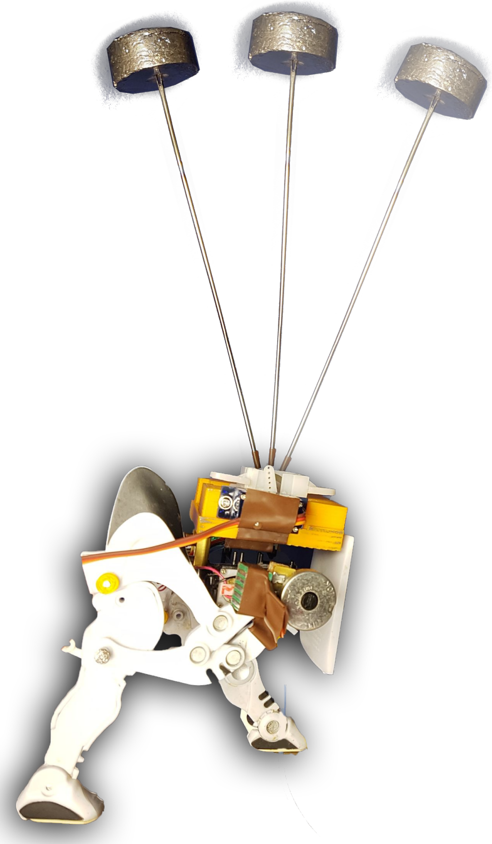
Stabilisation de la marche : balancier caudal

Lémurien utilisant sa queue pour rester vertical



Expérience personnelle

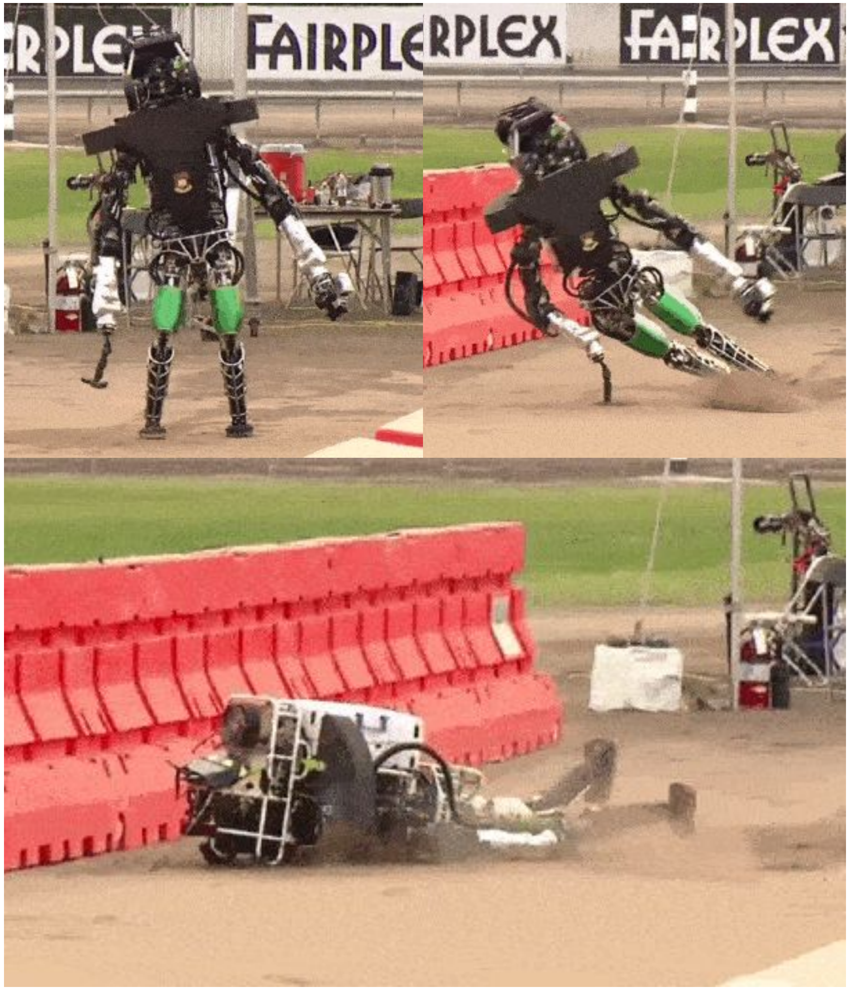
Dans quelle mesure l'utilisation d'une queue asservie permet-elle de stabiliser la marche d'un robot bipède ?

Numéro de candidat: 17045

Sommaire

1- Problématique
2- Modèle
3- Simulation
4- Expérience

La marche bipède : une tâche complexe à recréer

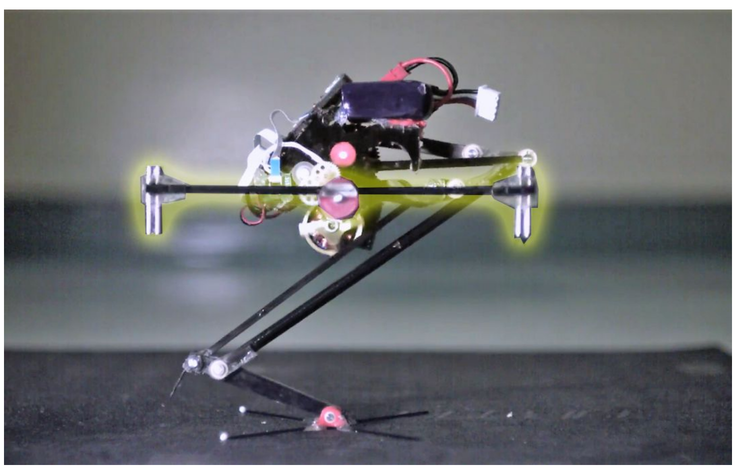


DARPA Robotic Challenge, 2015

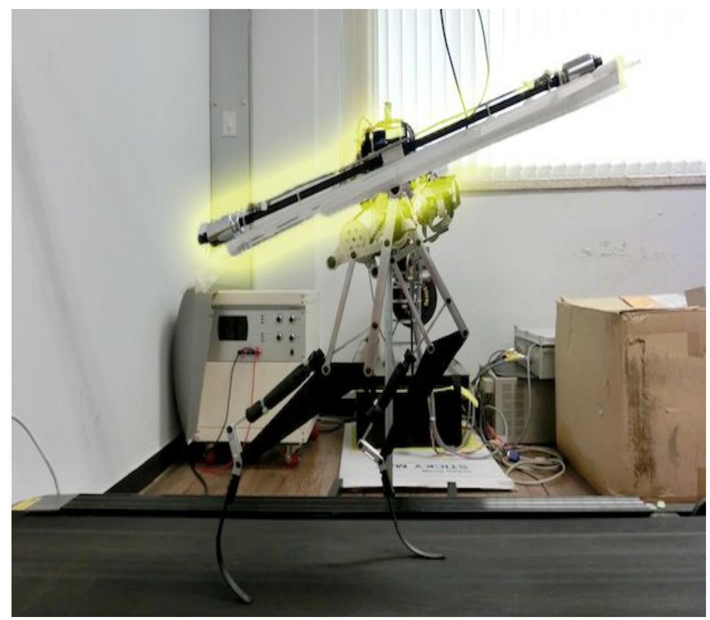
Utilisation d’une queue ?

