

ROBOT MARQUEUR DE PISTES D'ATHLETISME

Réalisé par : **ANASS EL HAYEL**

Code CNC : **OJ080T**

Encadré par : Mr.Essadki Mustapha

Filière : **TSI**

PLAN :

- I** CONTEXTUALISATION
ET PROBLEMATIQUE
- II** ANALYSE FONCTIONNELLE
DU ROBOT (SYSML)
- III** GESTION DE SUIVI
DE LA TRAJECTOIRE
- IV** DETECTION DES OBSTACLES

- V** CHOIX DU MOTEUR ET
ASSERVISSEMENT DE VITESSE
- VI** EQUILIBRAGE DE LA
ROUE ARRIERE
- VII** REALISATION
- VIII** CONCLUSION

I-CONTEXTUALISATION ET PROBLEMATIQUE

I-CONTEXTUALISATION ET PROBLEMATIQUE

II-ANALYSE FONCTIONNELLE DU ROBOT

III-SUIVI DE LA TRAJECTOIRE

IV-DETECTION DES OBSTACLES

V-CHOIX DU MOTEUR

VI-EQUILIBRAGE DE LA ROUE ARRIERE

VII-REALISATION

VIII-CONCLUSION

CONTEXTUALISATION



figure 1 :jeux olympiques 2008-finale course 100m



figure2 :jeux olympiques 2016 - demi-finale course 1500m

PROBLEMATIQUE RETENUE :

comment peut-on assurer un marquage robotisé et régulier des pistes d'athlétisme afin d'optimiser l'expérience athlétique ?



II-ANALYSE FONCTIONNELLE DU ROBOT

DIAGRAMME DE CAS D'UTILISATION:

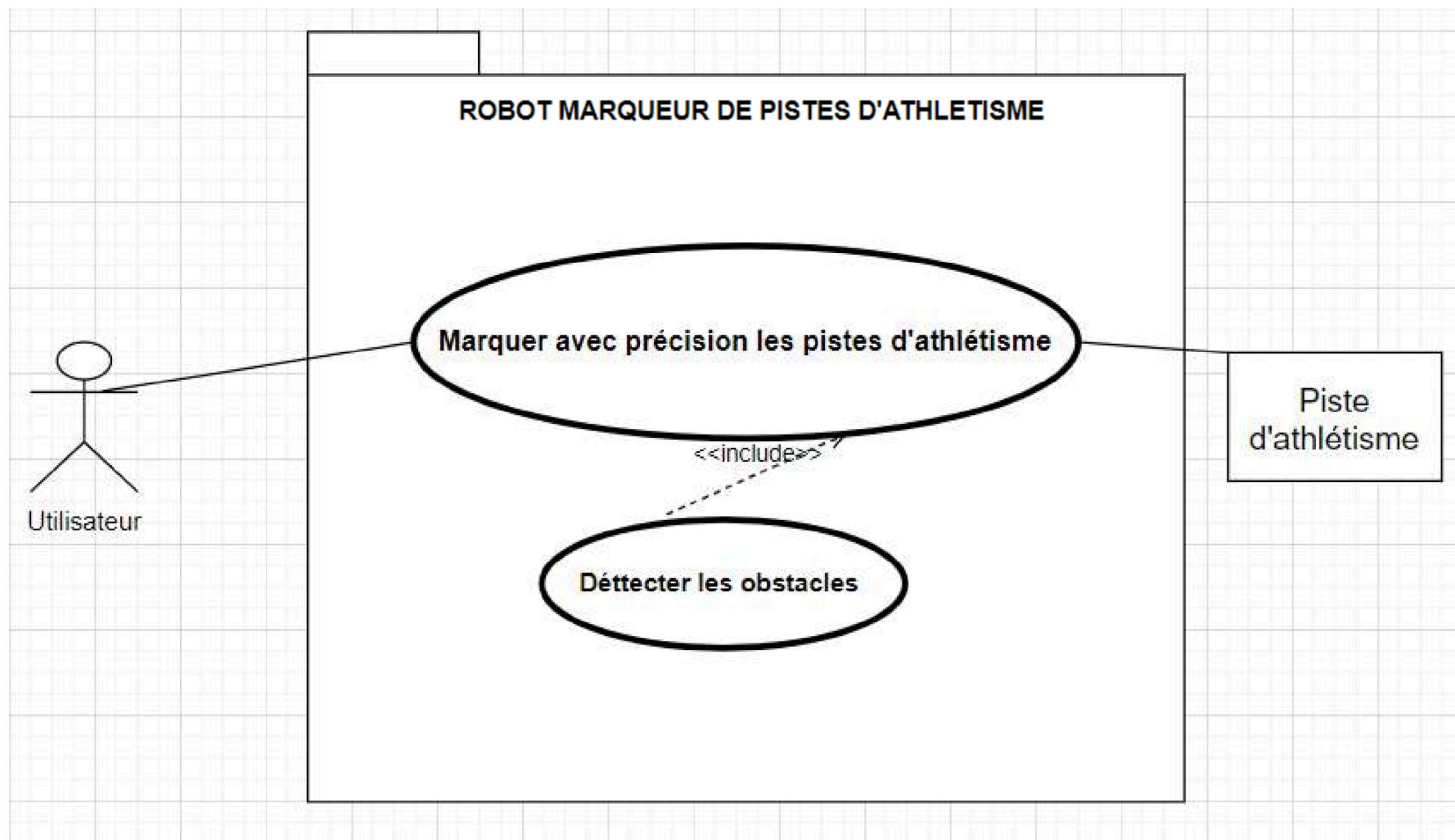
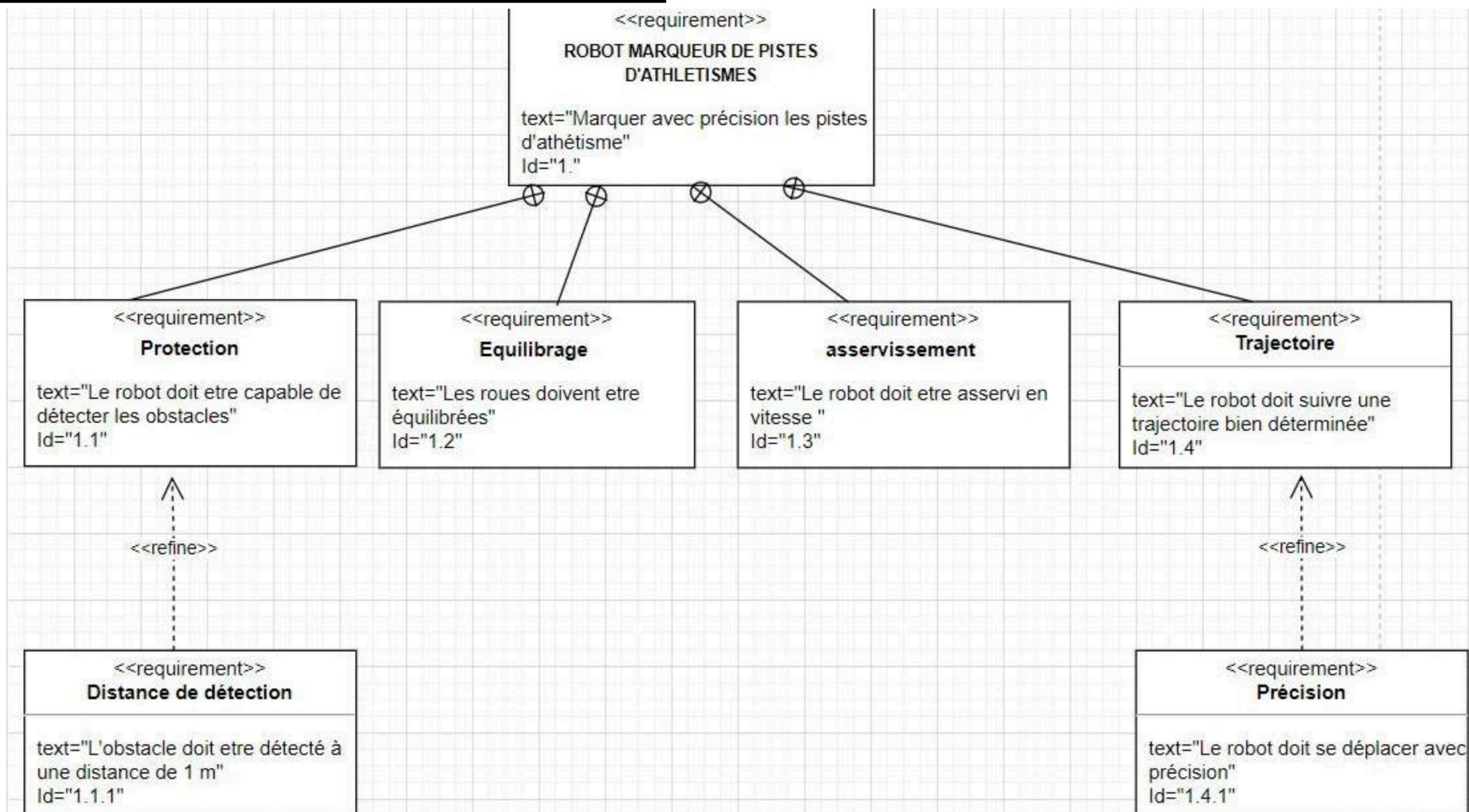


DIAGRAMME DES EXIGENCES:



III-GESTION DE SUIVI DE LA TRAJECTOIRE

■ Choix du capteur:

Caractéristique	Capteur Infrarouge	Capteur Ultrason
Principe de fonctionnement	Détecte la lumière infrarouge réfléchiée par la ligne	Utilise les ondes sonores pour mesurer la distance
Précision	Très précis sur des surfaces réfléchissantes claires et foncées	Moins précis pour suivre une ligne directement
Temps de réponse	Rapide (microsecondes à millisecondes)	Relativement lent (millisecondes à centièmes de secondes)
Influence des surfaces	Sensible aux couleurs et à la réflectivité de la surface	Moins affecté par les couleurs mais sensible aux textures et aux angles

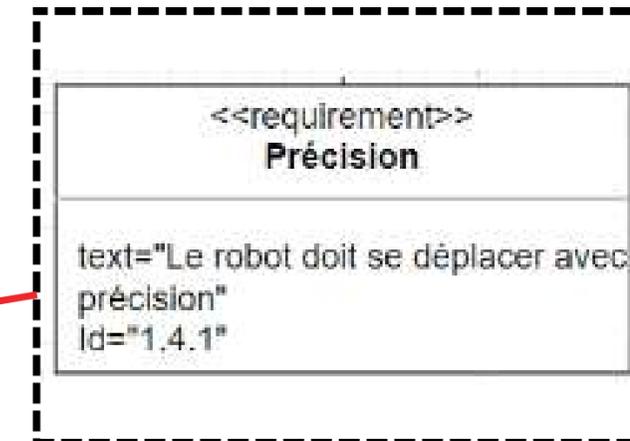


figure 3 : Caractéristiques des capteurs candidats

■ Modèle choisit :

le modèle choisit est un capteur infrarouge TCRT-5000

● CARACTERISTIQUES:

- Distance de mesure: 12mm
- Hauteur du module (partie plastique): 7 mm
- Voltage: 5V
- Collecteur max Voltage: 70V
- Temperature de fonctionnement: -25C ~ +85



figure 4 :TCRT-5000

● PRINCIPE DE FONCTIONNEMENT :

- la diode émettrice émet un faisceau de lumière infrarouge puis le phototransistor mesure la quantité de lumière réfléchi

