

Asservissement de vitesse et réversibilité énergétique d'une chaîne de puissance de trottinette électrique



TIPE 2025/2026

Code CNC : AG020T

Réalisé par : Soufiane SEBAAOUI

Encadré par : M. GUENZAOUZ Hassan

Plan de la présentation :

01

Introduction

A propos de la trottinette
électrique

02

Problématique

03

Objectifs

- 1/ Identifier les principaux composants de la trottinette électrique et représenter son fonctionnement par un schéma simplifié
- 2/ Concevoir et régler une boucle d'asservissement de vitesse
- 3/ Illustration du principe du freinage régénératif (KERS)

04

Conclusion

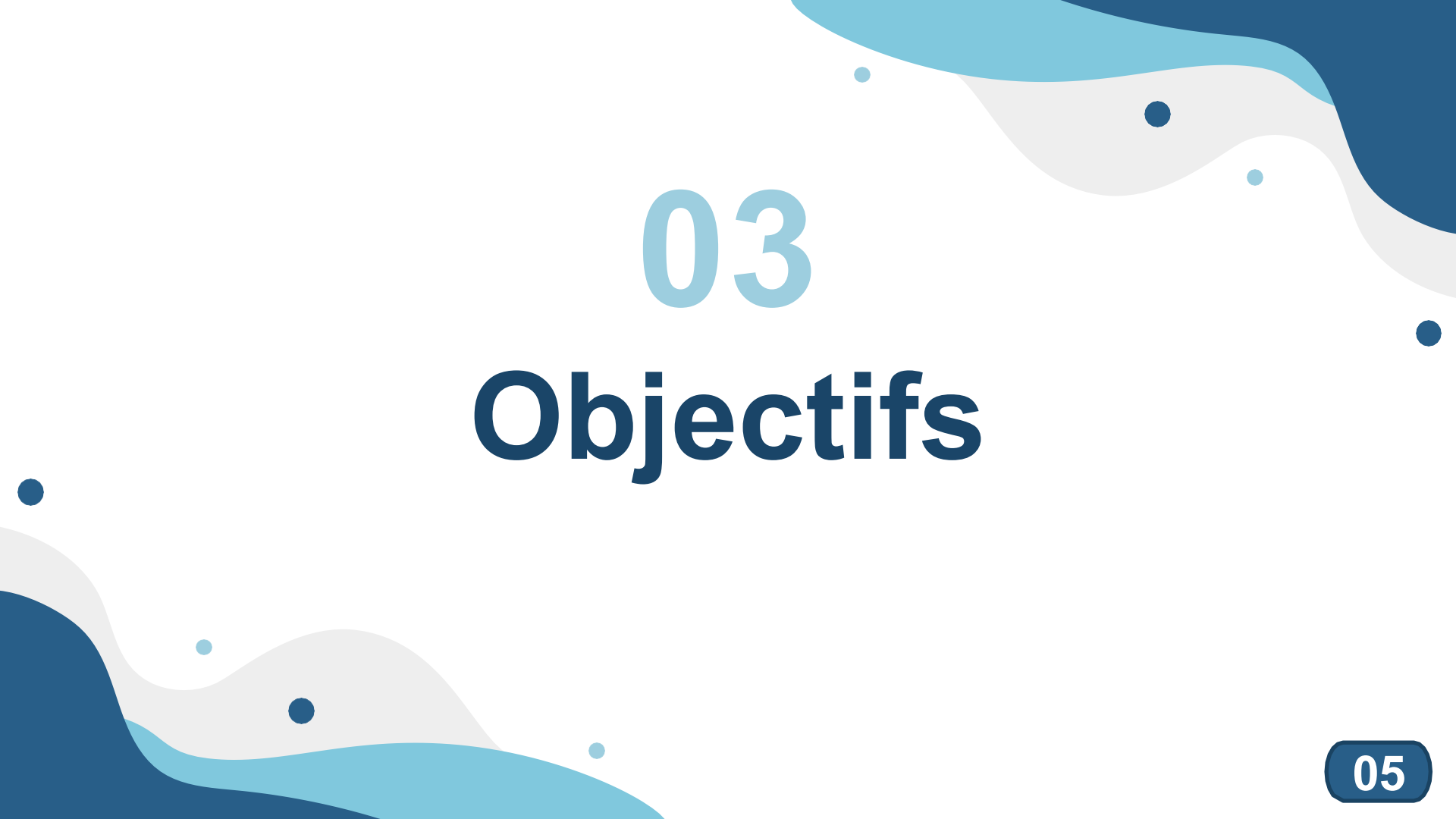
Introduction

Pourquoi la trottinette électrique ?



Problématique Retenue :

Comment concevoir un asservissement de vitesse robuste aux perturbations tout en comprenant le principe de freinage régénératif dans une trottinette électrique ?



03

Objectifs

Objectif 01

Identifier les principaux composants de la trottinette électrique et représenter son fonctionnement par un schéma simplifié

Vue globale sur la trottinette :

