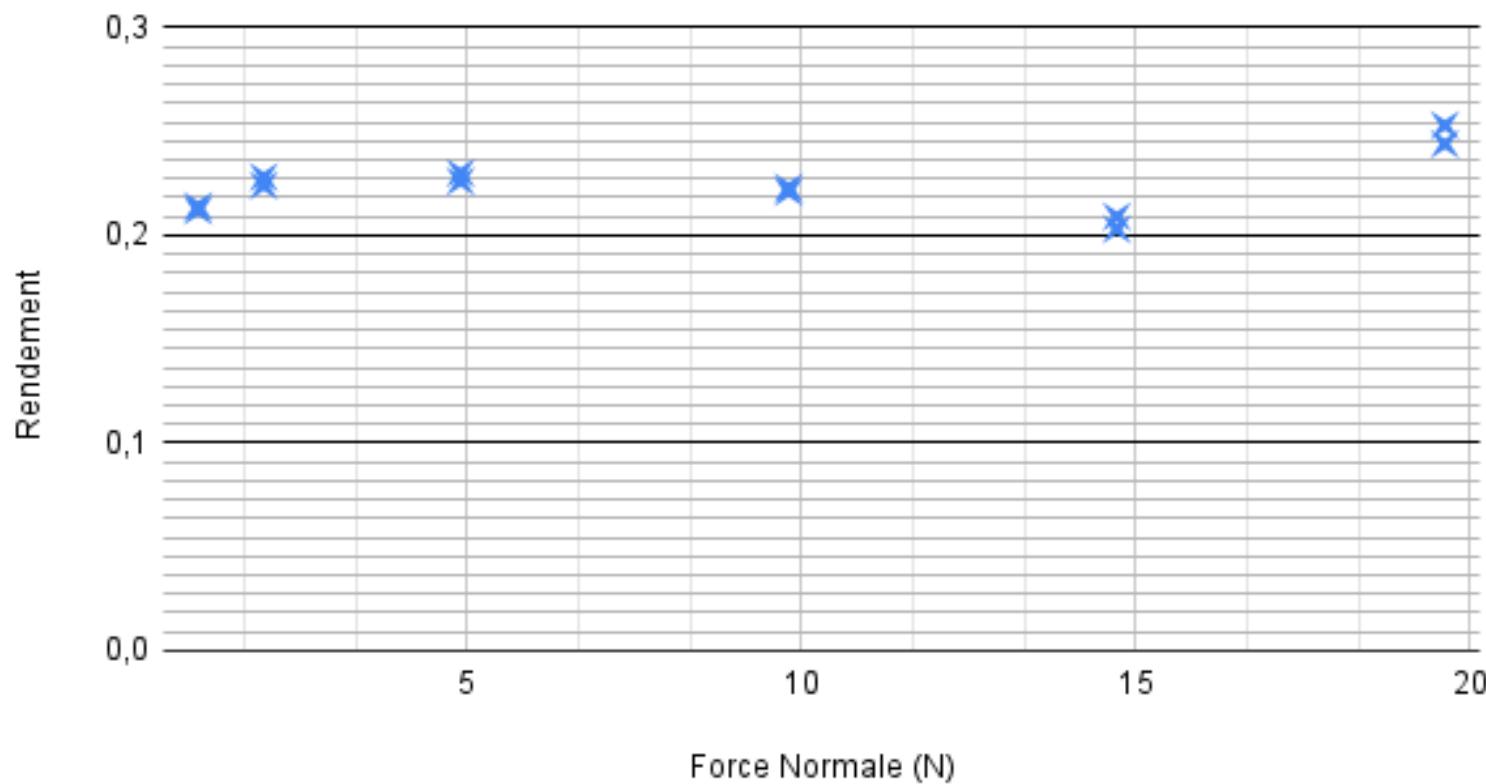


$$\eta = \frac{\omega_{c, \text{volant, final}}^2}{\omega_{c, \text{réservoir, initial}}^2}$$

Exemple d'une courbe de la vitesse angulaire des disques en fonction du temps pour une masse de 100g