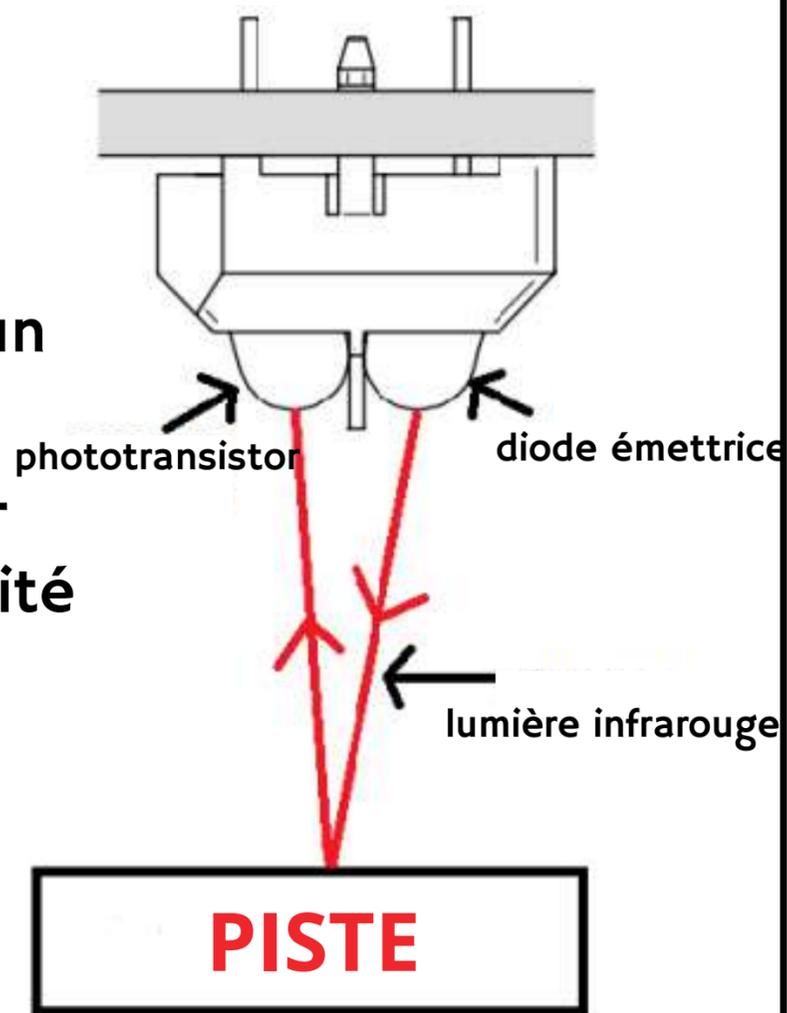


figure 5 : TCRT5000

● SCHEMA ELECTRIQUE:

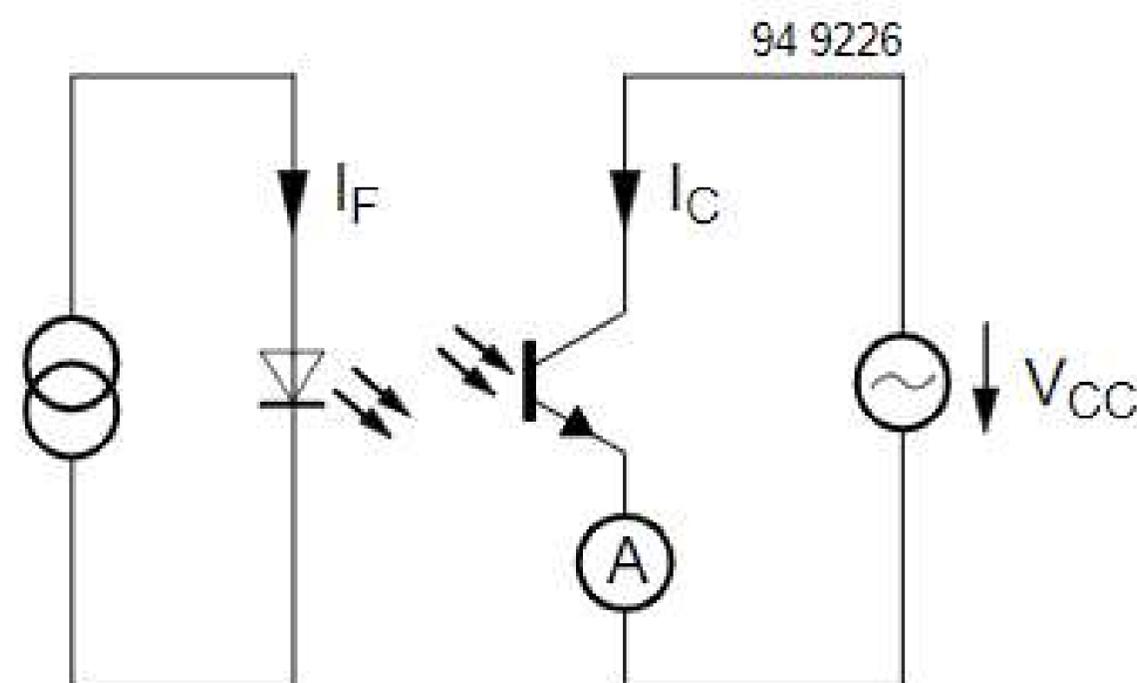
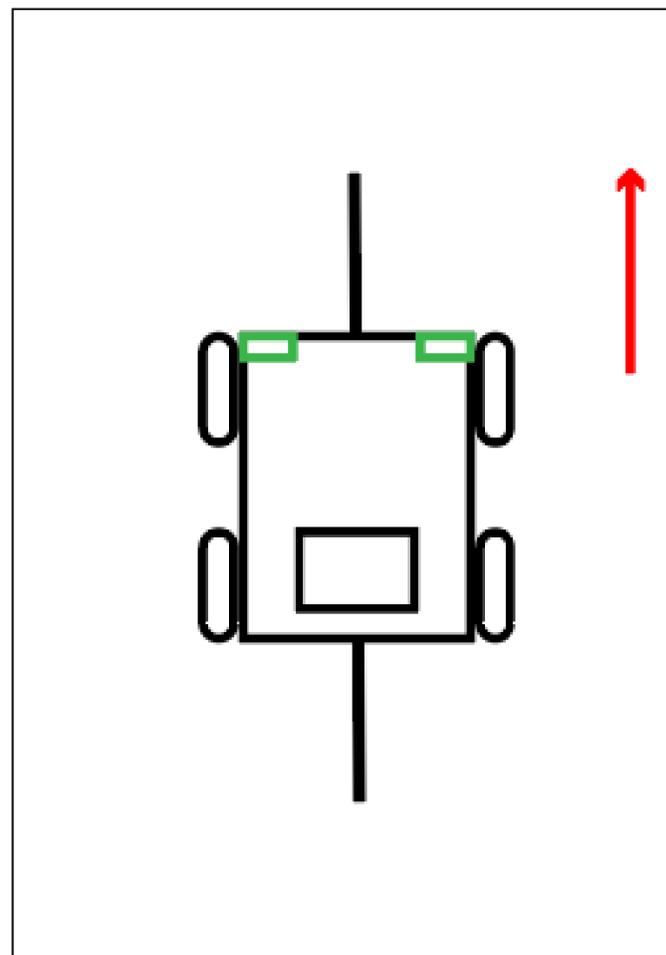
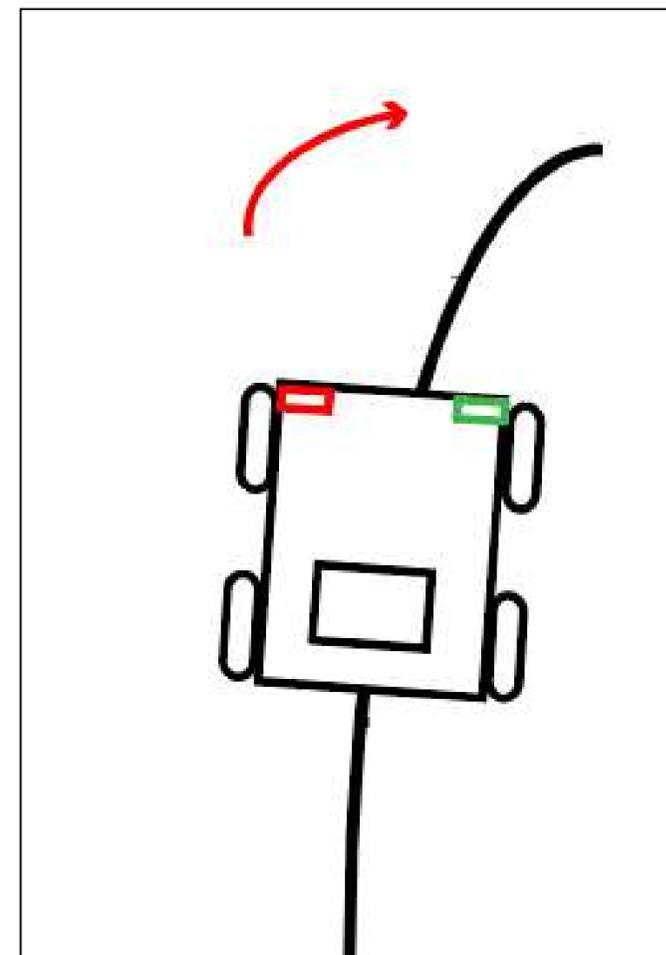


figure 6 :schéma électrique du TCRT 5000

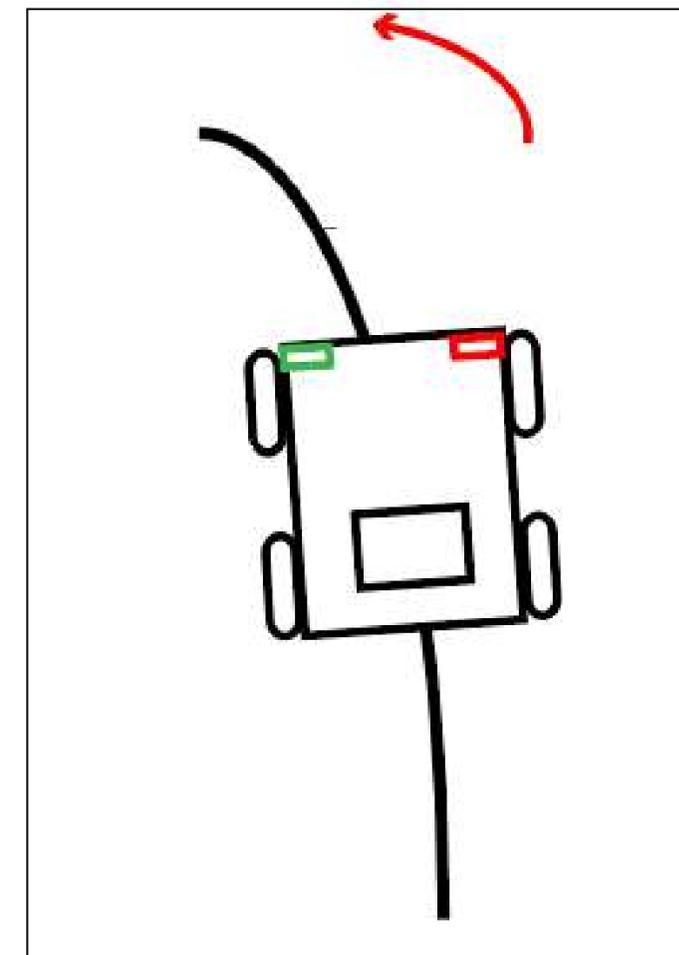
● PRINCIPE DE FONCTIONNEMENT GLOBALE :