Handle de Boston Dynamics

Dromaeosauridae

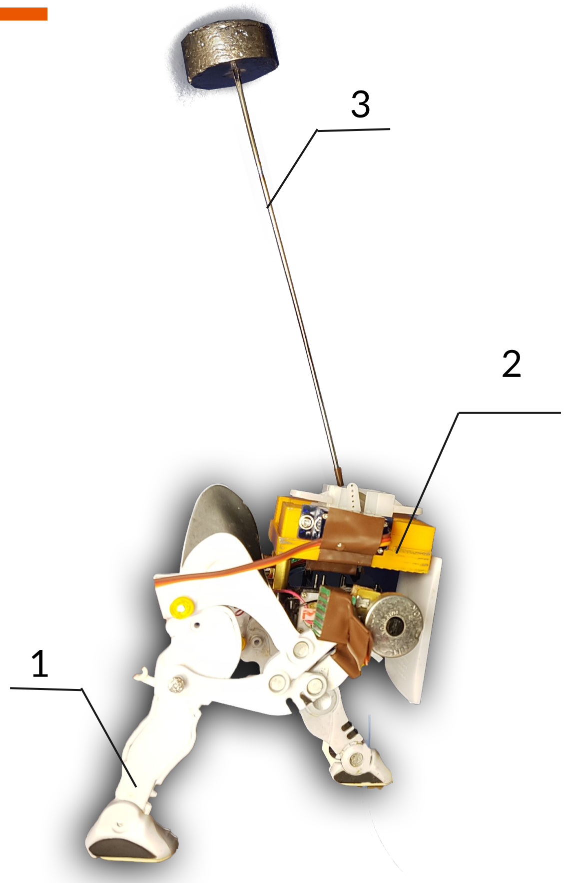


Robot SALTO de l’UC Berkeley

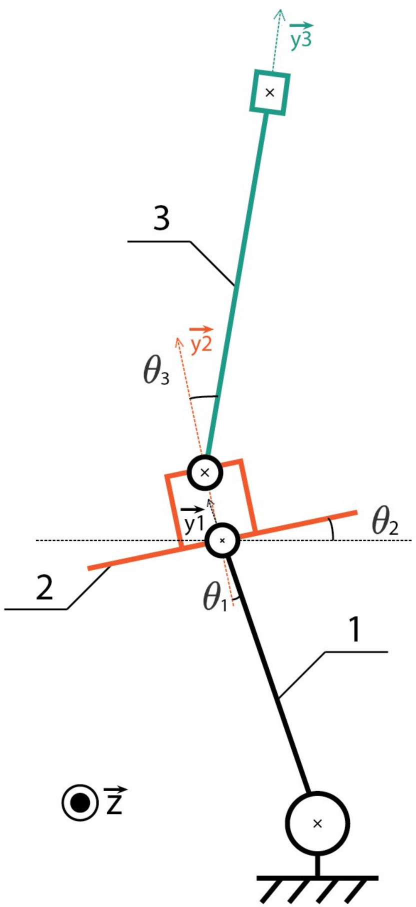


Raptor du KAIST

Modèle

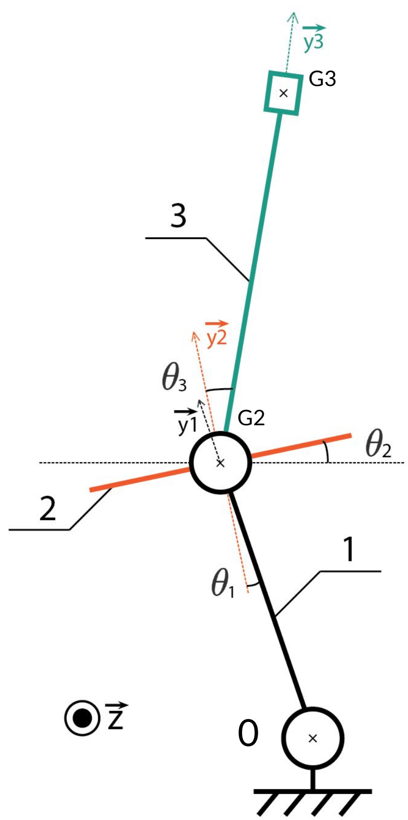


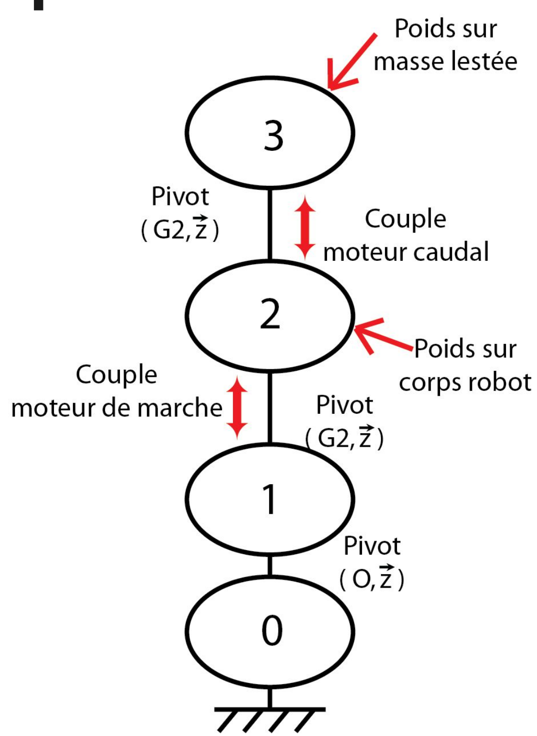
Dispositif expérimental mis en oeuvre

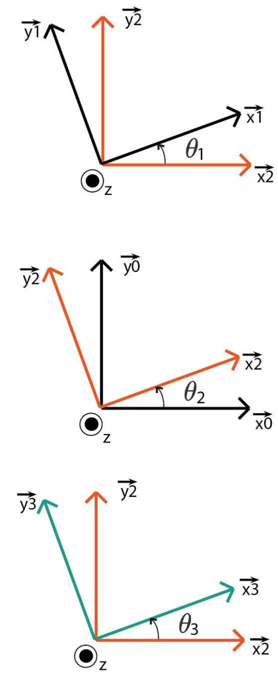


Modèle du dispositif expérimental

Graphe des liaisons et des actions mécaniques

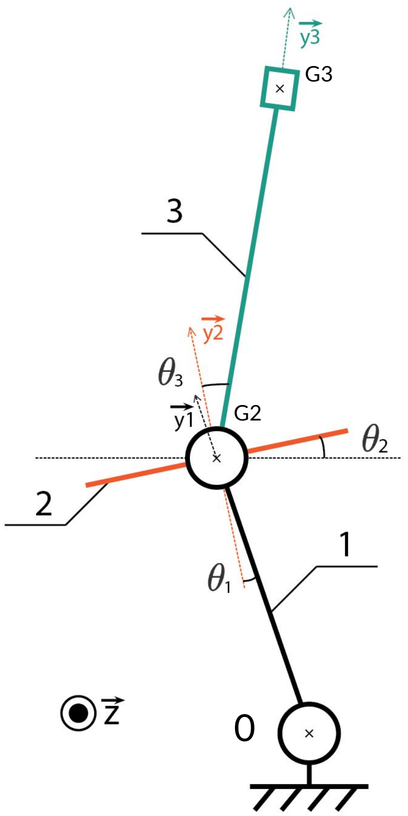






1. Jambe robot
2. Corps robot
3. Queue lestée

Hypothèses simplificatrices

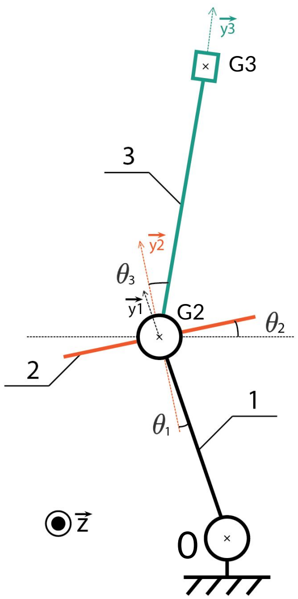


1. La masse des jambes du robot est négligée
2. L'angle θ1 est imposé par une sollicitation externe
3. Les liaisons sont énergétiquement parfaites

<u>But</u> : imposer un angle θ_2 nul

Équations