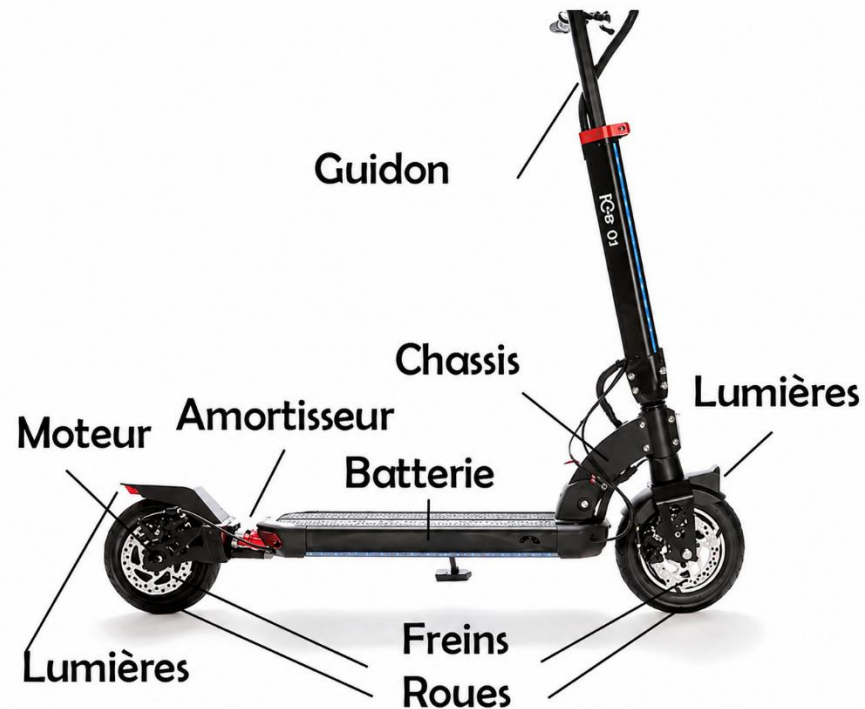


Diagramme des cas d'utilisation

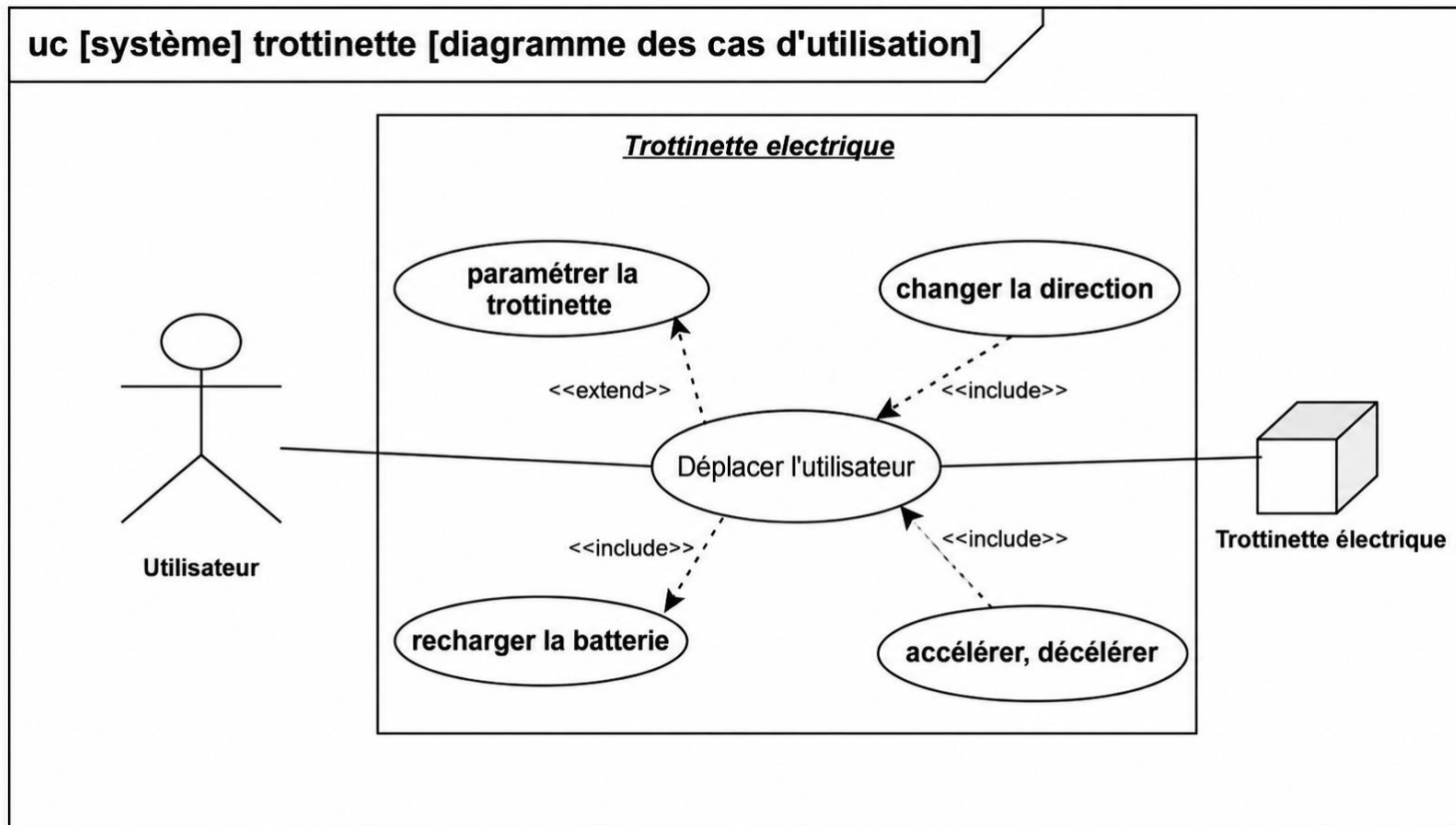


Diagramme des exigences

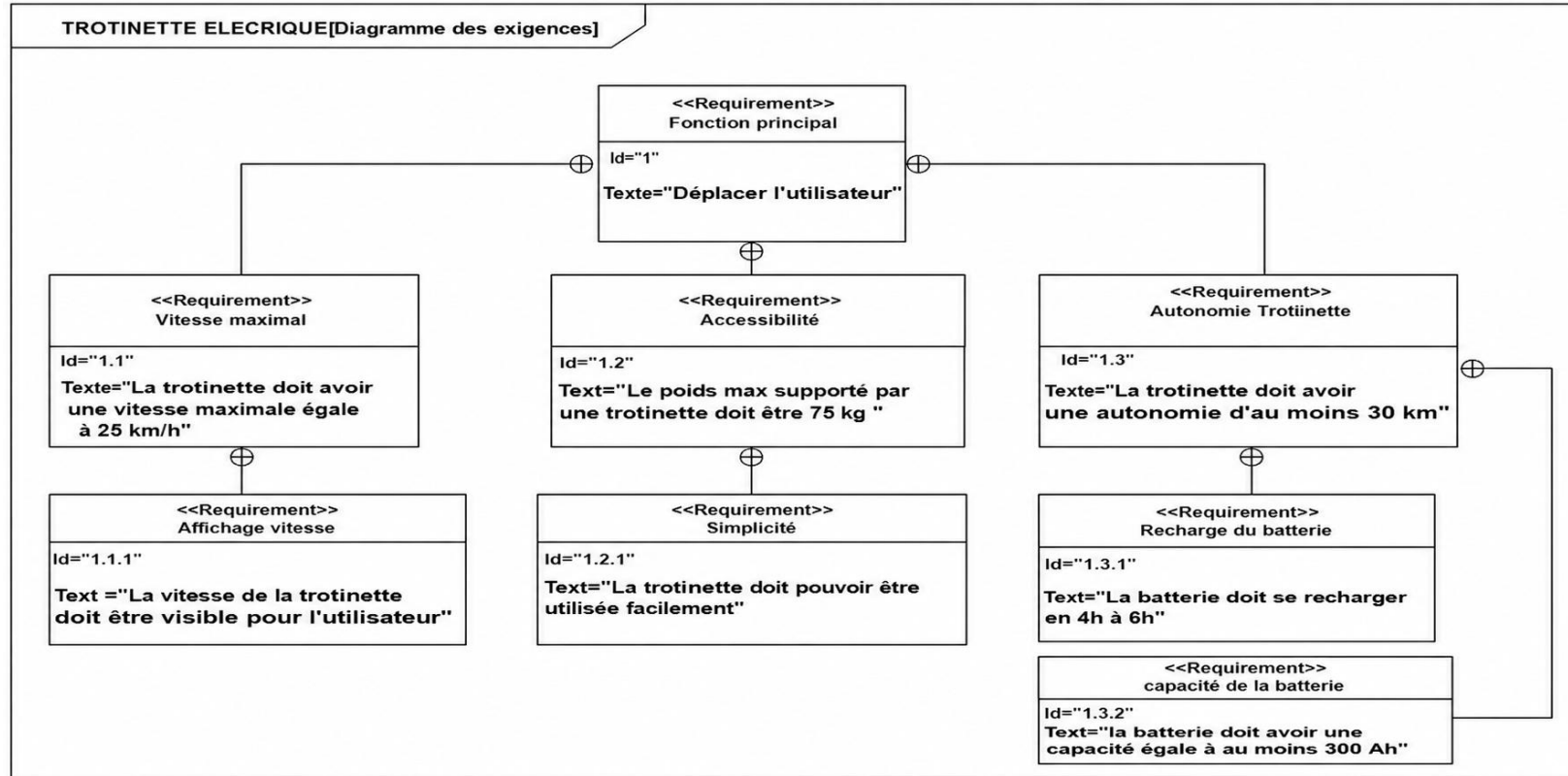
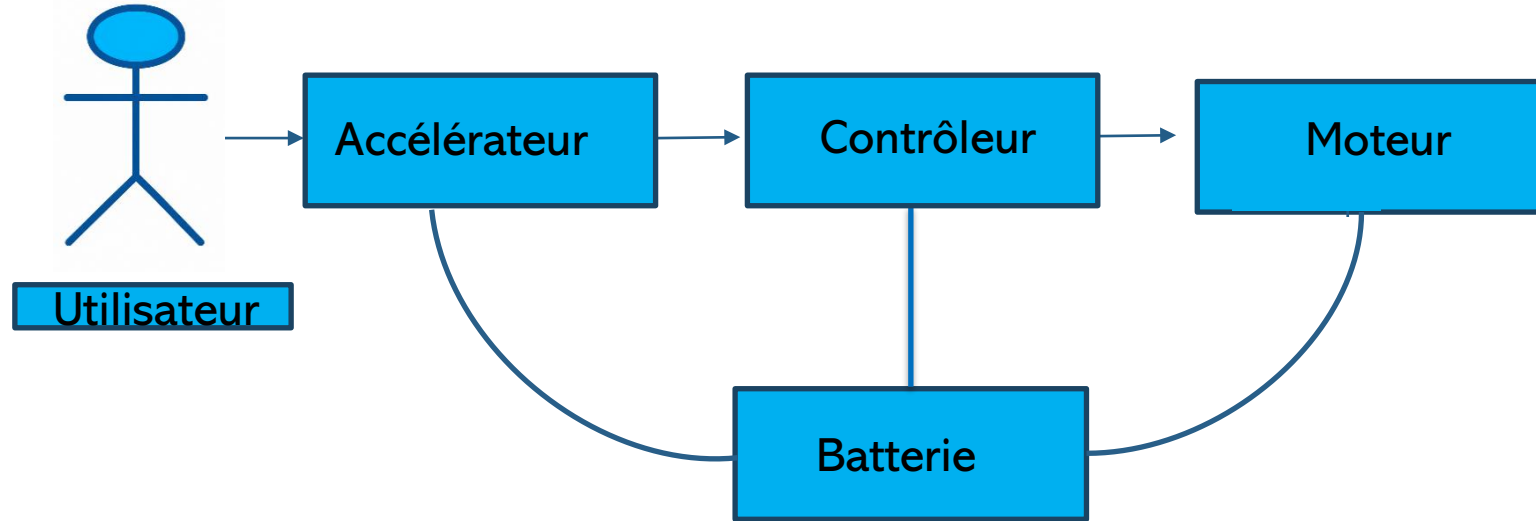