G	D
1	1



G	D
1	0



G	D
0	1

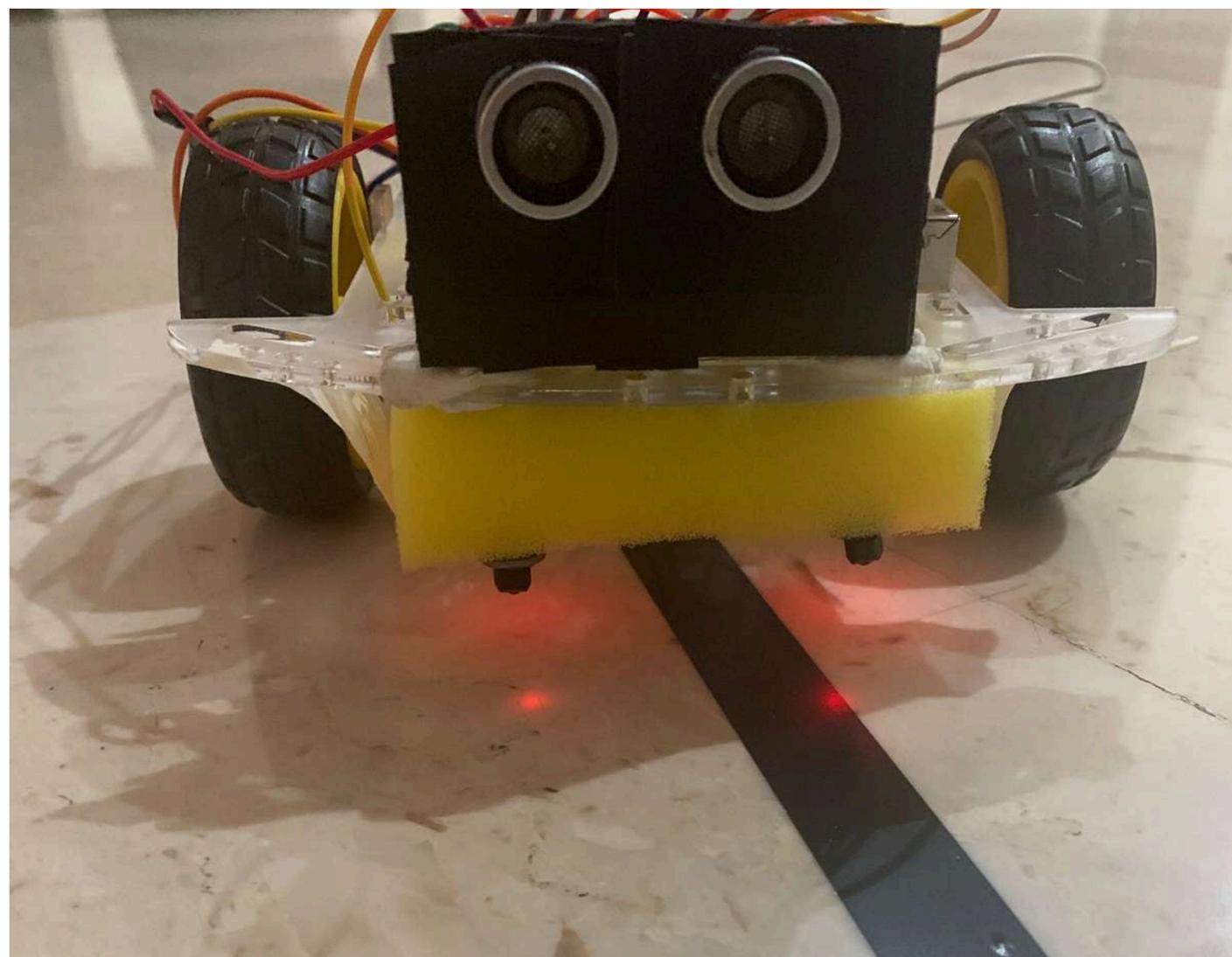


figure 7 : Expérience

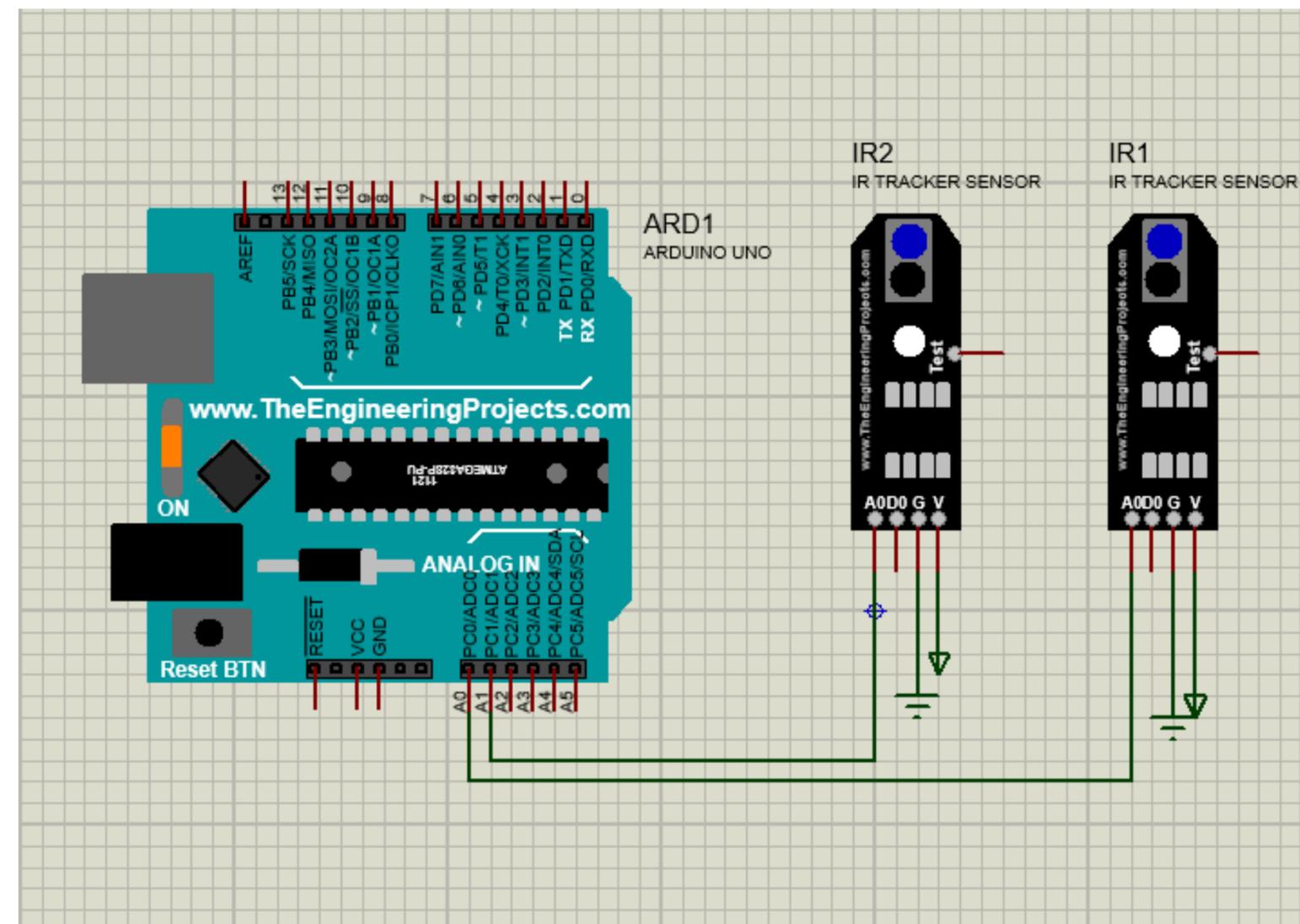


figure 8 :schéma de câblage du tcr5000 sur isis

Problème :

Lors de son déplacement, le robot zigzague d'une manière perturbante

■ Solution proposée



Ajouter un capteur au milieu



figure 9 :Réalisation de la solution proposé

IV-DETECTION DES OBSTACLES

■ Choix du capteur:

Capteur infrarouge SHARP GP2Y0A21



Figure 10 :GP2Y0A21

● Caractéristiques :

- Tension de fonctionnement: 4.5 V to 5.5 V
- Consommation de courant moyenne : 30 mA
- Distance de mesure : 10 cm - 80 cm
- Temps de réponse: 38 ± 10 ms

■ Choix du capteur:

Capteur ultrason HC-SR04:



Figure II:HC-SR04

● **Caractéristiques :**

-Alimentation: 5 Vcc

-Consommation: 15 mA

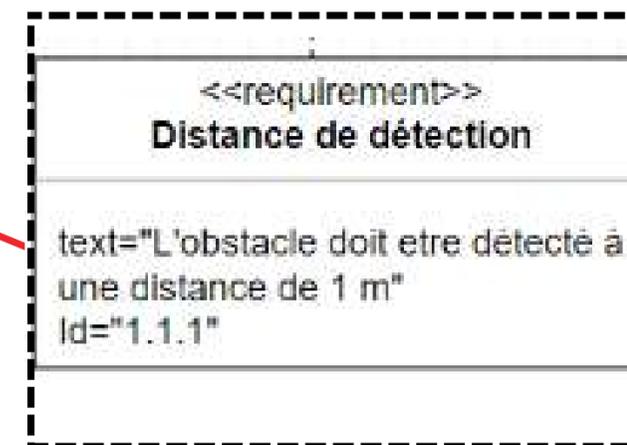
-Fréquence: 40 kHz

-Portée: de 2...10 cm à 4 m

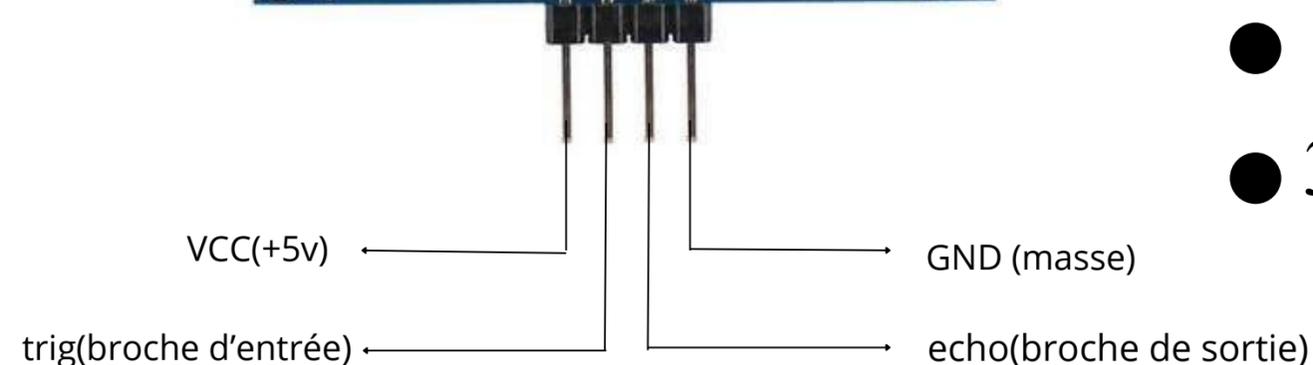
Dimensions: 45 x 21 x 18 mm

■ Modèle choisit :

portée de mesure(HC-SR04) : 6cm à 4 m > portée de mesure (SHARP GP2Y0A21) : 10 cm - 80 cm



■ Par conséquent le modèle choisit est le HC-SR04



- 1 sortie : echo
- 3 entrées : GND, VCC, TRIG

■ Principe de fonctionnement :

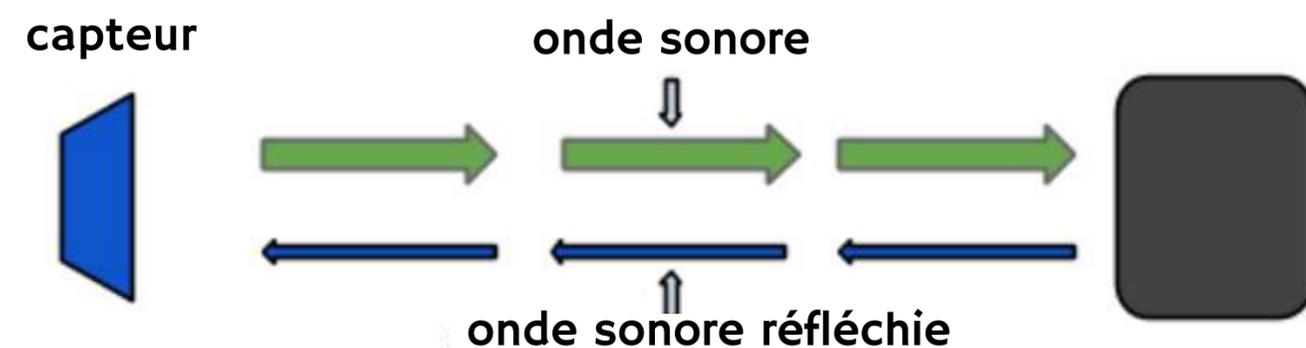


figure 12 : Scéma descriptive

■ portée de mesure:

$$\text{distance de mesure} = \text{vitesse du son} \times \frac{\Delta t}{2}$$



Δt : la durée pour que le signal ultrasonique émis atteigne un objet ,rebondisse sur cet objet, puis revienne au capteur.