$$\ddot{\theta}_3(\mathrm{a}+\mathrm{c}\cos(\theta_3+\theta_1))+\ddot{\theta}_2\,\mathrm{h}+\ddot{\theta}_1(\mathrm{k}\text{-}\mathrm{c}\cos(\theta_3+\theta_1))+(\dot{\theta}_1+\dot{\theta}_1)^2\,c\,sin(\theta_3+\theta_1)-(\dot{\theta}_3+\dot{\theta}_2)^2\,a\,sin(\theta_3+\theta_1)=b\,sin(\theta_2+\theta_1)+f\,sin(\theta_3+\theta_2)(2)$$

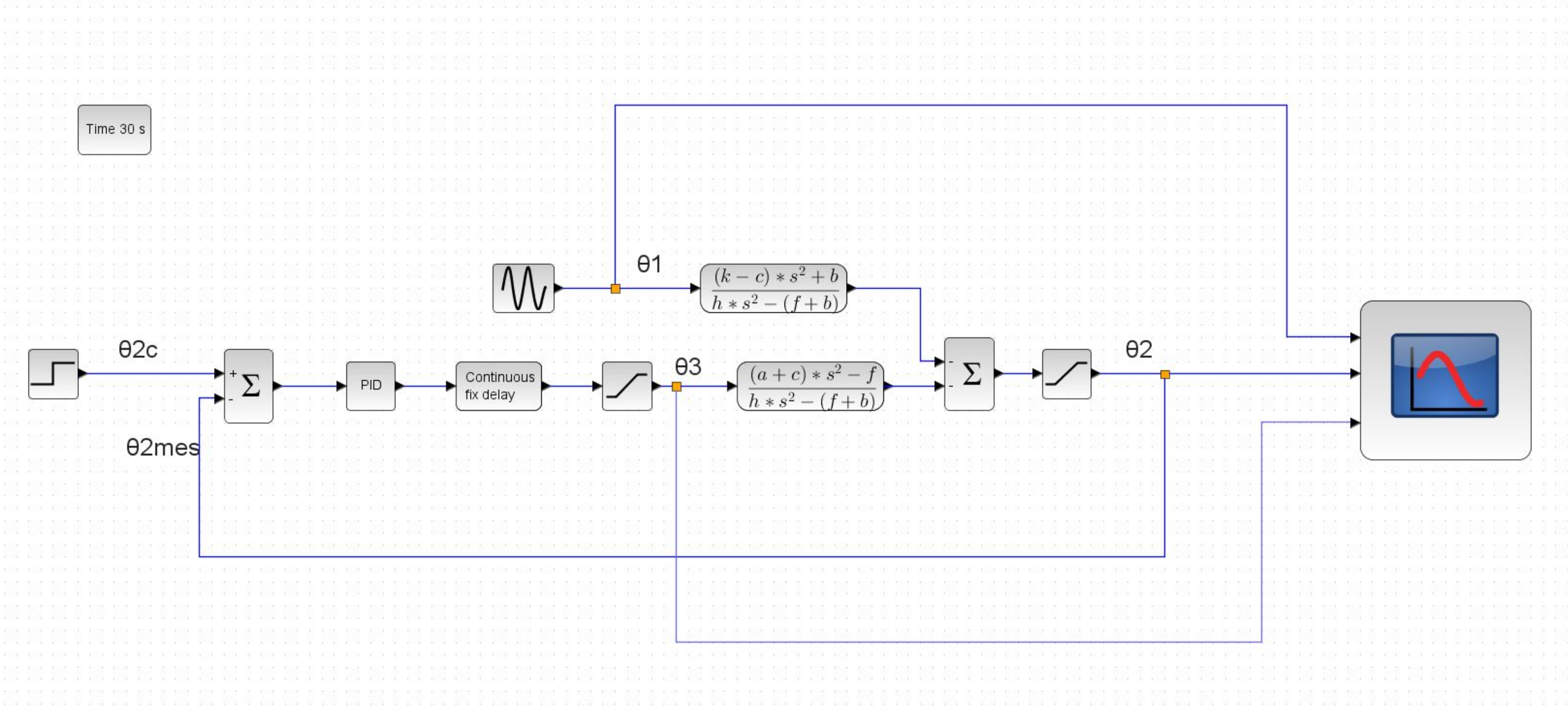


$$\ddot{\theta}_3(\mathrm{a}+\mathrm{c})+\ddot{\theta}_2\,\mathrm{h}+\ddot{\theta}_1(\mathrm{k}\text{-}\mathrm{c})=\mathrm{b}\,(\theta_2+\theta_1)+f\,(\theta_3+\theta_2)$$

$\mathrm{a}=m_3\,{L_3}^2$
$b=\mathrm{m}_2\,L_1\,\mathrm{g}+\,m_3\,L_1\,\mathrm{g}$
$\mathrm{c}=m_3\,L_3\,L_1$
$\mathrm{f}=m_3\,L_3\,\mathrm{g}$
$\mathrm{k}=m_3\,{L_1}^2$
$h=\mathrm{m}_3\,({L_3}^2+{\mathrm{L}_1}^2)+\mathrm{J}\Delta_2+m_2\,{L_1}^2$

$$\theta_3(\mathrm{p})\,(\mathrm{p}^2\,(a+c)-f)+\theta_1(\mathrm{p})\,(\mathrm{p}^2\,(k-c)-b)=-\theta_2(\mathrm{p})\,(\mathrm{p}^2\,h-(f+b))$$