Schéma de fonctionnement :



L'accélérateur

Détails sur le produit

Dimensions du produit (L x l x h) : 13 x 8 x 3 cm

Fabricant : ZAWELIYO

ASIN : B0D4228WXR

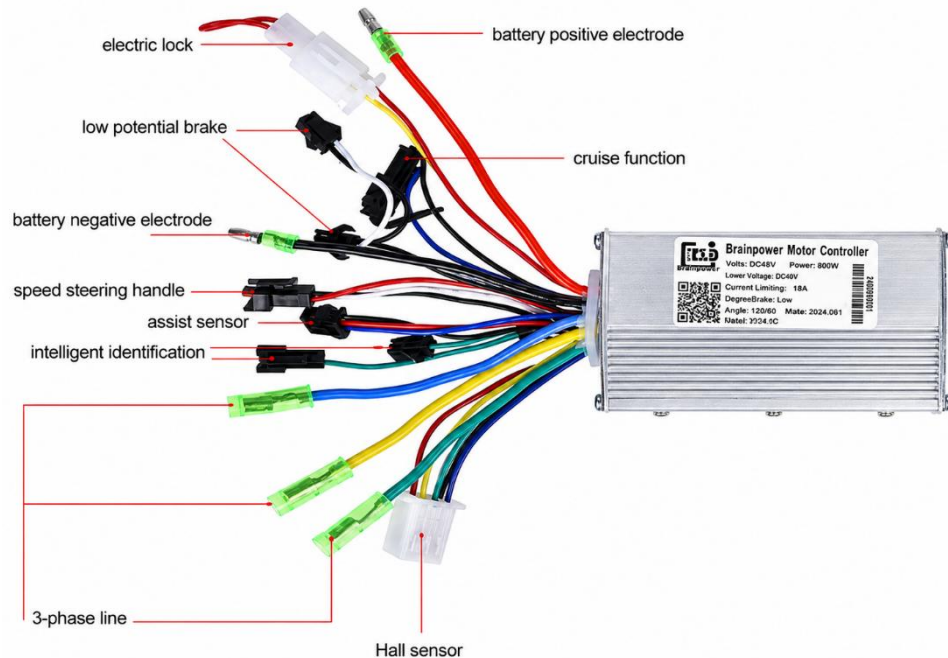
Numéro du modèle de l'article : ZAWELIYOthdzuvey9m711



Le contrôleur

PRINCIPALES CARACTÉRISTIQUES

- Matériel: alliage D'aluminium
- Tension de fonctionnement nominale: 36V
- Puissance nominale: 250W
- Courant: 13A
- Protection basse tension: 30V
- Angle de Phase de moteur approprié: 60 / 120 degrés
- Convient pour: vélo électrique, scooter, etc.
- Taille: Env. 8.8x5.2x3cm / 3.46x2.05x1.18 pouces
- Poids: Env. 200g
- Contrôleur Sans Balais



La batterie

- Pour le choix de la batterie, une batterie de 36 V est indispensable ce qui pose la question de la capacité nécessaire pour répondre l'exigence 1-3

- On se place dans les conditions moyennes ou le moteur va atteindre 200 W et une **vitesse** de 20 km /h. Alors, **l'énergie** qui sera **consommée par km** est :

$$E_{ck} = \frac{P}{v} = \frac{200}{20} = 10 \text{ wh/km}$$

- Alors pour répondre a l'exigence «la trottinette doit avoir une autonomie d'au moins 30 km »

$$E = E_{ck} \times 30 = 10 \times 30 = 300 \text{ wh}$$



Q.W wh'well	Li-ion Battery
Product Name: Lithium-ion Power Battery	
Product Model: 18650 10S3P	
Nominal Voltage: 36V	
Nominal Capacity: 9.6Ah/345.6Wh	
Charge Voltage: 42V	
Production Date: 2023-02-10	
Attention: Can only use a matching charger charge! Do not heat the battery or near the fire source! Do not reverse or short-circuit the negative !	
  CE MSDS UN38.3 MADE IN CHINA	

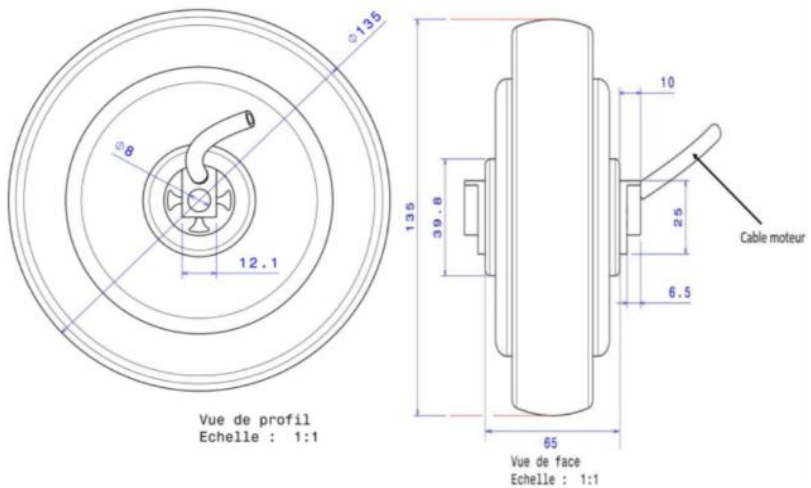
Dimension standard de roues avec un moteur intégré entre 10 et 20 pouces

Ne possède pas de balais ni de collecteur,
Avantages :
• Moins d'usure mécanique (pas de frottements de balais)
• Moins d'entretien
• Plus silencieux
• Rendement plus élevé
• Meilleur couple à bas régime



Donc on va opter pour la roue à moteur intégré du modèle **Brushless direct drive**
Caractéristiques :

Poids	1.9 kg
Puissance max	250W
Entraxe fourche min/max	65mm
Motor kv (RPM / V)	29 RPM/Volt
Axe moteur	Boulon de 8mm
Diamètre extérieur Pneu	134mm
Vitesse max sous 12V	7Km/h
Vitesse max sous 24V	15Km/h
Vitesse max sous 36V	30Km/h
Largeur bande de roulement	39mm



La batterie

- Pour le choix de la batterie, une batterie de 36 V est indispensable ce qui pose la question de la capacité nécessaire pour répondre l'exigence 1-3