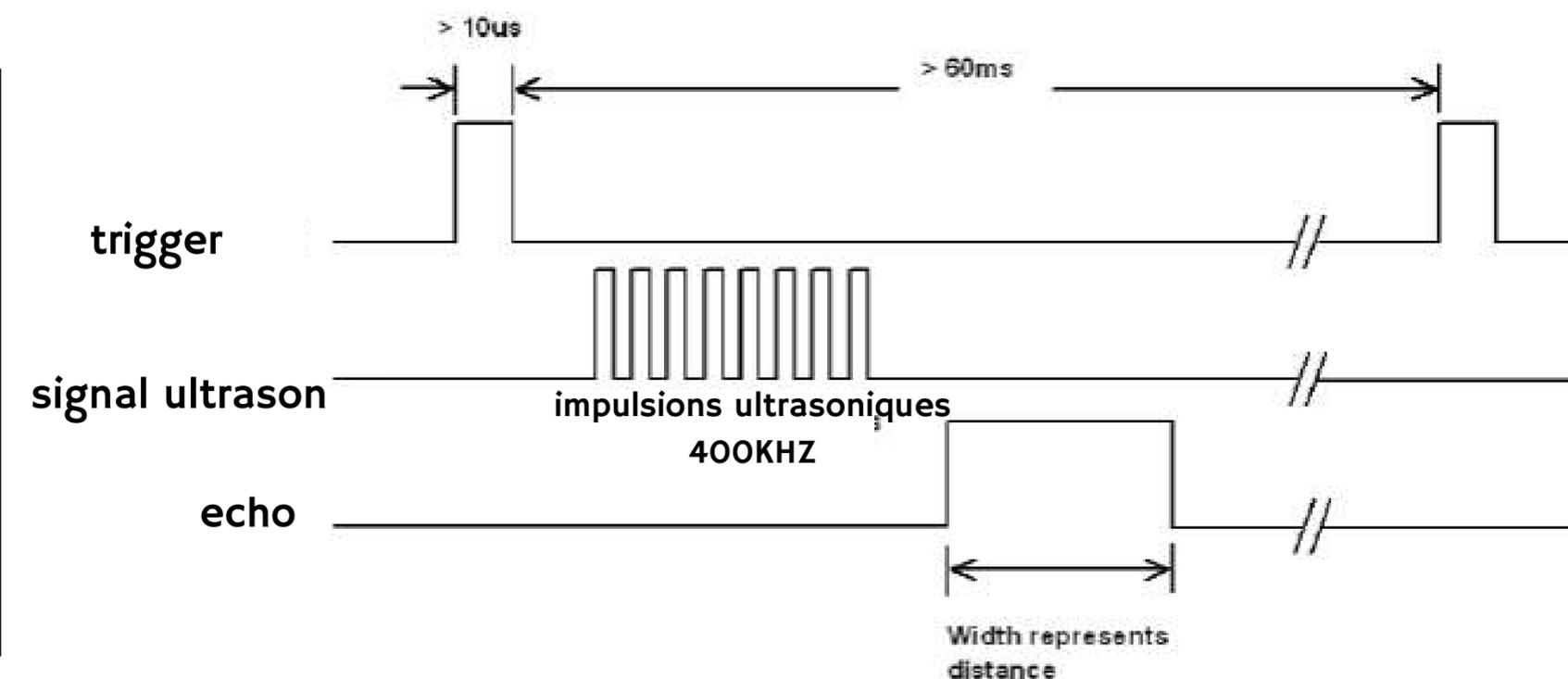


figure 13 : Principe de fonctionnement

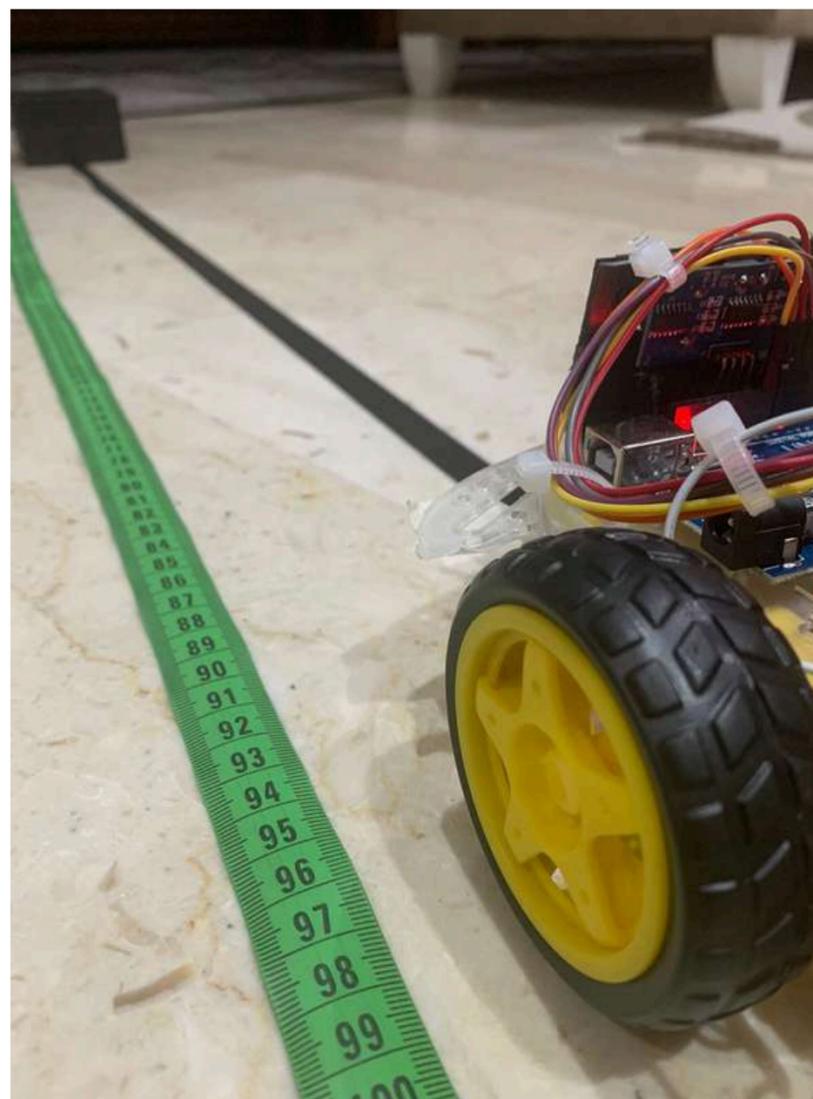


figure 14:Expérimentation du capteur ultrason

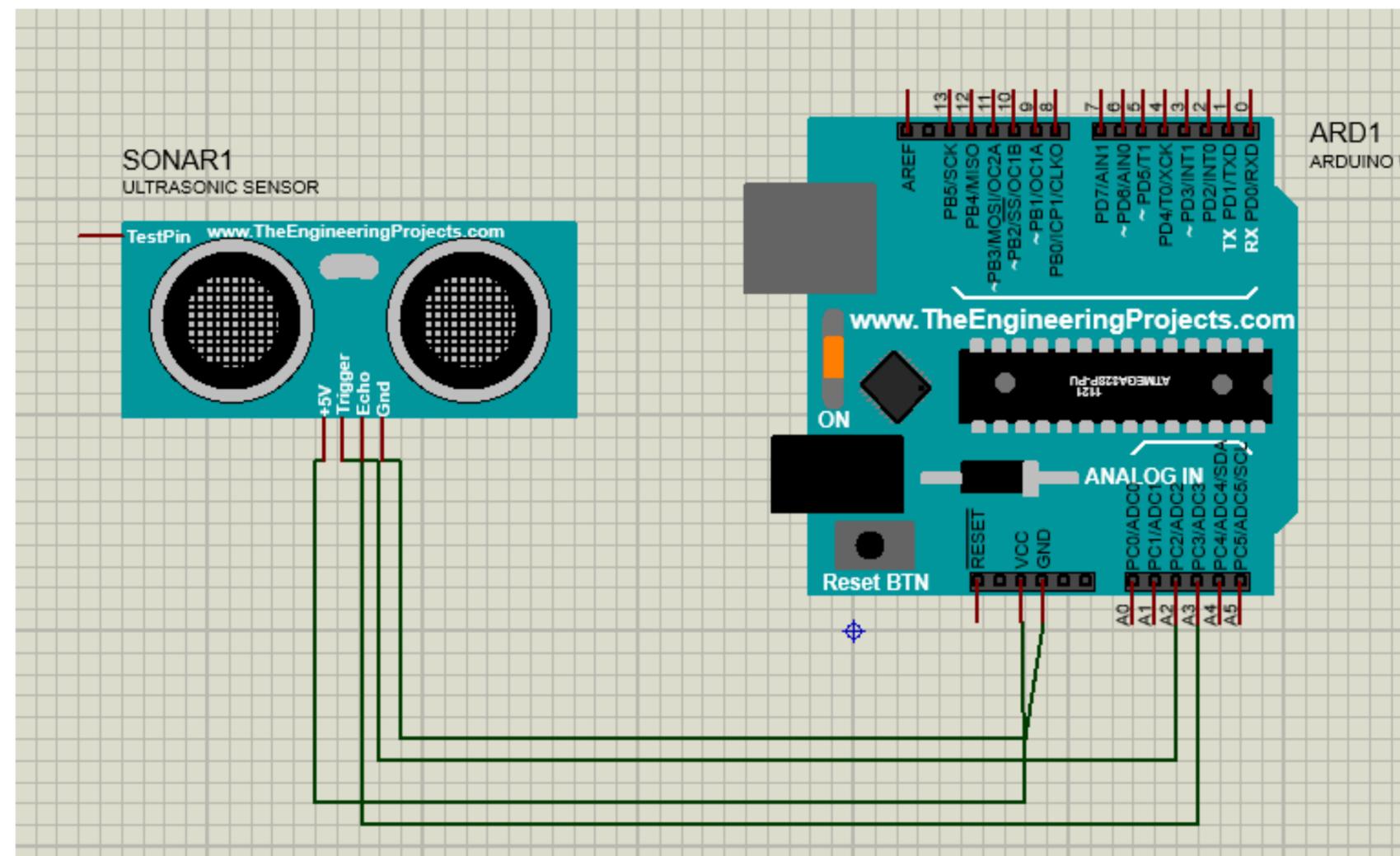


figure 15 :schéma de câblage du HC-SR04 sur isis

V-CHOIX DU MOTEUR
ET ASSERVISSEMENT
DE VITESSE

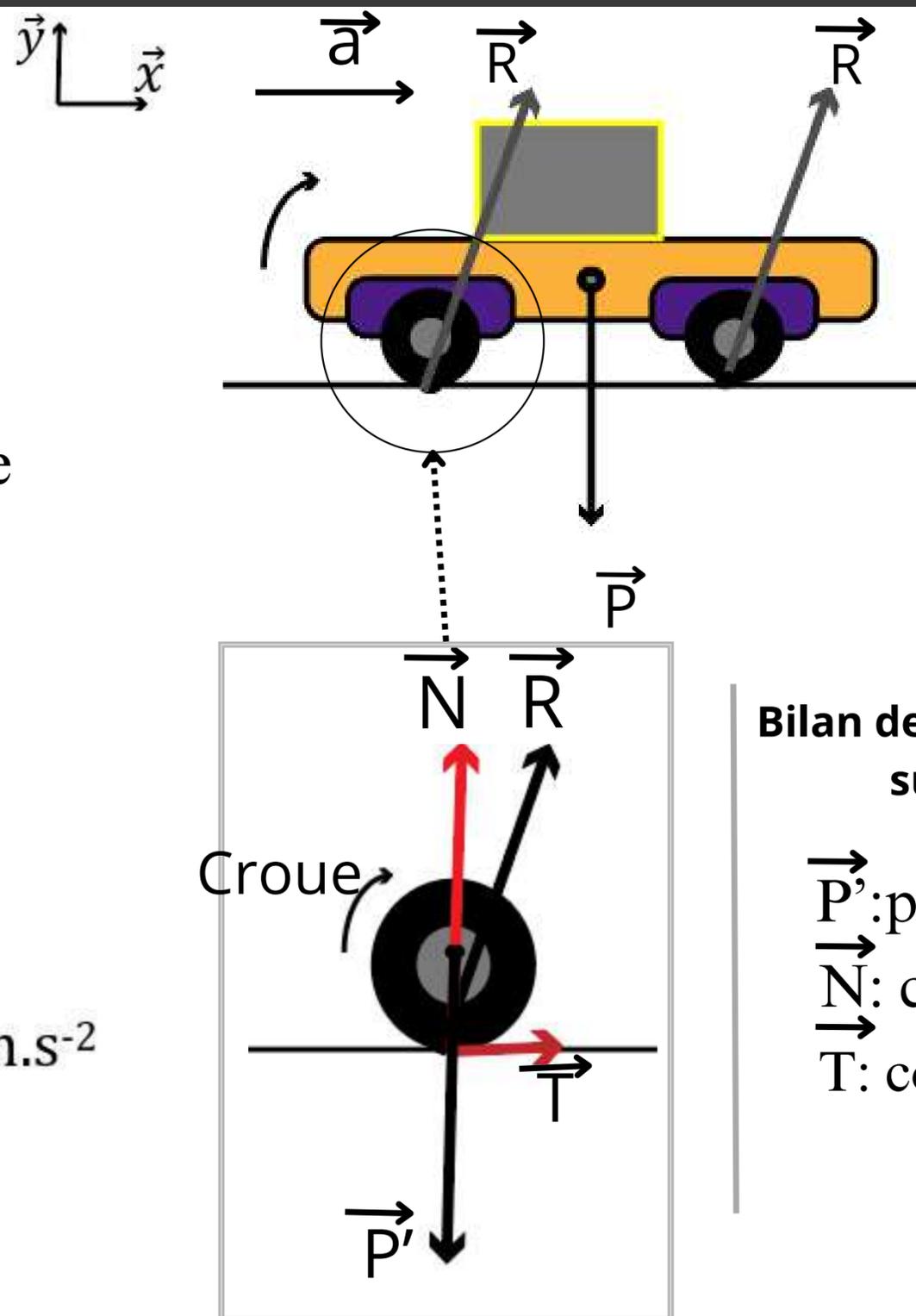
CHOIX DU MOTEUR :