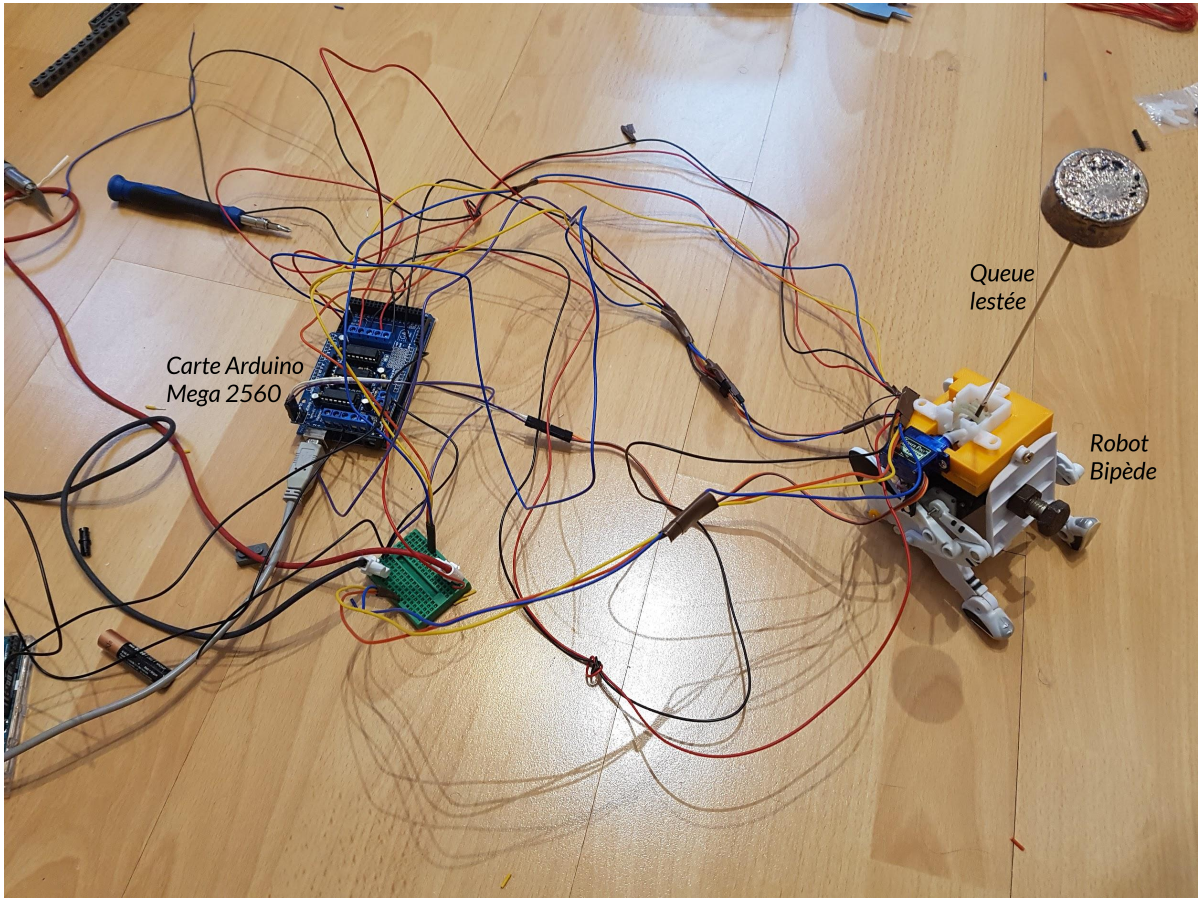
Simulation



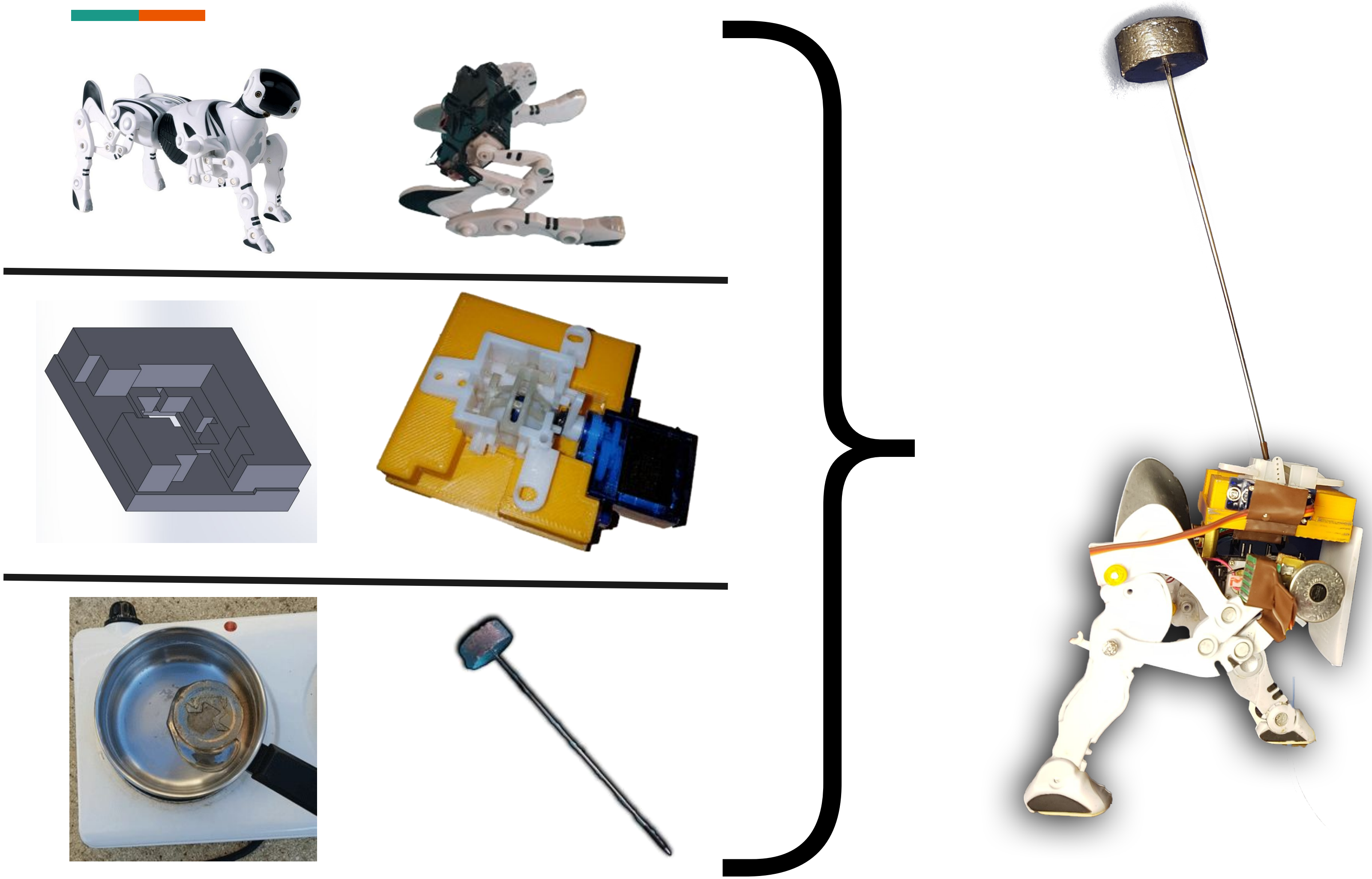
Time 30 s
θ1
$\frac{(k-c)*s^2+b}{h*s^2-(f+b)}$
θ2c
θ2
θ3
PID
Continuous
fix delay
$\frac{(a+c)*s^2-f}{h*s^2-(f+b)}$
θ2mes