- On se place dans les conditions moyennes ou le moteur va atteindre 200 W et une **vitesse** de 20 km /h. Alors, **l'énergie** qui sera **consommée par km** est :

$$E_{ck} = \frac{P}{v} = \frac{200}{20} = 10 \text{ wh/km}$$

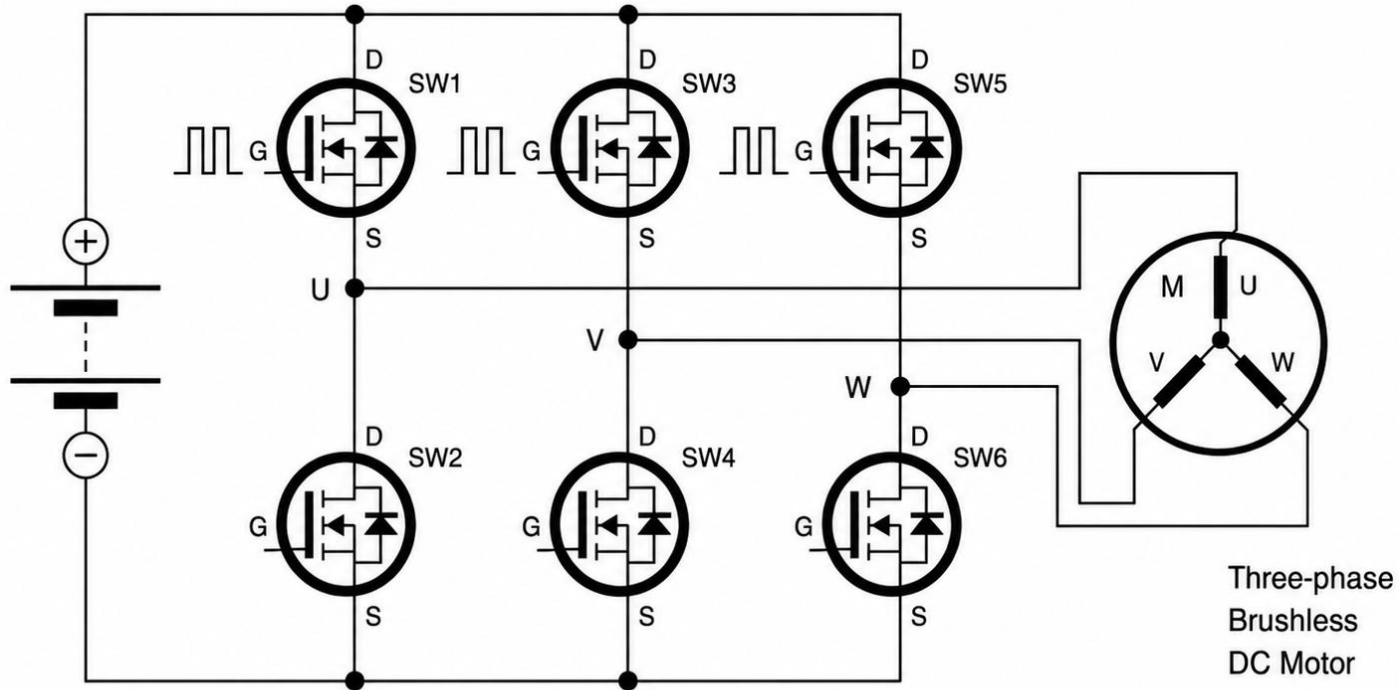
- Alors pour répondre a l'exigence «la trottinette doit avoir une autonomie d'au moins 30 km »

$$E = E_{ck} \times 30 = 10 \times 30 = 300 \text{ wh}$$

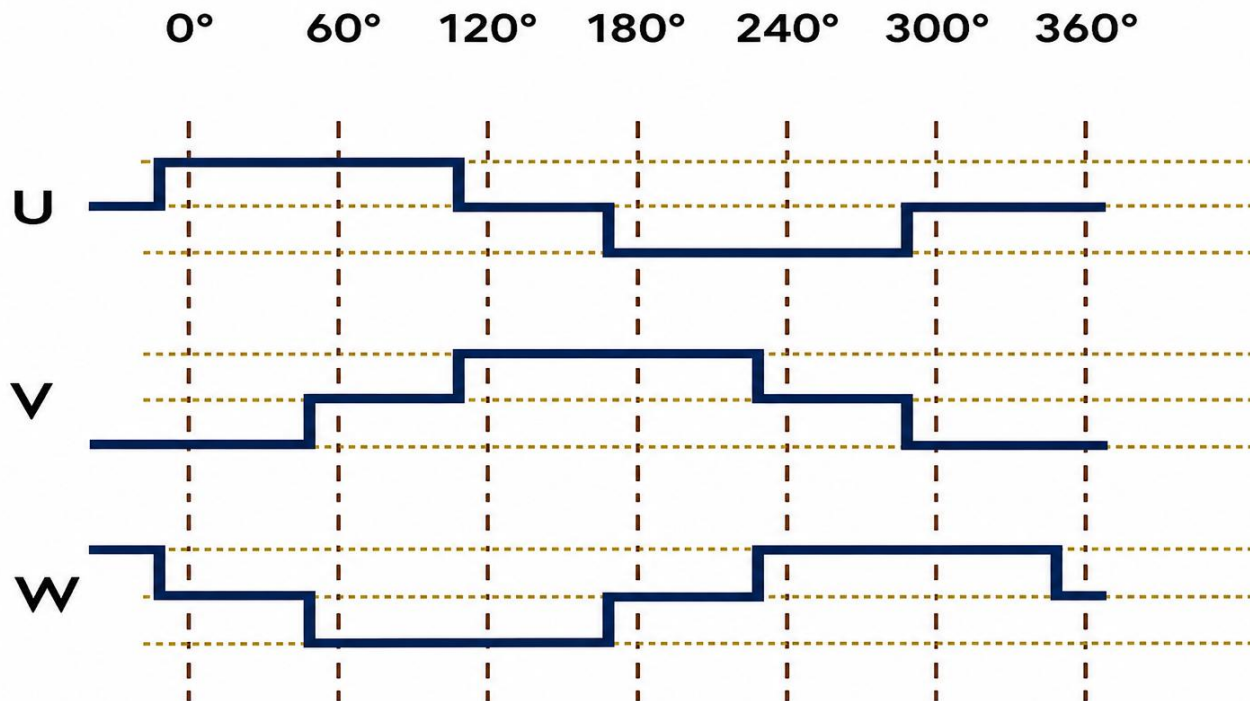


Q.W wh'well	Li-ion Battery
Product Name: Lithium-ion Power Battery	
Product Model: 18650 10S3P	
Nominal Voltage: 36V	
Nominal Capacity: 9.6Ah/345.6Wh	
Charge Voltage: 42V	
Production Date: 2023-02-10	
Attention: Can only use a matching charger charge! Do not heat the battery or near the fire source! Do not reverse or short-circuit the negative !	
  CE MSDS UN38.3 MADE IN CHINA	

Contrôle de la vitesse du moteur



Allure des tensions de phase

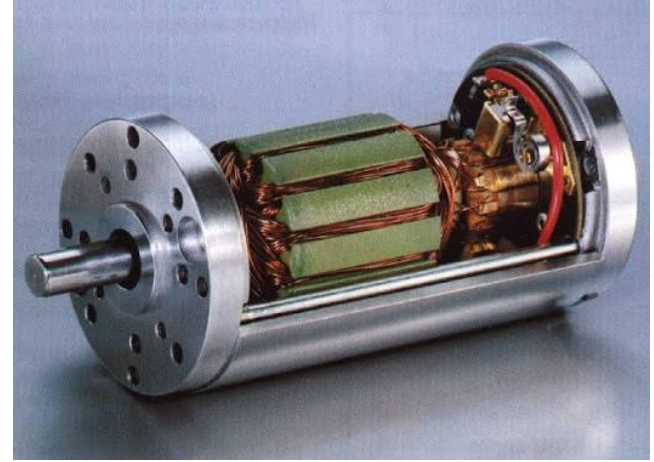
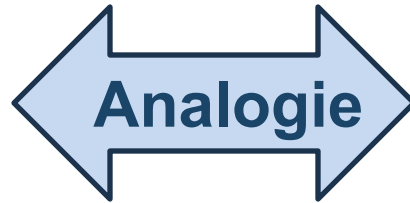