Hypothèses:

- La masse est uniformément répartie

Données nécessaires :

- Masse(Kg) : 50 kg
- Vitesse maximale : 4m/s
- Diamètre d'une roue: 20 cm
- Accélération au démarrage: 2m.s^{-2}



Bilan des forces appliqués sur le robot :

\vec{P} : poids du robot
 \vec{R} : réaction au sol(pour une roue)

Bilan des forces appliqués sur une roue :

\vec{P}' : poids de la roue arriere
 \vec{N} : composante normale
 \vec{T} : composante tangentielle

■ TRC :

$$\sum \vec{F}_{ext} = M\vec{a}$$

$$\vec{P} + \vec{T} + \vec{N} = M\vec{a}$$

--Par projection on trouve:

/ (Ox) :

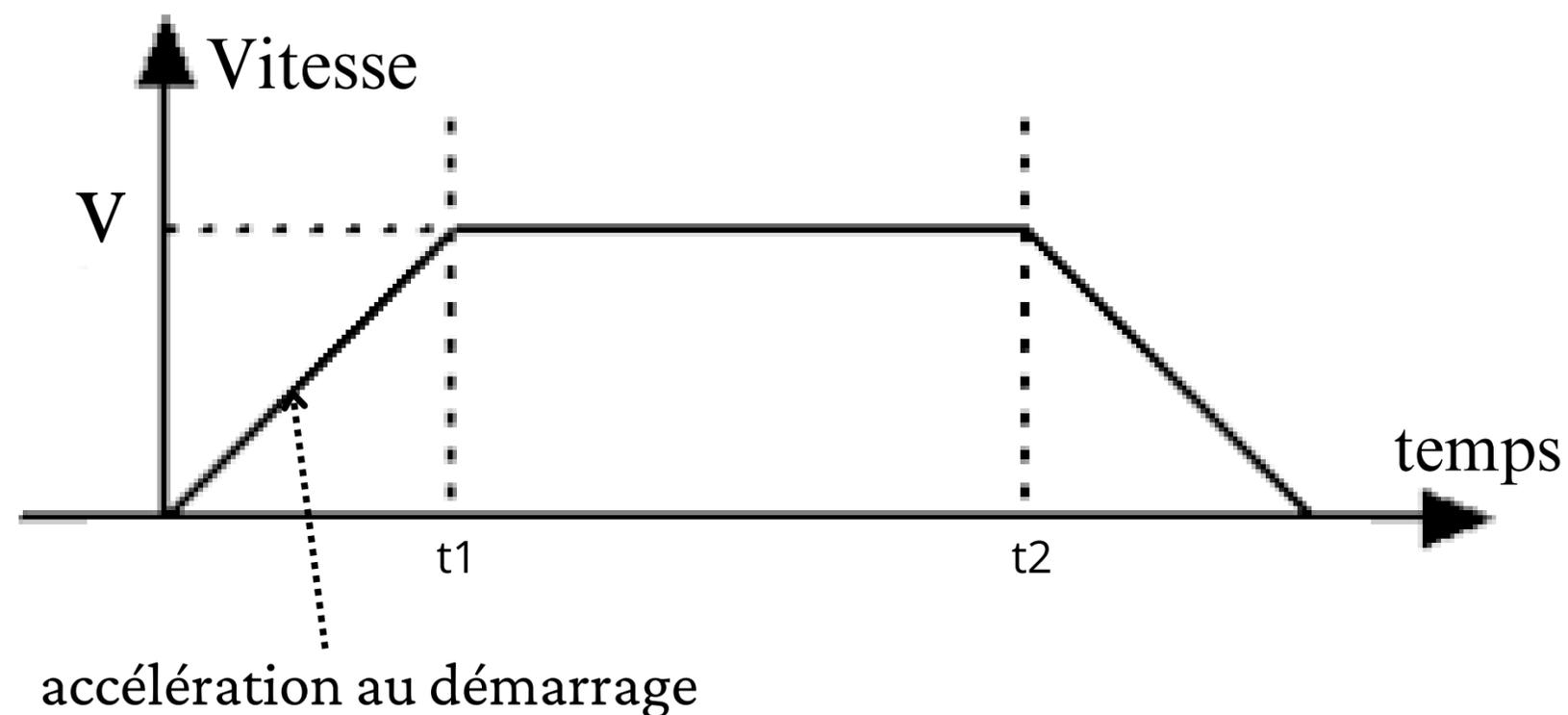
$$T = \frac{Ma}{4}$$

AN : $T = 25N$

● Moment d'inertie de la roue

$$J = M \frac{d^2}{32} = 0,0625 \text{ kg.m}^2$$

● Loi de vitesse pour une période de marquage:



■ TMC :

$$\sum \vec{M}(\vec{F}_{ext}) = J \frac{d\vec{\Omega}}{dt} = 2J \frac{\vec{a}}{d}$$

--avec :

$$\vec{M}_G(\vec{T}) = \vec{R} \wedge \vec{T} = RT \vec{e}_z$$

$$\vec{M}_G(\vec{P}) = \vec{0}$$

$$\vec{M}_G(\vec{N}) = \vec{0}$$

--Par projection sur (GZ):

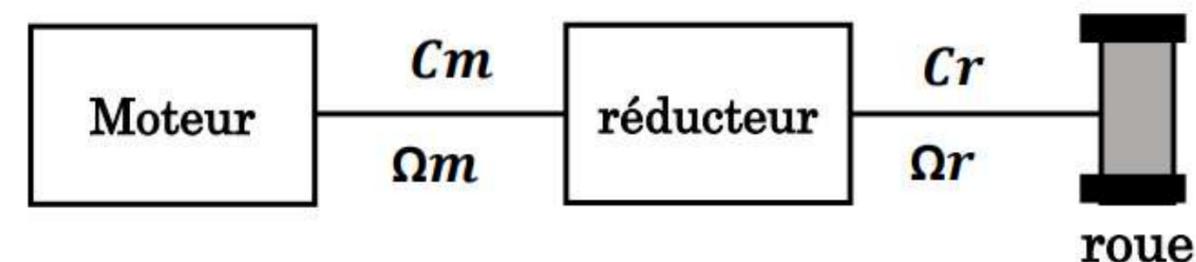
$$M_{GZ} = \frac{d}{2} T - C_r = 2J \frac{a}{d}$$

--d'où:

$$C_r = \frac{d}{2} T - 2J \frac{a}{d}$$

$$\text{AN : } C_r = 1.25 \text{ Nm}$$

■ chaîne cinématique:



● Vitesse à l'arbre de la roue

$$\Omega_r = \frac{2v}{d} = 40 \text{ rad/s} \quad \text{donc: } N_r = 380 \text{ tr/min}$$

● Vitesse à l'arbre moteur :

La vitesse la plus rencontrée au marché: $N_m = 2000 \text{ tr/min}$

● Rapport de transmission:

$$r = \frac{N_m}{N_r} \cong 5.5$$

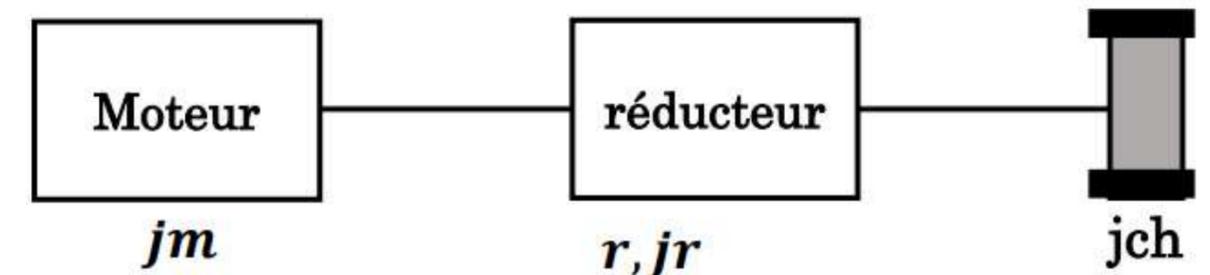
- Couple moteur :

$$C_m = \frac{Cr}{r} = 0,2 \text{ Nm}$$

- Moteur choisi : M540E0741

Specification	Units	Servo motors	
		M540E 0741	M543E 1270
Maximum Voltage	Vdc	40	60
Typical Voltage	Vdc	24 ←	36
Maximum Continuous Output Power	Watts	52	94
Maximum No-load speed	rpm	6000	4700
Typical speed @ rated torque	rpm	2500 ←	2250
Rated Torque	Nm	0.2 ←	0.4
Maximum Peak Torque	Nm	1.1	1.44
Typical . No load current	Amps	0.5	0.30
Rotor Inertia	Kgcm ²	0.270	0.530
Mechanical time constant	milli secs	8.4	8.0
Torque Constant	Nm / A	0.071	0.121
Voltage Constant	V / 1000 rpm	7.41	12.7
Terminal Resistance	Ohms	1.55	2.2
Rotor inductance	mH	3.39	6.4

- Calcul du moment d'inertie total :



- moment d'inertie total ramené à l'arbre moteur

$$j_t = j_m + \frac{j_{ch}}{r^2} + j_r \quad / \quad (\text{L'effet inertiel du réducteur est négligé})$$

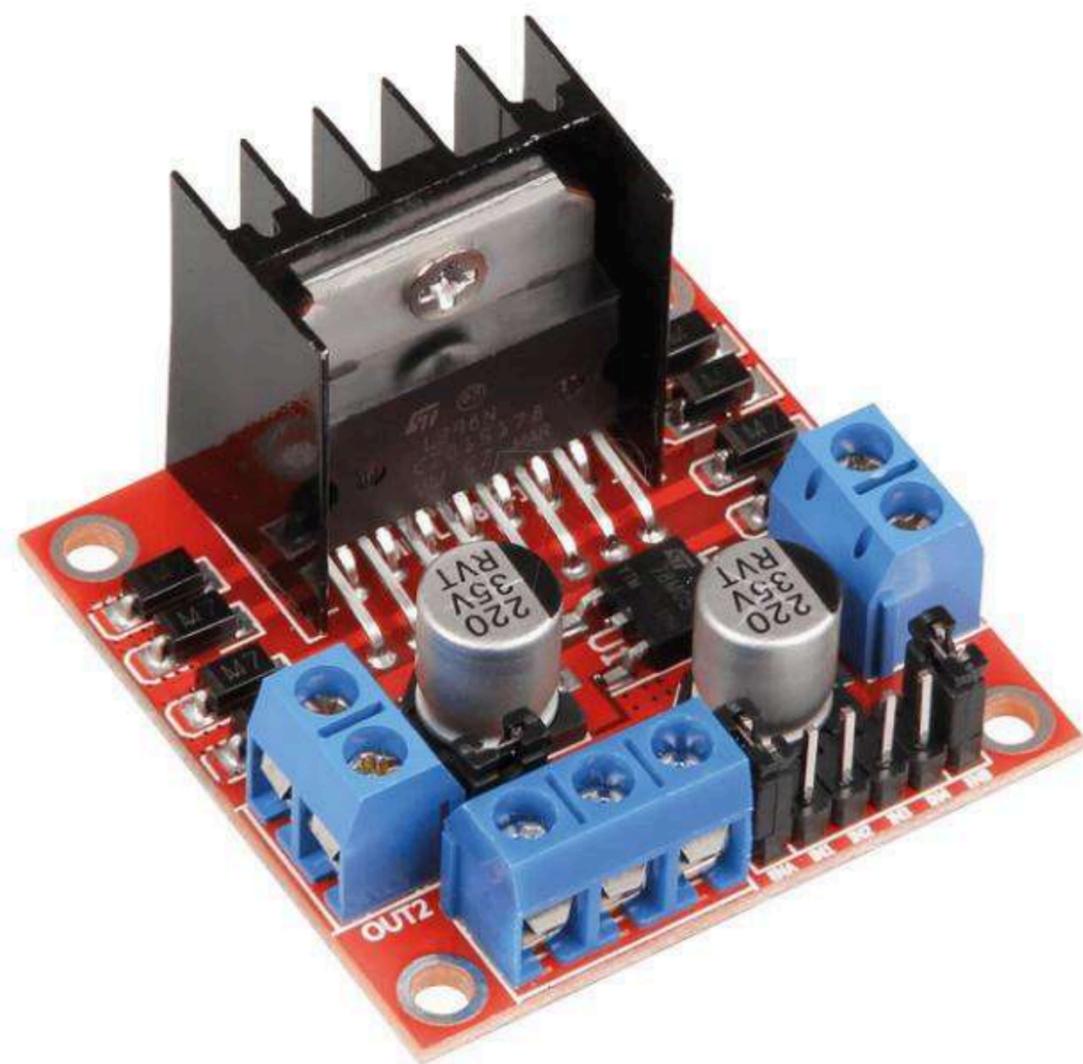
--Avec

$$j_{ch} = 0,0625 \text{ kg.m}^2$$

--Alors

$$J_t = 2 \times 10^{-3} \text{ kg.m}^2$$

● **Pont en H L289N :**



-Cet hacheur très populaire dans le marché a la capacité de piloter de divers moteurs ainsi que 2 moteurs à la fois.