Amplitude θ1 = 30 °

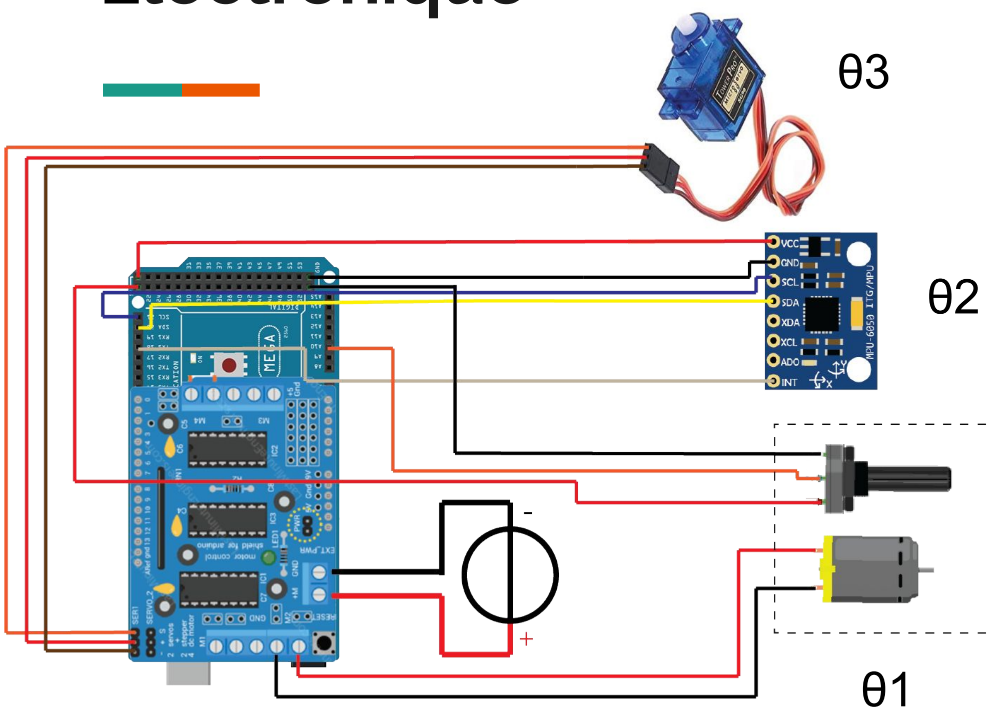
Expérience

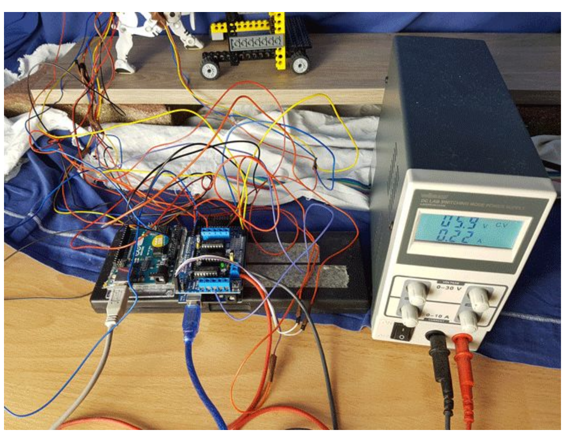


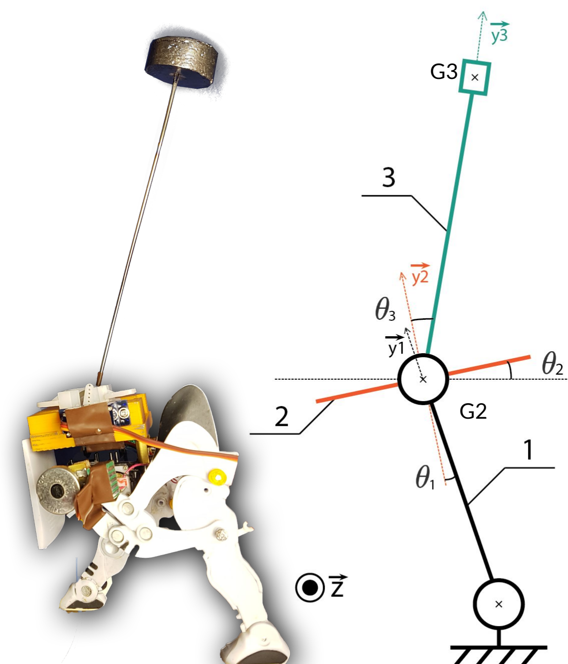
Conception du robot



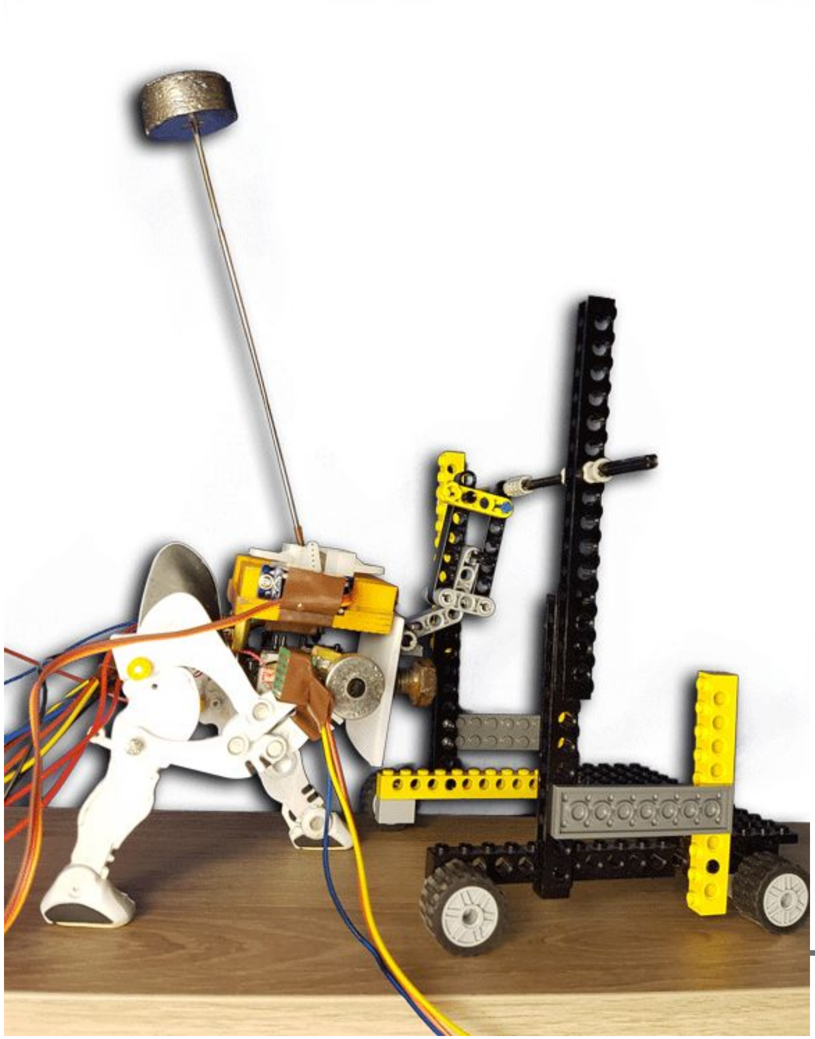
Électronique

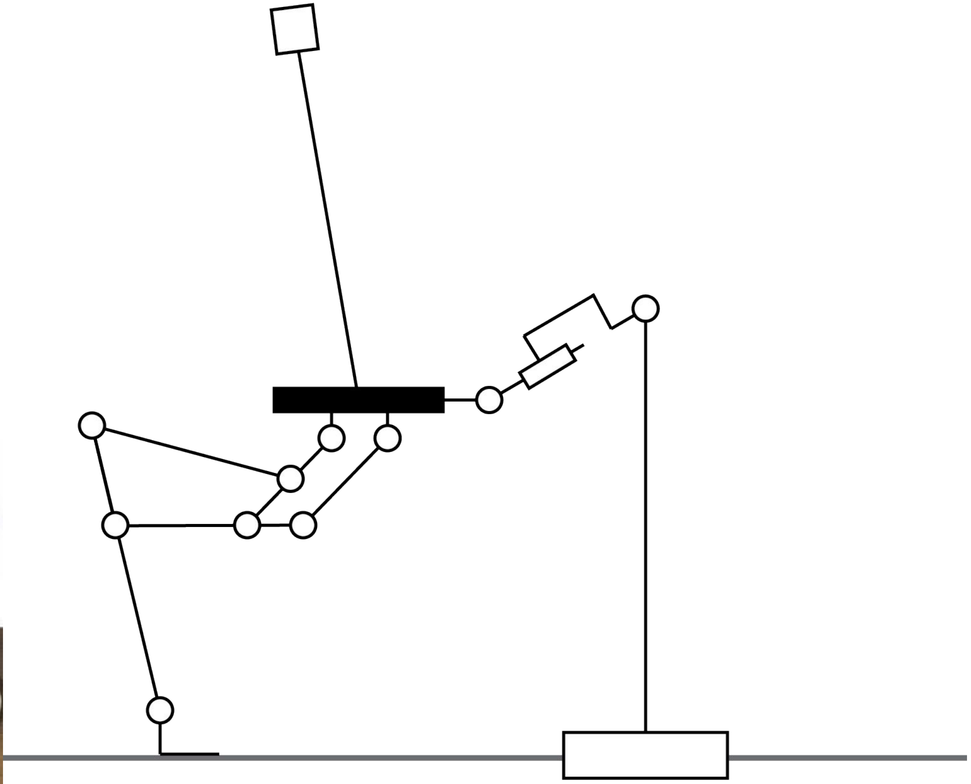






Dispositif expérimental





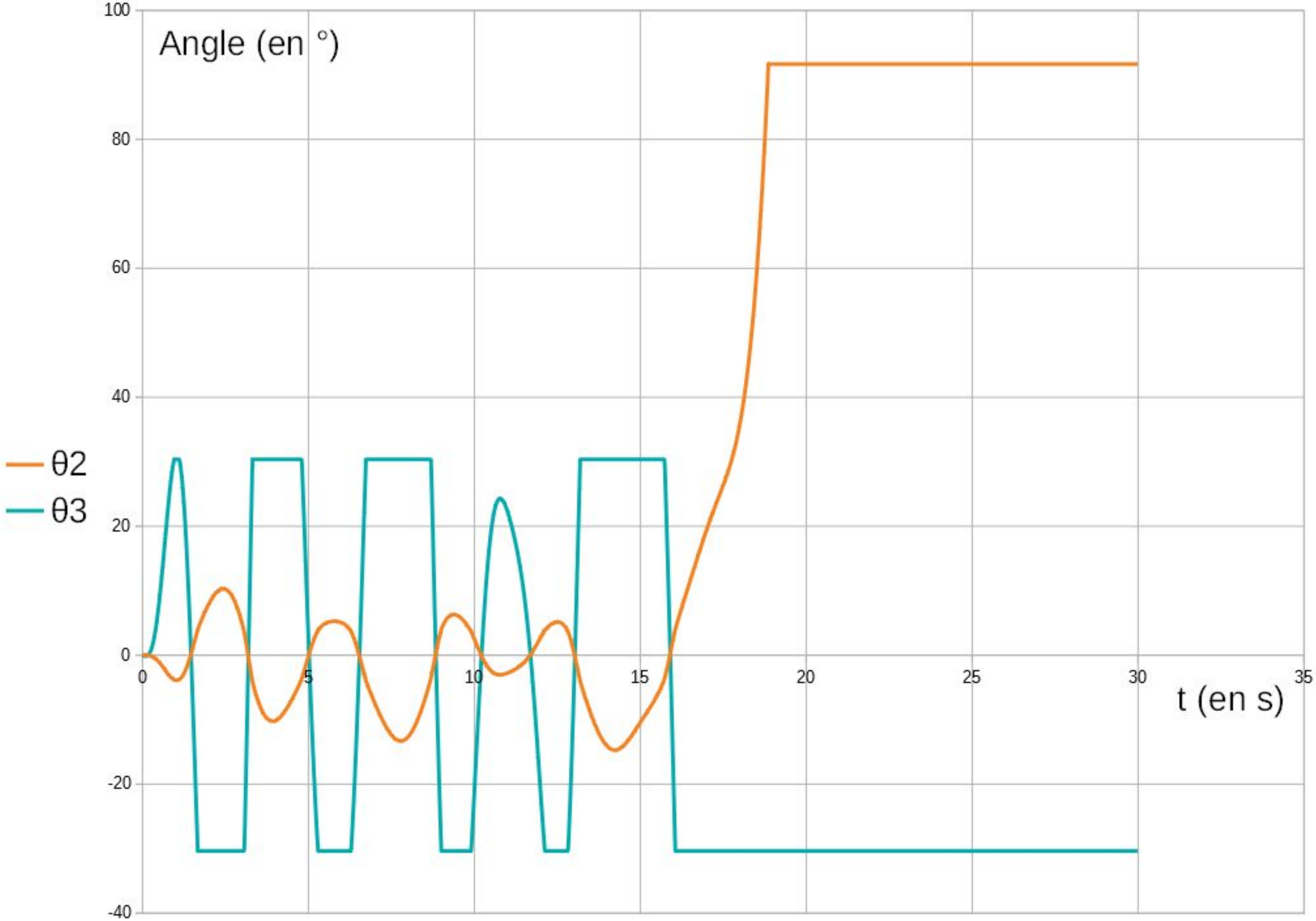
Servomoteur

Cmax=1.8 kgf·cm soit 0.18 Nm

m3=111g

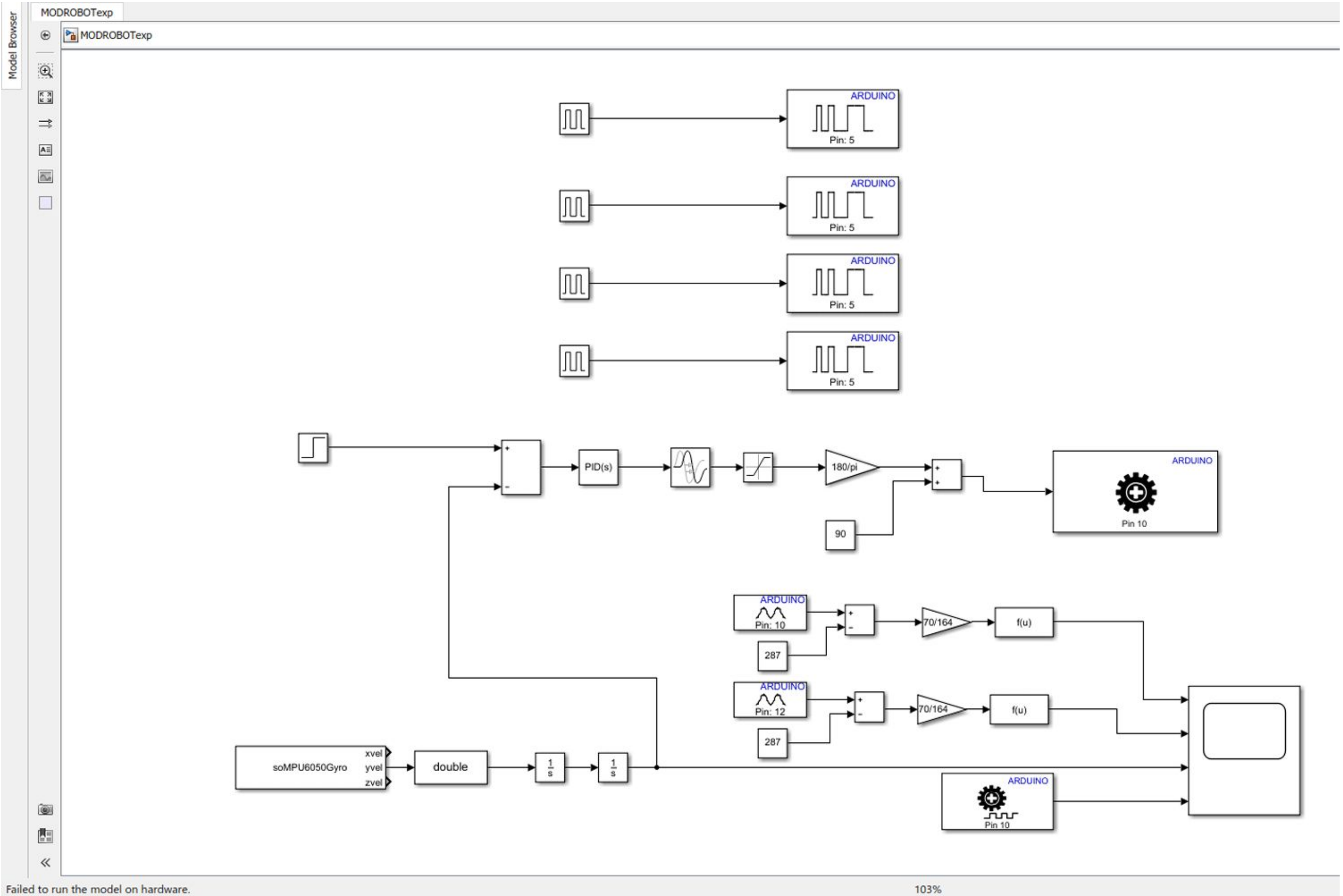
m3=88g

Amplitude θ1 = 30 °

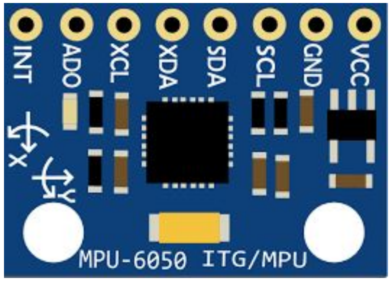


Premier programme de contrôle