Objectif 02

**Concevoir et régler une
boucle d'asservissement de
vitesse**



Machine synchrone



Machine à courant continu

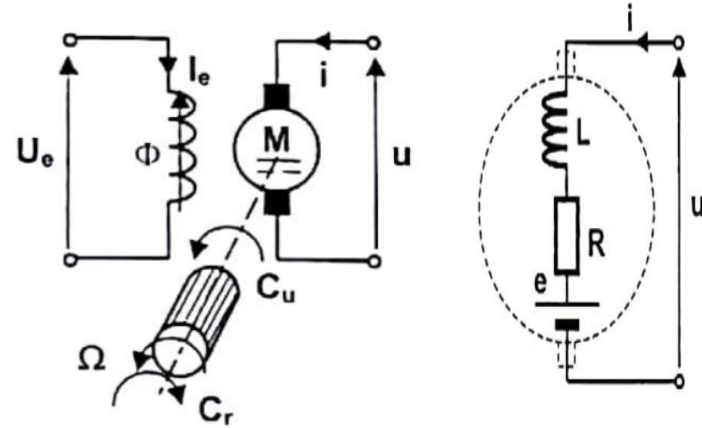
Machine à courant continu:

$$u(t) = Ri(t) + L \frac{di(t)}{dt} + e(t)$$

$$e(t) = K_e \cdot \omega(t)$$

$$C_m(t) = K_c \cdot i(t)$$

$$J \frac{d\omega(t)}{dt} = C_m(t) - C_r(t) - f \cdot \omega(t)$$



$u(t)$: la tension appliquée au moteur (V)

$e(t)$: la force électromotrice (V)

R : la résistance de l'induit (Ω)

L : l'inductance de l'induit (H)

$i(t)$: le courant dans l'induit (A)

$\omega(t)$: la vitesse angulaire du moteur (rad/s)

K_e : constante de fem (V·s/rad)

K_c : constante de couple (N·m/A)

J : moment d'inertie de l'ensemble en rotation (kg·m²)

f : coefficient de frottements visqueux (N·m·s/rad)

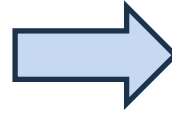
Modélisation de la MCC :

$$U(p) = R \cdot I(p) + L \cdot p \cdot I(p) + E(p)$$

$$C_m(p) = K_c \cdot I(p)$$

$$J \cdot p \cdot \Omega(p) + f \cdot \Omega(p) = C_m(p) - C_r(p)$$

$$E(p) = K_e \cdot \Omega(p)$$

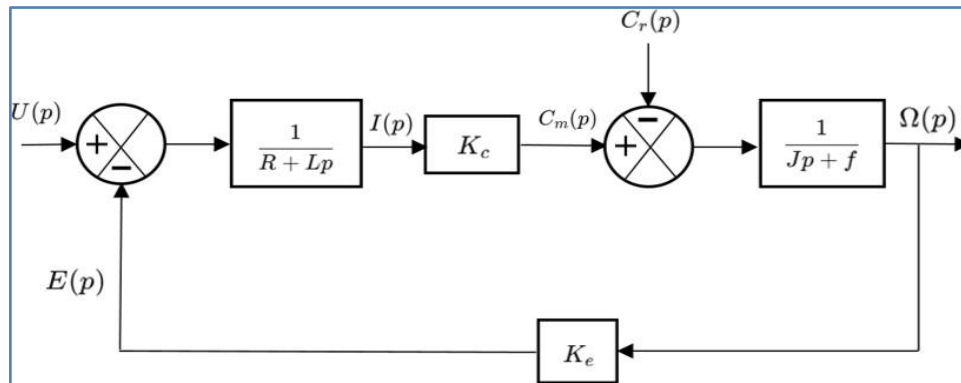


$$\frac{I(p)}{U(p) - E(p)} = \frac{1}{R + Lp}$$

$$\frac{C_m(p)}{I(p)} = K_c$$

$$\frac{\Omega(p)}{C_m(p) - C_r(p)} = \frac{1}{Jp + f}$$

$$\frac{E(p)}{\Omega(p)} = K_e$$



LES COMPOSANTS :



Batterie 9V



Moteur DC 12V



Carte Arduino



Breadboard

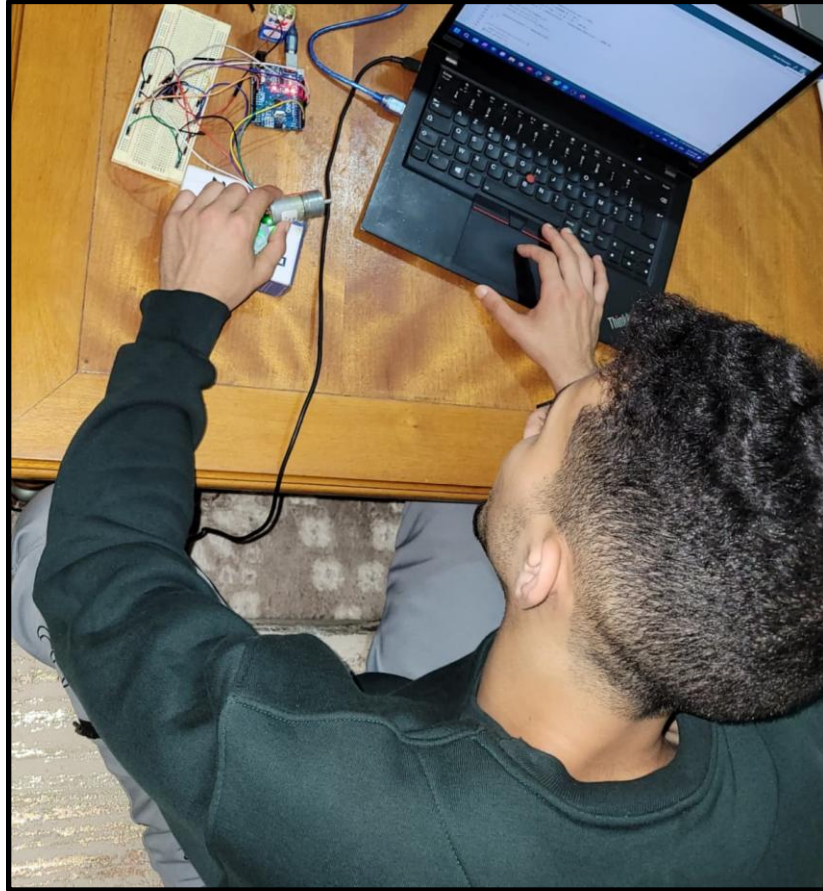
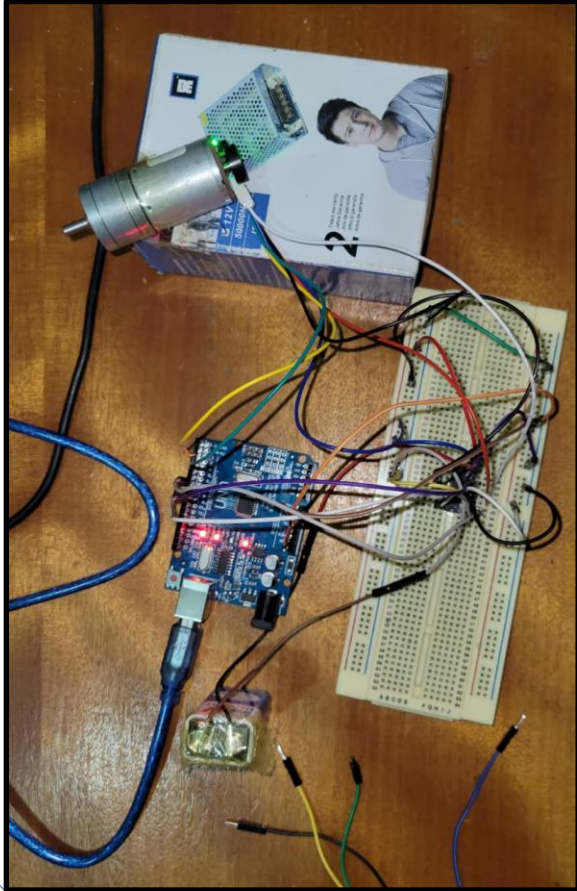