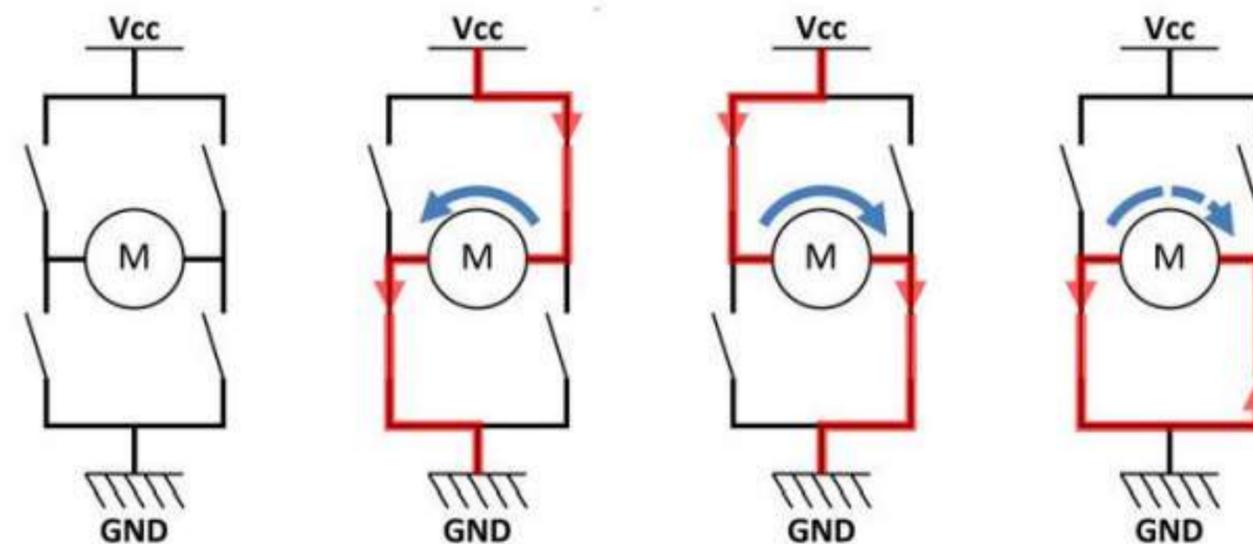
● **Caractéristiques techniques :**

Alimentation : 12 V 24 V 36V 15 A

Puissance de sortie maximale : 100 W

--La MCC ayant une puissance de 52 W , elle est bien compatible avec le hacheur L289N

● **Principe :**



Asservissement de vitesse:

Pour assurer un fonctionnement optimal en accord avec les consignes, le robot doit être contrôlé en vitesse afin d'atteindre 4 m/s (sa vitesse maximale) pendant la période de marquage.

● Modèle de la machine à courant continu:

$$1. \quad u(t) = e(t) + R i(t) + L \frac{di(t)}{dt}$$

$$2. \quad J_t \frac{d\Omega(t)}{dt} = C_m(t) - C_r(t)$$

$$3. \quad e(t) = K \Omega(t)$$

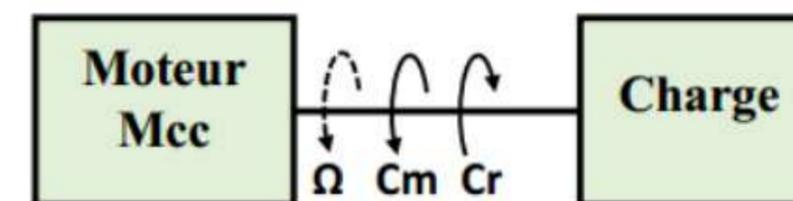
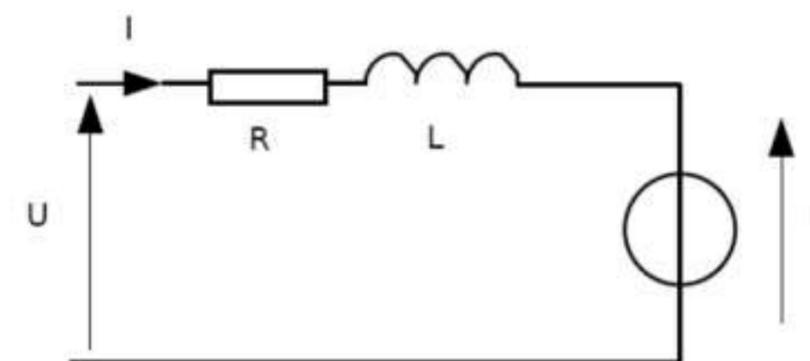
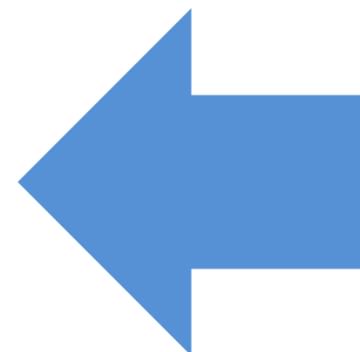
$$4. \quad C_m(t) = K i(t)$$

$$R = 1.55 \, \Omega$$

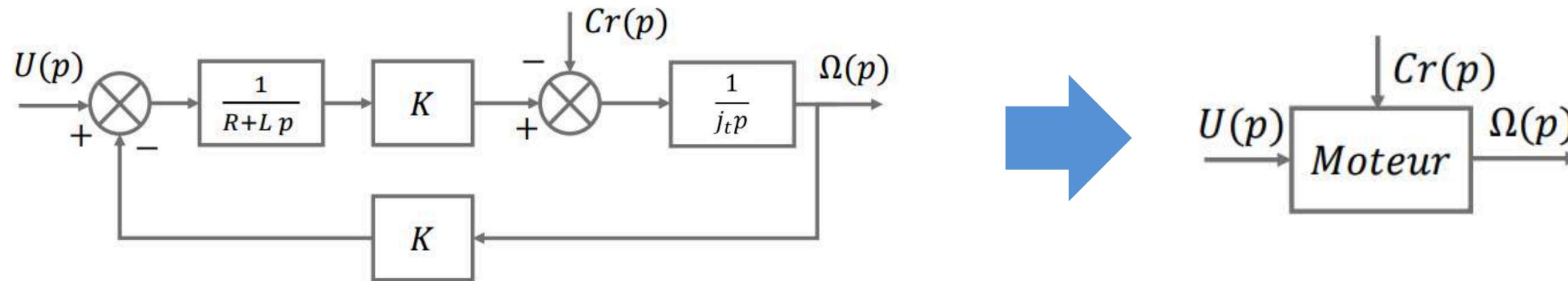
$$L = 3.39 \, \text{mH}$$

$$K = 0.071$$

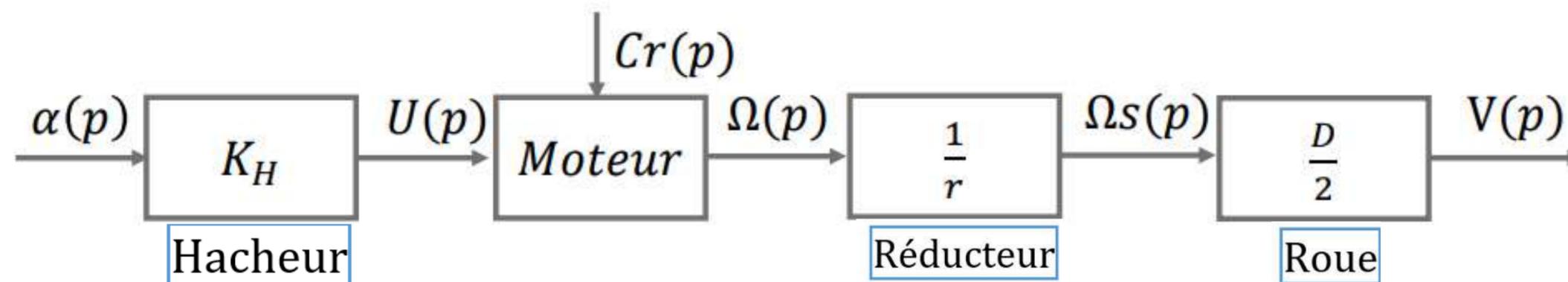
$$J_t = 2 \times 10^{-3} \, \text{kg.m}^2$$



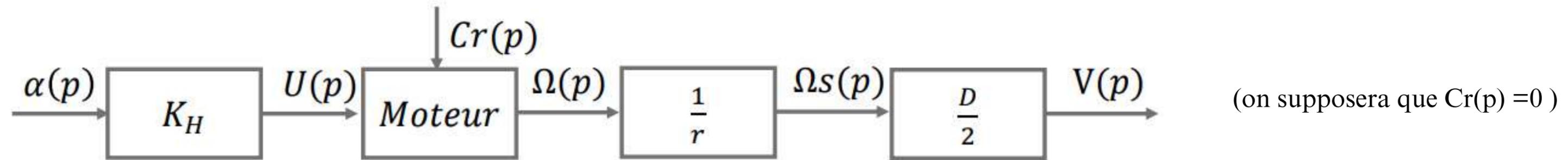
- **Modèle de la machine à courant continu:**



- **Modèle de la chaine cinématique:**



● **Détermination des fonctions de transferts**



■ **Fonction de la machine $M(p)$:**

$$M(p) = \frac{k}{L \cdot j_t p^2 + R \cdot j_t p + k^2} \quad \longrightarrow \quad M(p) = \frac{K_m}{(1 + \tau_e p)(1 + \tau_m p)} \quad \longrightarrow$$

$\tau_e = \frac{L}{R}$	$\tau_m = \frac{R \cdot j_t}{k^2}$	$K_m = \frac{1}{k}$
2,19ms	0.6s	14,08

■ **Fonction approchée $M(p)$:**

On a : $\tau_m \gg \tau_e \quad \longrightarrow \quad M(p) \cong \frac{K_m}{1 + \tau_m p}$ (Principe du pole dominant)

■ **Fonction de transfert du système en BO :**

$$FTBO(p) \cong \frac{K_m \cdot K_H \cdot \frac{1}{r} \cdot \frac{D}{2}}{1 + \tau_m p} = \frac{K_o}{1 + \tau_m p} \quad \text{avec} \quad K_o = 6 \quad \text{et} \quad \tau_m = 0.6 \text{ s}$$

● Systeme sans correction :