Modèle Matlab

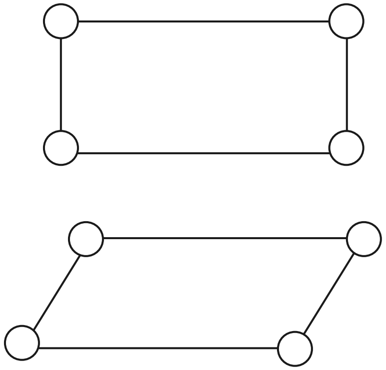


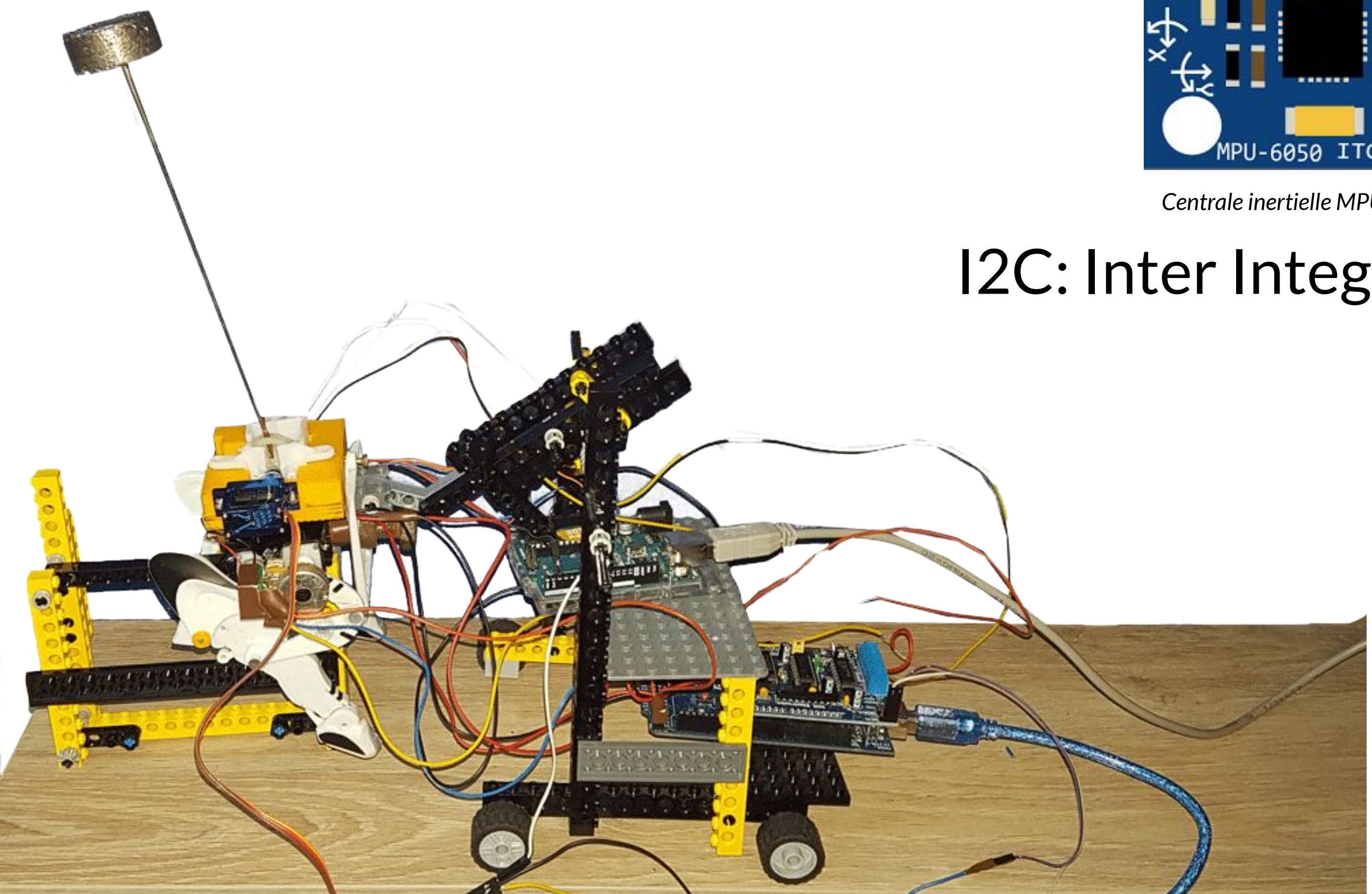
Second dispositif expérimental



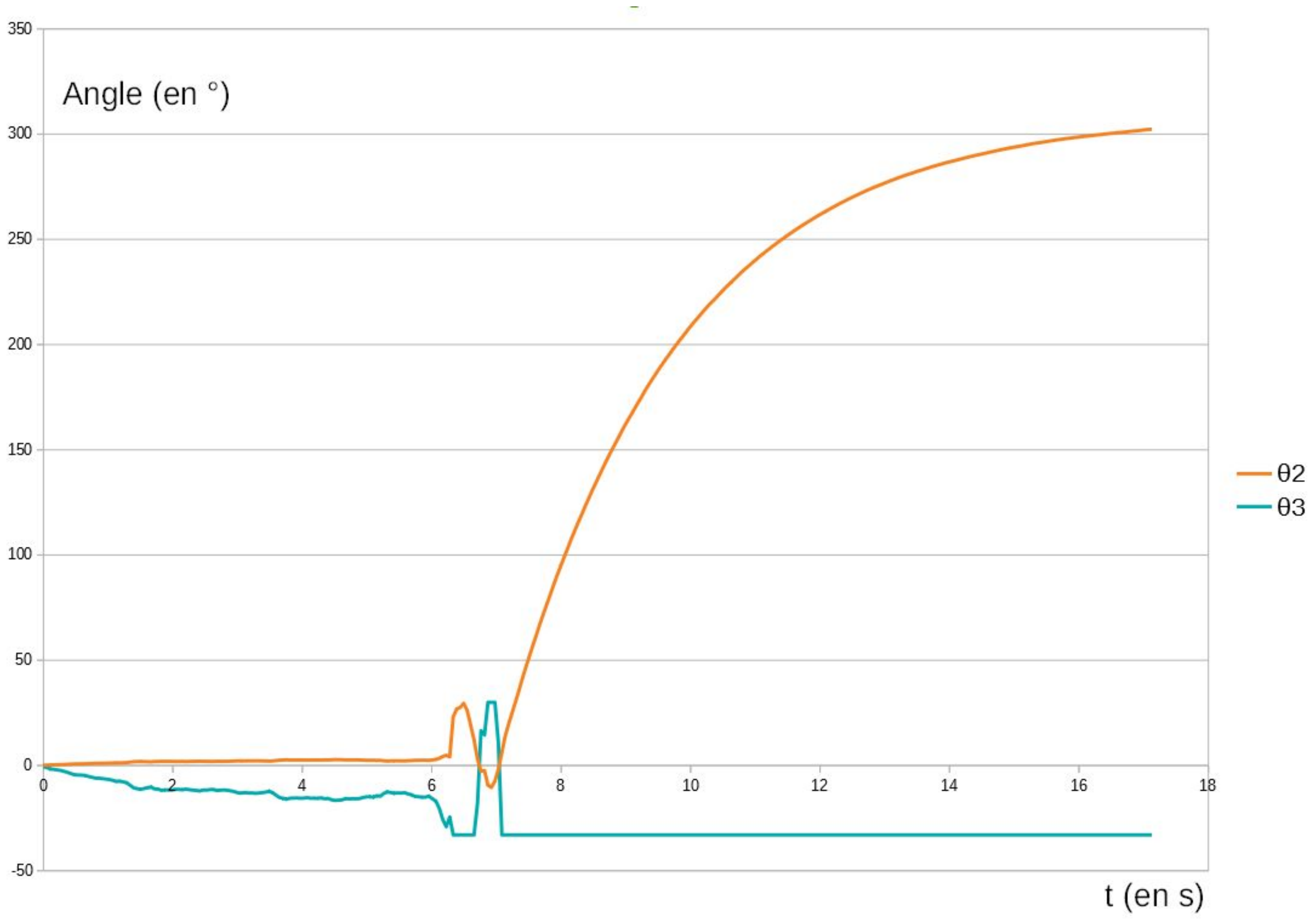
Centrale inertielle MPU 6050

I2C: Inter Integrated Circuit





Amplitude θ1 = 30 °



Limites de l'expérience

Annexe: programmes arduino

Programme_marche

```
// Adafruit Motor shield library
// copyright Adafruit Industries LLC, 2009
// this code is public domain, enjoy!

#include <AFMotor.h>

AF_DCMotor gauchemotor(3,MOTOR34_1KHZ);
AF_DCMotor droitemotor(4,MOTOR34_8KHZ);