L293D



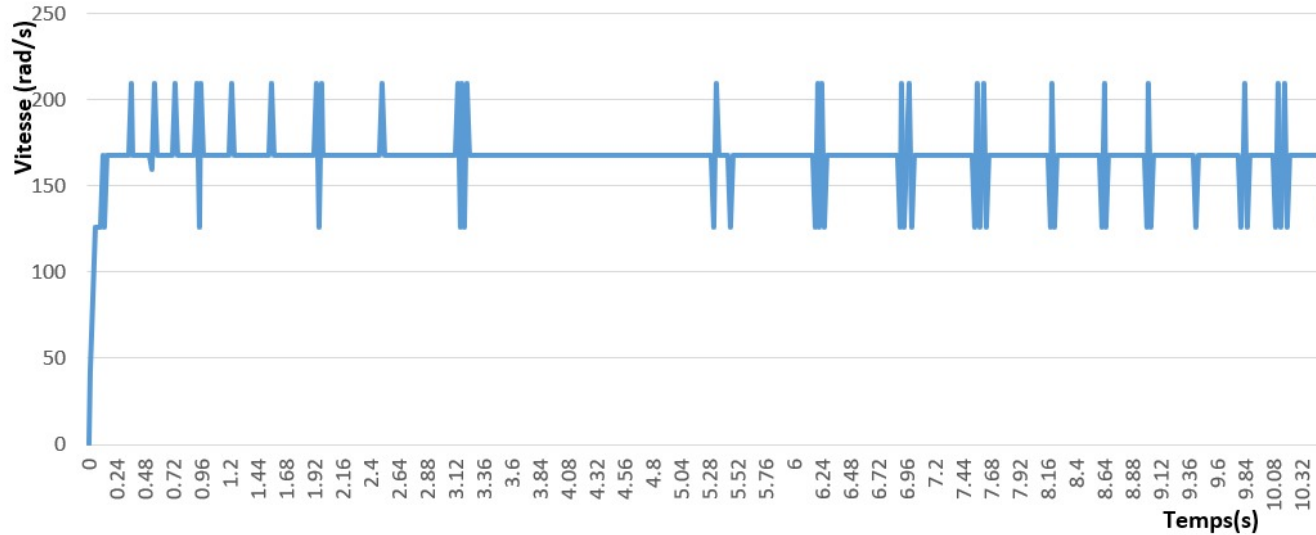
Fils conducteurs

Expérience :



Réponse temporelle du moteur à un échelon de 9V :

Réponse temporelle



Détermination des paramètres de la MCC :

Le système a un comportement du 1^{er} ordre, on détermine ses paramètres par identification, donc :

$$H(p) = \frac{K}{1+\tau p}$$

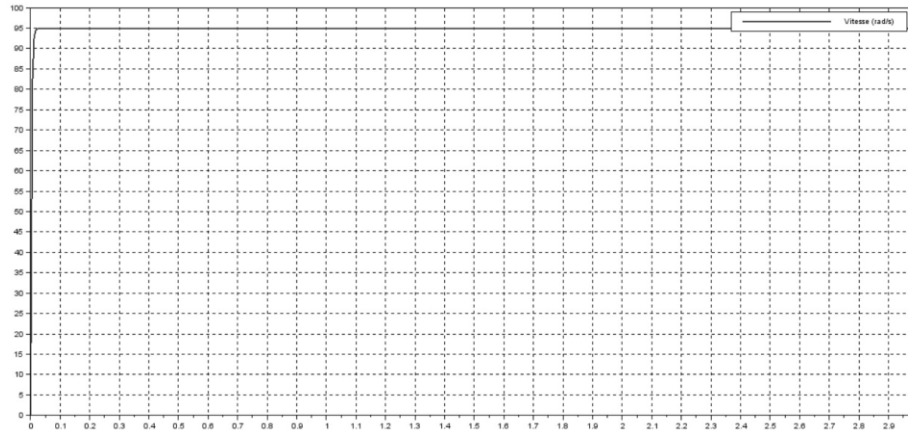
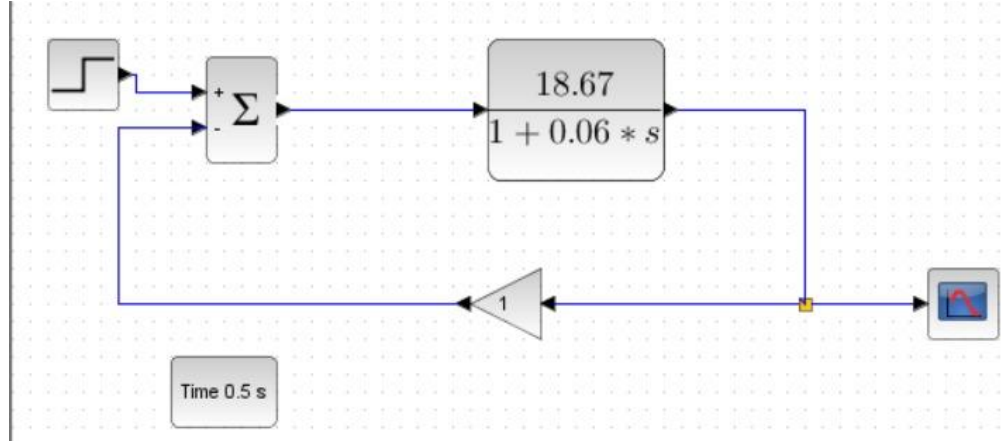
Avec:

$$K = \frac{\Omega_{finale}}{U} = \frac{168}{9} \approx \mathbf{18.67 \text{ rad/s/V}}$$

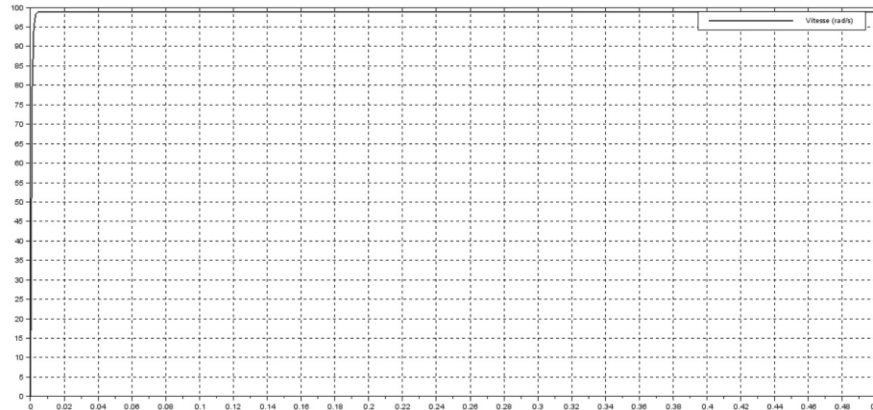
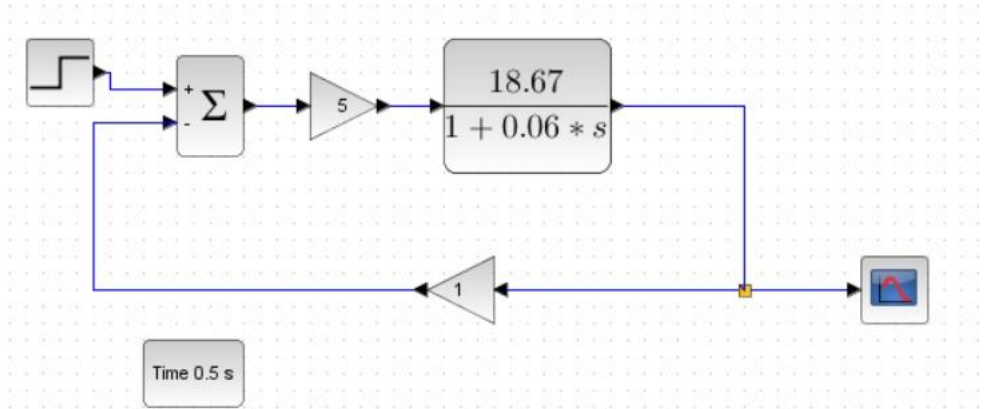
Et :