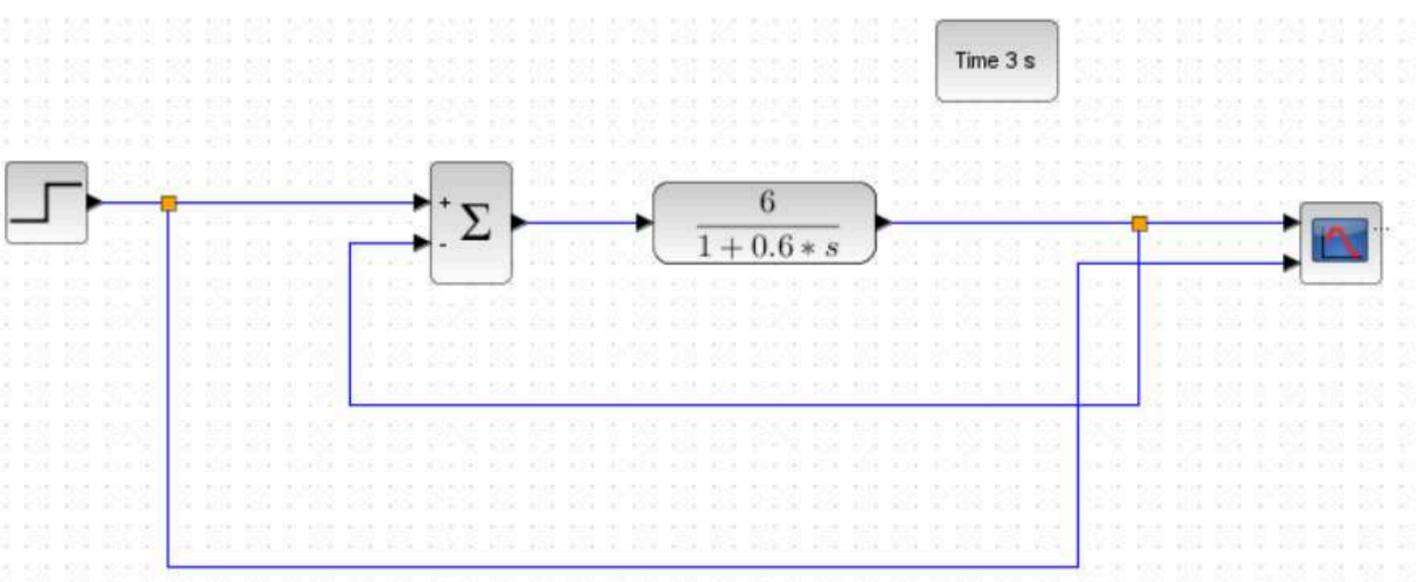
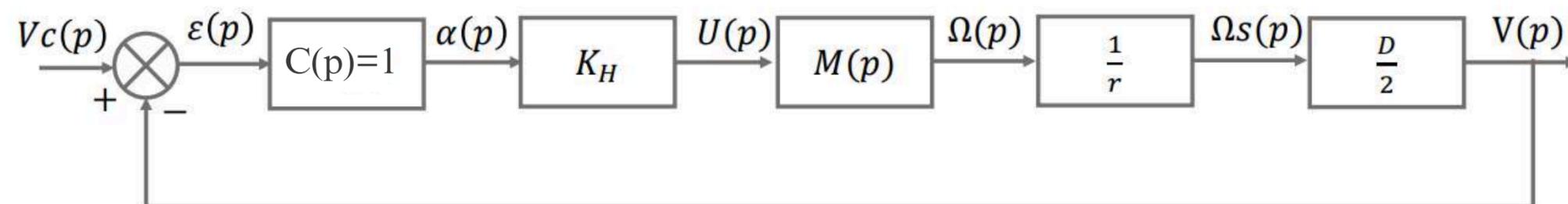


figure 16 :modélisation du système réduit non corrigé sur scilab

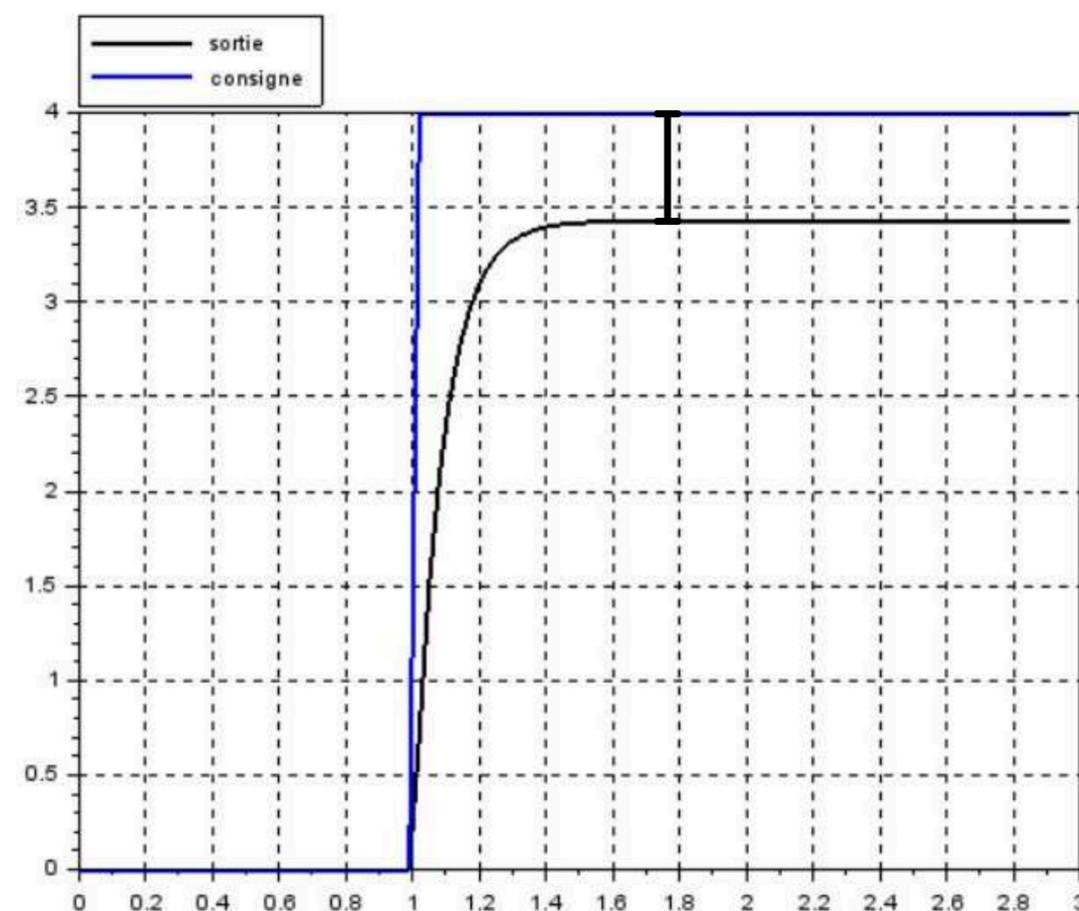
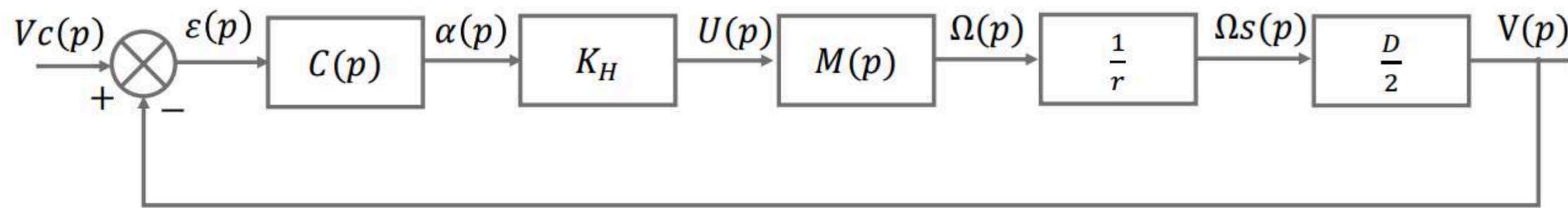


figure 17 :réponse du système à la consigne donnée

Conclusion :
 l'existence d'une erreur statique non nulle ce qui endommage la précision du système d'où la nécessité d'une **correction**

● Choix des paramètres du correcteur

■ Schéma bloc du système corrigé



■ Choix du correcteur

Type de correcteur	Effet sur l'erreur statique	Effet sur la réponse transitoire
Proportionnel-Intégral (PI)	Réduit l'erreur statique à zéro sans influencer significativement sur les autres caractéristiques	Combinaison d'amélioration de la réponse transitoire et réduction de l'erreur statique
Proportionnel-Dérivé (PD)	Ne réduit pas l'erreur statique à zéro	Améliore la réponse transitoire et la stabilité sans un dépassement important

Correcteur (PI)

$$C(p) = K_p \left(1 + \frac{1}{T_i \cdot p} \right) = \frac{K_p (1 + T_i \cdot p)}{T_i \cdot p}$$

■ Méthode de compensation du pôle :

$$T_i = \tau_m \longrightarrow FTBO(p) = H(p).C(p) = \frac{K_p K_o}{T_i \cdot p} \quad K_p = 1$$

$$\longrightarrow FTBF(p) = \frac{FTBO(p)}{1+FTBO(p)} = \frac{K_o}{1+\frac{T_i}{K_p} \cdot p} = \frac{K_o}{1+\tau \cdot p} \quad \text{on prend: } \tau = T_i$$

$$T_i = 0.6s$$

■ Calcul de l'erreur statique :

$$\varepsilon(p) = V_c(p) - H(p).C(p).\varepsilon(p) \longrightarrow \varepsilon(p) = \frac{V_c(p)}{1+\frac{1}{T_i \cdot p}}$$

$$\varepsilon_s = \lim_{t \rightarrow \infty} \varepsilon(t) = \lim_{p \rightarrow 0} p \cdot \varepsilon(p) = \lim_{p \rightarrow 0} p \cdot \frac{\frac{V_c}{p}}{1+\frac{1}{T_i \cdot p}} = \lim_{p \rightarrow 0} \frac{p \cdot V_c}{p + \frac{1}{T_i}} = 0$$

Donc on a éliminé l'ecart statique

● Systeme avec correction :