int potpind = 10;
int potping = 8;

void setup() {
  gauchemotor.setSpeed(255);
  droitemotor.setSpeed(255);

  gauchemotor.run(FORWARD);
  droitemotor.run(BACKWARD);

}

void loop() {
  if (analogRead(potpind)>740) {
    droitemotor.run(BACKWARD);
  }
  if (analogRead(potping)>518) {
    gauchemotor.run(BACKWARD);
  }
  if (analogRead(potping)<303) {
    droitemotor.run(FORWARD);
  }
  if (analogRead(potpind)<525) {
    droitemotor.run(FORWARD);
  }
  delay(50);
}
```

programme_queue

```
#include <MPU6050_tockn.h>
#include <Wire.h>
#include <Servo.h>

MPU6050 mpu6050(Wire);
Servo myservo;
float thetazero=0;

void setup() {
  Serial.begin(9600);
  myservo.attach(9);
  Wire.begin();
  mpu6050.begin();
  mpu6050.calcGyroOffsets(true);
  mpu6050.update();
  thetazero = mpu6050.getAngleY();
}

void loop() {
  mpu6050.update();
  float thetadeux=-mpu6050.getAngleY()+ thetazero;
  myservo.write(constrain(6*(thetadeux) + 97, 67, 130));
  delay(50);
  Serial.print(millis());
  Serial.print(";");
  Serial.print(thetadeux);
  Serial.print(";");
  Serial.println(constrain(6*(thetadeux) + 97, 67, 130)-97);

}
```