$$\tau \approx \mathbf{0.06 \text{ s}}$$

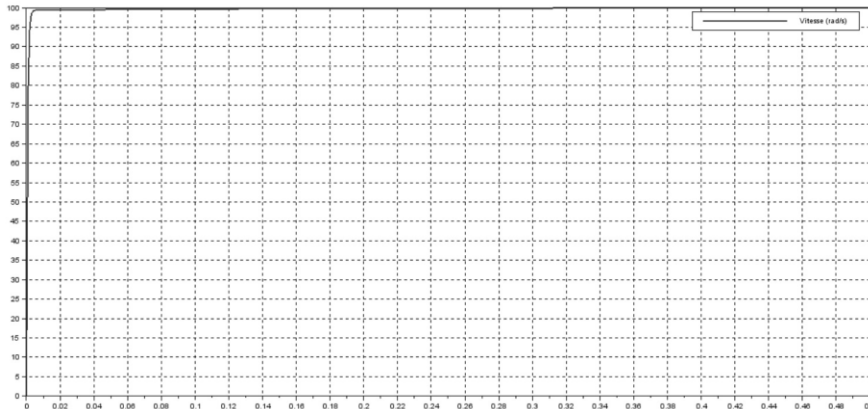
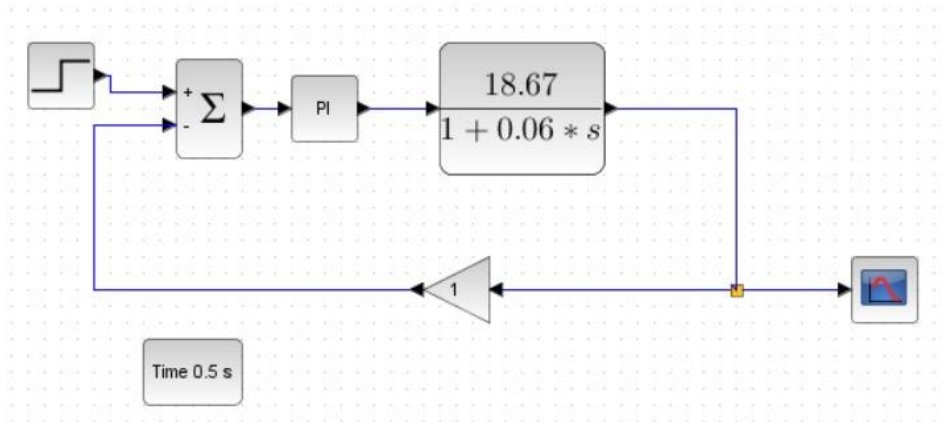
Schéma bloc et simulation de la réponse à une consigne de 100 rad/s :



Test du correcteur proportionnel :

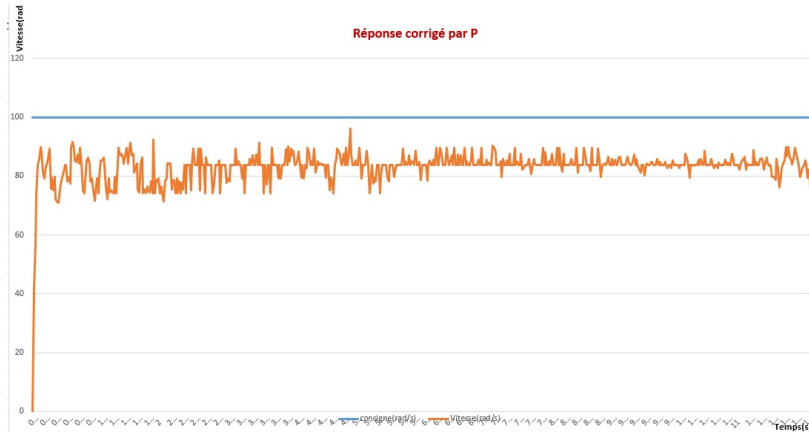


Validation du correcteur PI :

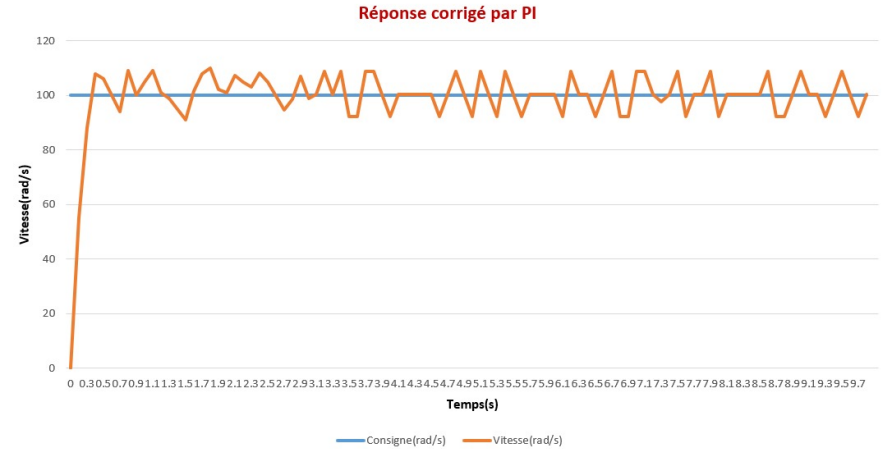


Le choix d'un correcteur PI avec $K_p=5$ et $K_i=35$ permet d'annuler l'erreur statique sans avoir un dépassement.

Expérimentalement :



$K_p=5$



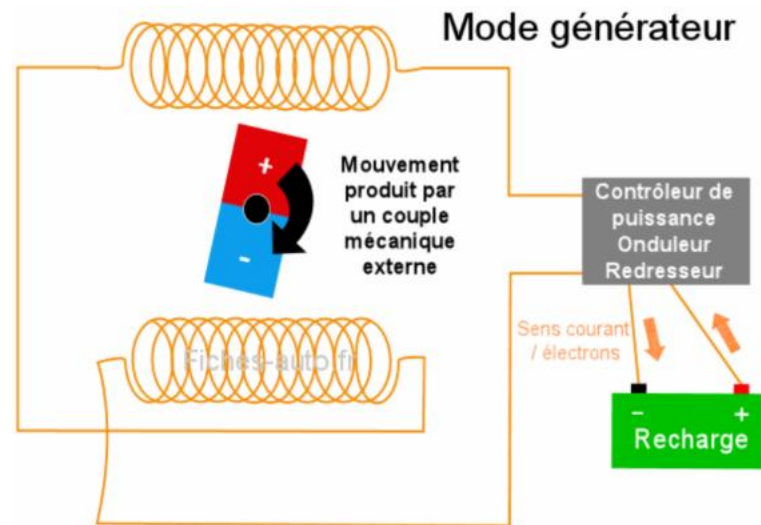
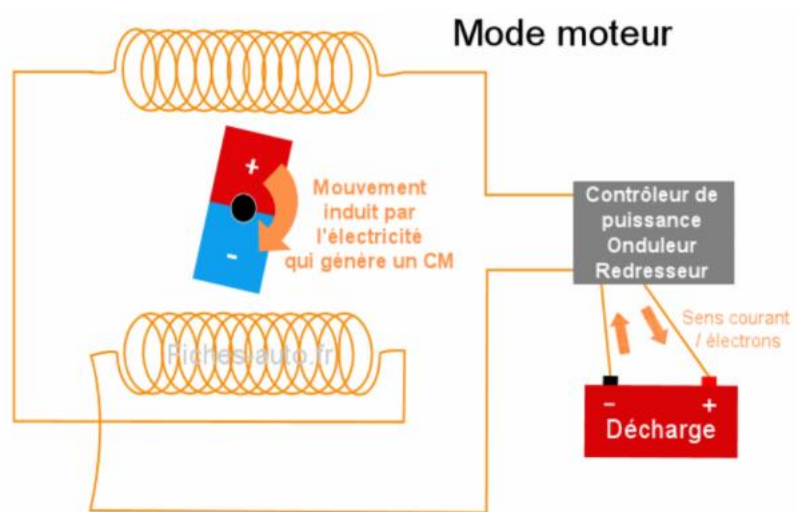
$K_p=1$ et $K_i=5$