Résultats : description

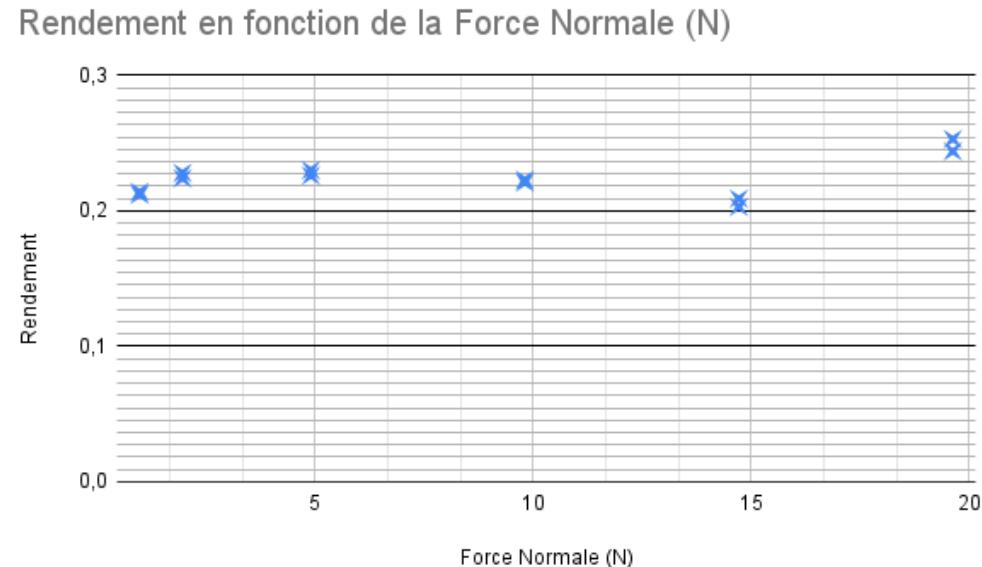
Rendement en fonction de la Force Normale (N)



IV – Synthèse, points à améliorer et conclusion

Résultats : interprétation

- Pas d'impact *a priori*
- Rendement : $\approx 22\%$
- \Rightarrow Facteur négligeable



- Frottement sec des pivots : rapidité
- \Rightarrow Autres échelles : force normale pas trop faible (longueur)

Pistes d'amélioration

Approfondissements possibles

- Alternances de glissement et non glissement
⇒ autre modèle
- Autres matériaux sur la tranche des disques
- Masselottes (Inertie)
- Plus grandes vitesses (équilibrage)
- Mesures avec des masses plus élevées

FIN

Annexe A : code Arduino

```
const int portCapteurA=0; // Capteur achete
const int seuilA = 250;
const int portCapteurB=2; // Capteur fabrique
const int seuilB = 250;
bool valA=false;
bool valB=false;

void setup() {
    Serial.begin(9600);
    pinMode(portCapteurA, INPUT);
    pinMode(portCapteurB, INPUT);
    delay(100);
}

void loop() {
    if (valA != lecture(portCapteurA, seuilA)) {
        if (valA == true) {
            Serial.print("A\n");
        }
        valA = !valA;
    }
    if (valB != lecture(portCapteurB, seuilB)) {
        if (valB == true) {
            Serial.print("B\n");
        }
        valB = !valB;
    }
}

bool lecture(int port, int seuil) {
    return (analogRead(port)>seuil);
}
```

Annexe B : code Python (fonctions)

```
# acquisition

def acquerir():
    return (ticks_A, ticks_B) # donnees = (ticks,ticks)

# gestion de fichiers

def enregistrer(chemin, ticks):

def lire(chemin):
    return ticks

def enregistrer_ensemble(chemin, donnees):

def lire_ensemble(chemin):
    return (lire(chemin+"_A"),lire(chemin+"_B"))

# traitement des donnees

def calculer_vitesses(ticks, n=4):
    return (temps, vitesses)

def calculer_energie_cinetique(ticks):
    return (temps, E_c)

def obtener_maximum(liste):
    return (t,M)

def calculer_rendement(wA,wB):
    return (wB_finale**2/wA_initiale**2)

# enregister les ticks puis afficher la vitesse angulaire du
# disque A au cours du temps
chemin = input("Chemin ? ")
enregistrer_ensemble(chemin, acquerir())
(vA,vB)=lire_ensemble(chemin)
wA = calculer_vitesses(vA)
wB = calculer_vitesses(vB)
plt.plot(wB[0],wB[1])
plt.plot(wA[0],wA[1])
plt.xlabel("temps (s)")
plt.ylabel("vitesse angulaire (rad/s)")
plt.show()

# calculer puis enregistrer les vitesses (pour utilisation par
# Weislenger Loic)
enregistrer_ensemble('trouverfrottementsvitesses',
                     calculer_vitesses(
                         lire_ensemble('trouverfrottements')[0]))

# afficher l'energie cinetique au cours du temps
ticksA = lire("trouverfrottements3_A")
E_c_A = calculer_energie_cinetique(ticksA)
plt.plot(E_c_A[0],E_c_A[1])
plt.xlabel("temps (s)")
plt.ylabel("Ec (J)")
plt.show()

# calculer le rendement d'une experience
chemin = input("Chemin ? ")
(vA,vB)=lire_ensemble(chemin)
wA = calculer_vitesses(vA)
wB = calculer_vitesses(vB)
print("rendement : ", calculer_rendement(wA,wB))
```