Objectif 03

Illustration du principe du freinage régénératif (KERS)

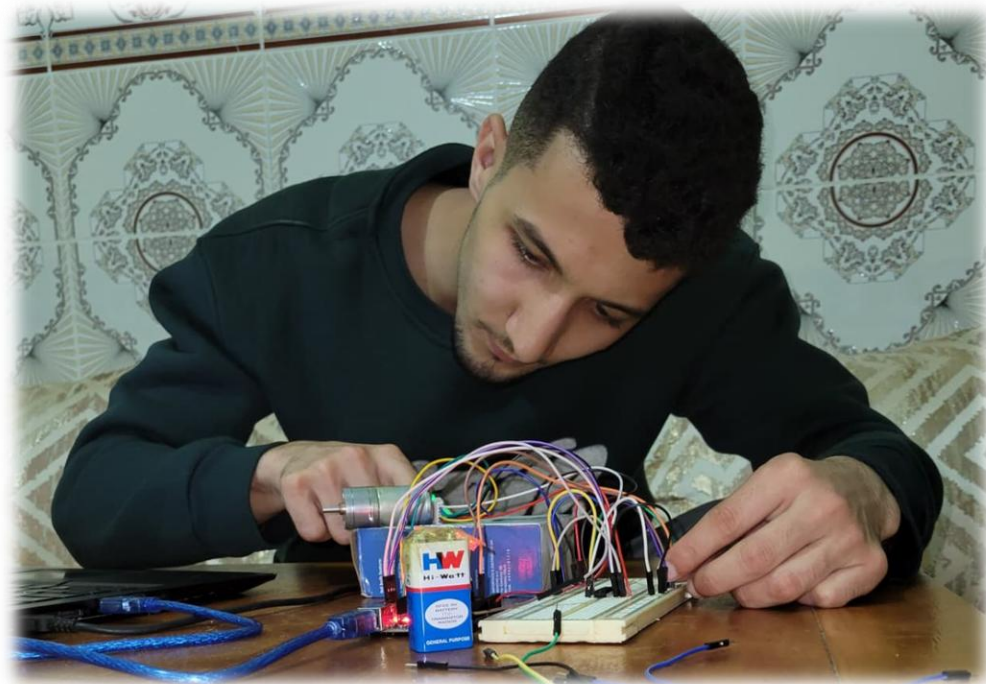
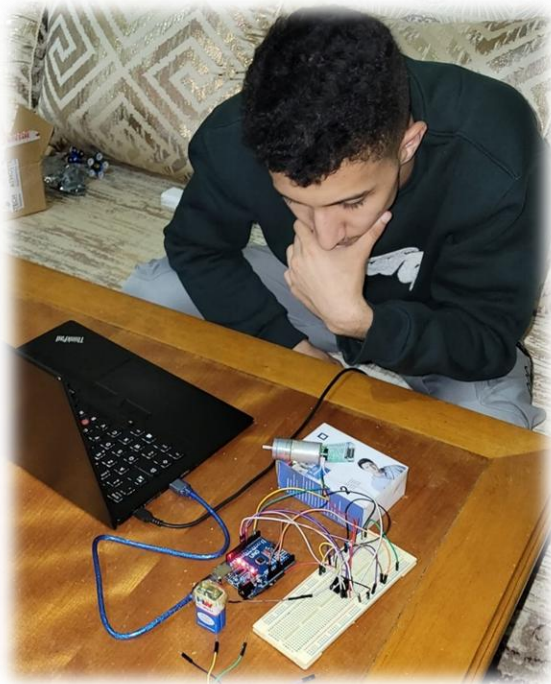


Illustration du principe du freinage régénératif (KERS) :

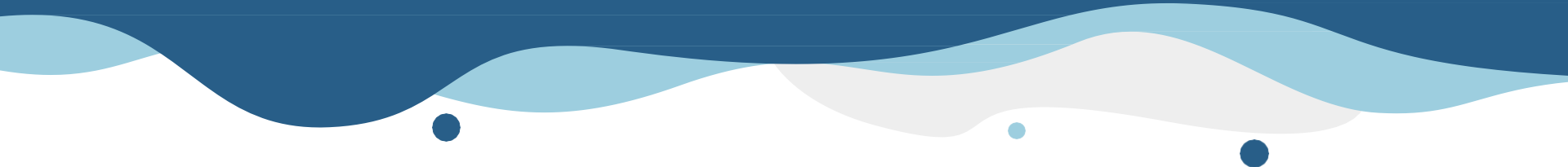




Conclusion



Merci pour votre attention!



ANNEXE

Code Arduino pour le traçage de la réponse de MCC sans correction :

```
1 // --- BROCHES DU PONT EN H (L293D) ---
2 const int ENA = 9; // PWM (Contrôle de vitesse)
3 const int IN1 = 7; // Direction 1
4 const int IN2 = 8; // Direction 2
5
6 // --- BROCHE DE L'ENCODEUR ---
7 const int ENCA = 2; // DOIT être sur la broche 2 (Interruption)
8
9 // --- LA CONSTANTE MODIFIÉE (TRICHE) ---
10 // En baissant cette valeur à 7.5, l'Arduino va multiplier par 2
11 // la vitesse calculée. Tes 101 rad/s deviendront environ 202 rad/s.
12 const float PPR = 7.5;
13
14 // --- VARIABLES GLOBALES ---
15 volatile long impulsions = 0;
16 unsigned long tempsPrecedent = 0;
17 unsigned long tempsDebutMesure = 0;
18
19 void setup() {
20     Serial.begin(9600);
21
22     pinMode(ENA, OUTPUT);
23     pinMode(IN1, OUTPUT);
24     pinMode(IN2, OUTPUT);
25     pinMode(ENCA, INPUT_PULLUP);
26
27     attachInterrupt(digitalPinToInterrupt(ENCA), compterImpulsions, RISING);
28 }
```

```
29 digitalWrite(IN1, LOW);
30 digitalWrite(IN2, LOW);
31 analogWrite(ENA, 0);
32
33 Serial.println("Temps(s);Vitesse(rad/s)");
34
35 delay(2000);
36
37 // TOP DÉPART
38 digitalWrite(IN1, HIGH);
39 digitalWrite(IN2, LOW);
40 analogWrite(ENA, 255);
41
42 tempsDebutMesure = millis();
43 tempsPrecedent = millis();
44 }
45
46 void loop() {
47   unsigned long tempsActuel = millis();
48   unsigned long dt = tempsActuel - tempsPrecedent;
49
50   if (dt >= 20) {
51     noInterrupts();
52     long impulsionsLues = impulsions;
53     impulsions = 0;
54     interrupts();
55
```

```
56     float tours = (float)impulsionsLues / PPR;
57     float rpm = tours * (60000.0 / dt);
58     float vit_rad_s = rpm * (2.0 * PI / 60.0);
59     float temps_s = (tempsActuel - tempsDebutMesure) / 1000.0;
60
61     Serial.print(temps_s);
62     Serial.print(";");
63     Serial.println(vit_rad_s);
64
65     tempsPrecedent = tempsActuel;
66 }
67 }
68
69 void compterImpulsions() {
70     impulsions++;
71 }
```