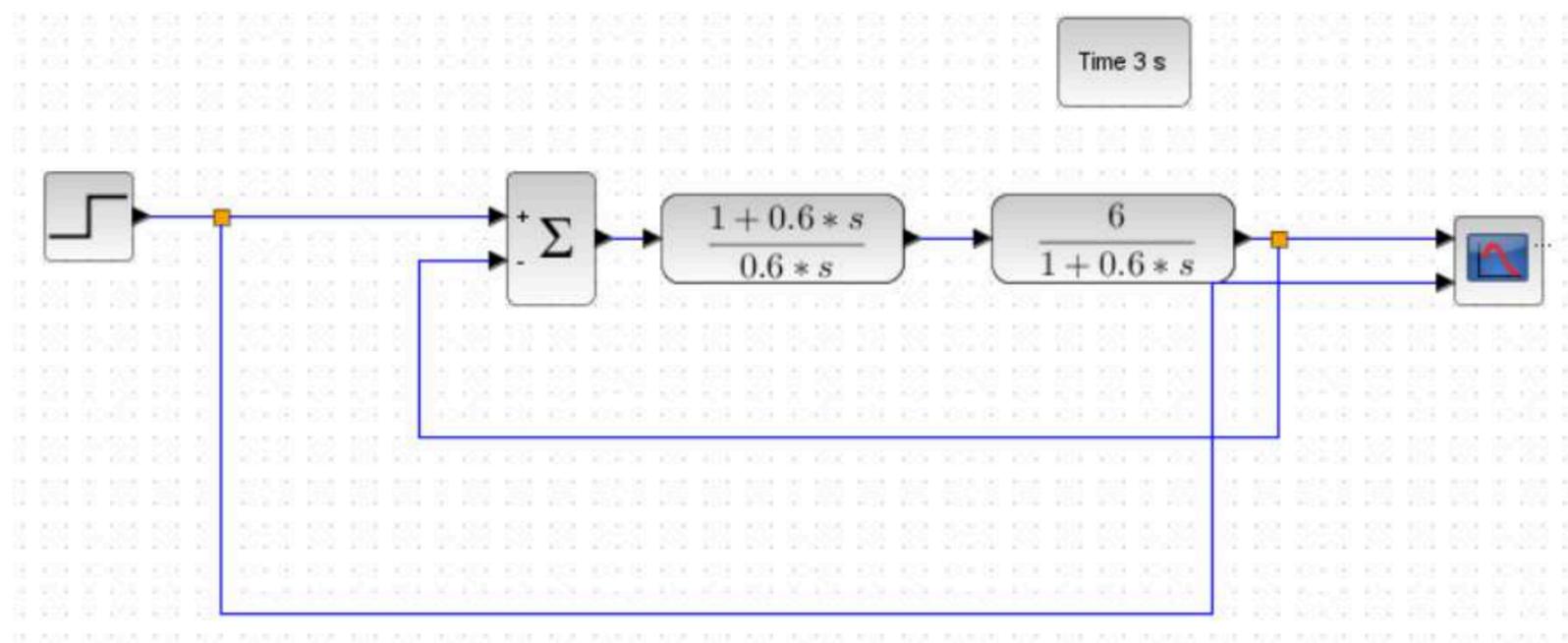


figure 18 :modélisation du système réduit corrigé sur Scilab

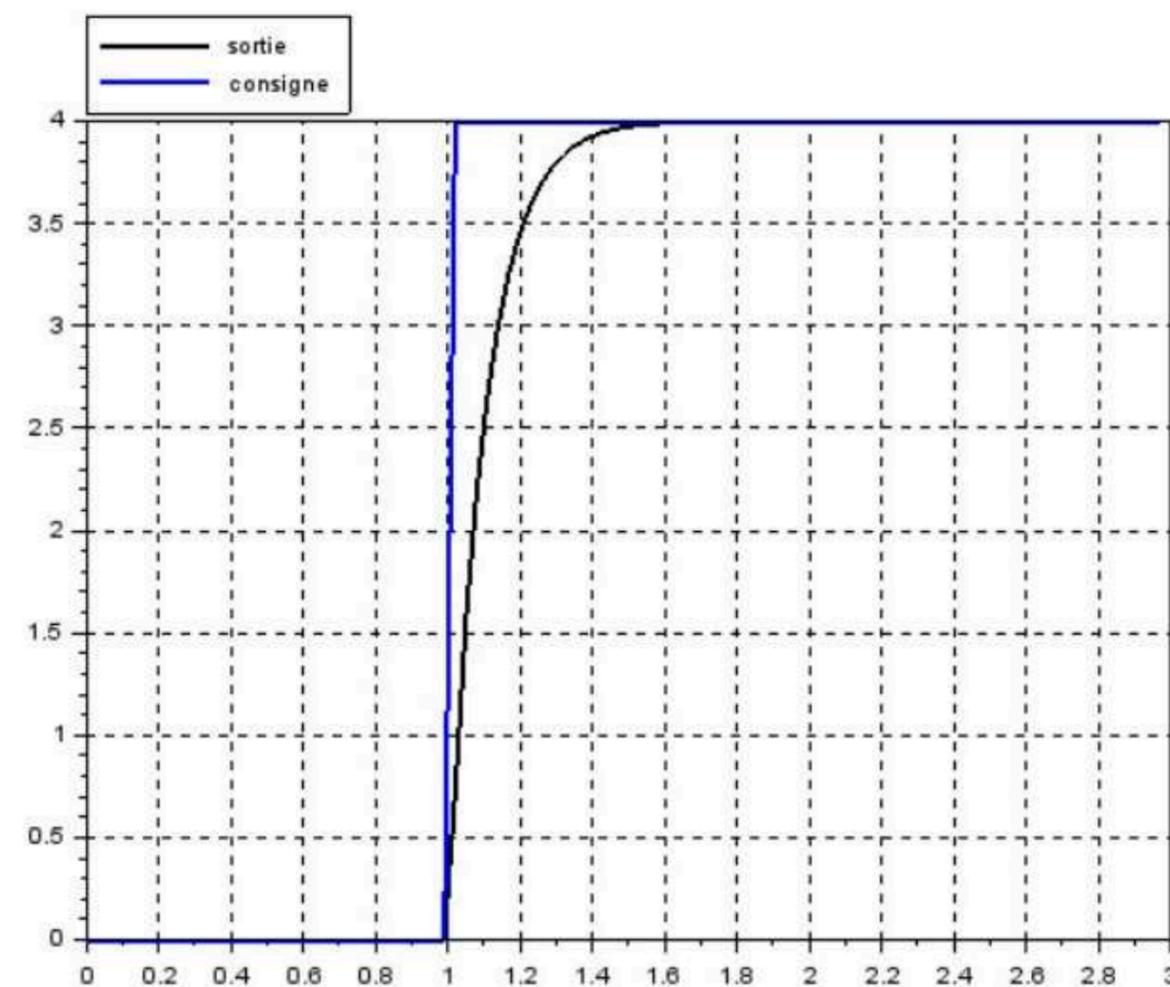


figure 19 :réponse du système corrigé à la consigne donnée

■ Conclusion:

L'erreur statique est éliminée

VI-EQUILIBRAGE DE LA ROUE ARRIERE

Equilibrage de la roue arrière:

● Principe :

Pour réaliser l'équilibrage il faut déterminer :

- Les deux masselottes à ajouter notées **m1** et **m2**
- Les positions angulaires de fixation des masselottes notées **θ_1** et **θ_2**

roue équilibré.SLDASM

Options...

Remplacer les propriétés de masse... Recalculer

Inclure les corps/composants cachés

Afficher la masse du cordon de soudure

Indiquer les valeurs de coordonnées relatives à: -- par défaut --

Propriétés de masse de roue équilibré
Configuration: Défaut
Système de coordonnées: -- par défaut --

Masse = 431.00 grammes

Volume = 4584.41 millimètres cubes

Superficie = 6278.12 millimètres carrés

Centre de gravité: (millimètres)
X = -6.28
Y = -0.06
Z = 29.98

Principaux axes et moments d'inertie: (grammes * millimètres carrés)
Pris au centre de gravité.

lx = (0.50, 0.86, 0.00)	Px = 40608.61
ly = (-0.86, 0.50, 0.00)	Py = 40608.64
lz = (0.00, 0.00, 1.00)	Pz = 62841.58

Moments d'inertie: (grammes * millimètres carrés)
Pris au centre de gravité et aligné avec le système de coordonnées de sortie. (En utilisant la notation de tenseur positive.)

Lxx = 40608.63	Lxy = 0.01	Lxz = 0.02
Lyx = 0.01	Lyy = 40608.62	Lyz = -0.01
Lzx = 0.02	Lzy = -0.01	Lzz = 62841.58

Moments d'inertie: (grammes * millimètres carrés)
Pris au système de coordonnées de sortie. (En utilisant la notation de tenseur positive.)

Aide Imprimer... Copier dans le Presse-papiers



figure 20 :Caractéristiques de la roue

● Conditions d'équilibrage :

■ Equilibrage statique

$$G \in (O, \vec{z})$$

$$\overrightarrow{OG} \cdot \vec{x} = 0$$

$$\overrightarrow{OG} \cdot \vec{y} = 0$$

■ Equilibrage dynamique

(O, \vec{z}) axe principal:

$$I_{xz} = E = 0$$

$$I_{yz} = D = 0$$

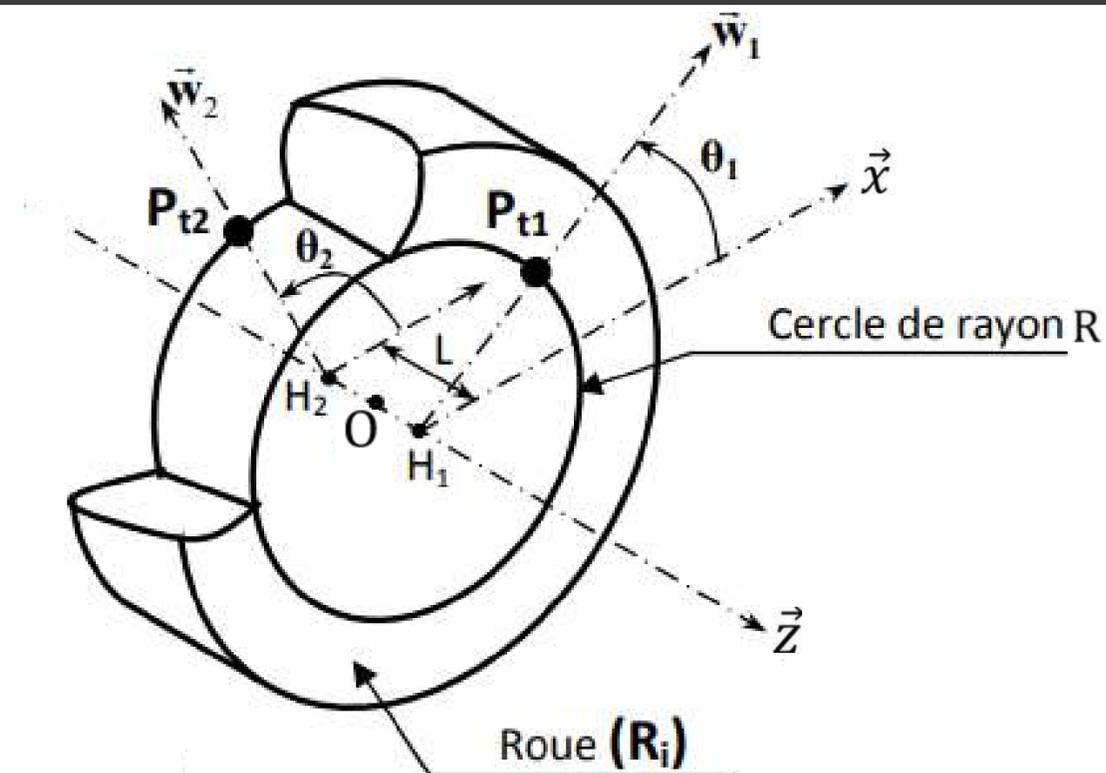
■ Les 4 équations obtenues

$$M_r x + m_1 R \sin \theta_1 + m_2 R \sin \theta_2 = 0$$

$$M_r y + m_1 R \cos \theta_1 + m_2 R \cos \theta_2 = 0$$

$$E + m_1 \frac{L}{2} R \sin \theta_1 - m_2 \frac{L}{2} R \sin \theta_2 = 0$$

$$D + m_1 \frac{L}{2} R \cos \theta_1 - m_2 \frac{L}{2} R \cos \theta_2 = 0$$



■ Matrice d'inertie de la roue :

$$I(O, S) = \begin{bmatrix} A' & -F' & -E' \\ & B' & -D' \\ (sym.) & & C' \end{bmatrix}$$

- **Après les calculs :**

Masselotte 1:

$$\tan \theta_1 = \frac{E + M_r \frac{L}{2} x}{D + M_r \frac{L}{2} y}$$

$$m_1 = \frac{1}{LR} \sqrt{(E + M_r \frac{L}{2} x)^2 + (D + M_r \frac{L}{2} y)^2}$$

Masselotte 2:

$$\tan \theta_2 = \frac{E - M_r \frac{L}{2} x}{D - M_r \frac{L}{2} y}$$

$$m_2 = \frac{1}{LR} \sqrt{(E - M_r \frac{L}{2} x)^2 + (D - M_r \frac{L}{2} y)^2}$$

● Application numérique

$$R=100\text{mm}, L=80\text{ mm}, M_r = 431g$$

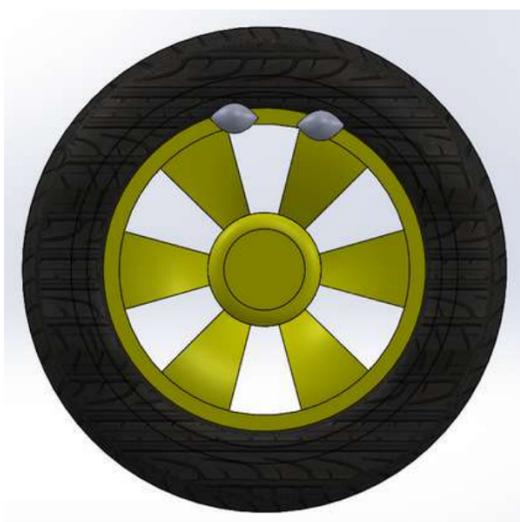
$$m_1 = 13,6\text{ g}$$

$$\theta_1 = 89^\circ$$

$$m_2 = 13,4\text{ g}$$

$$\theta_2 = 95^\circ$$

● Réalisation de l'équilibrage



Masse = 431.00 grammes

Volume = 4584.41 millimètres cubes

Superficie = 6278.12 millimètres carrés

Centre de gravité: (millimètres)

X = -6.28

Y = -0.06

Z = 29.98

figure 21 : propriétés de la roue avant l'équilibrage

● Conclusion

■ Le centre de gravité est porté seulement par l'axe de rotation.

■ L'axe de rotation est un axe principal.

Remplacer les propriétés de masse... Recalculer

Inclure les corps/composants cachés

Créer la fonction Centre de masse

Afficher la masse du cordon de soudure

Indiquer les valeurs de coordonnées relatives à: -- par défaut --

Masse = 458.00 grammes

Volume = 4593.07 millimètres cubes

Superficie = 6304.57 millimètres carrés

Centre de masse: (millimètres)

X = 0.00
Y = 0.00
Z = 25.42

Principaux axes et moments d'inertie: (grammes * millimètres carrés)
Pris au centre de gravité.

lx = (0.00, 0.00, 1.00)	Px = 955.93
ly = (-0.01, -1.00, 0.00)	Py = 4722.73
lz = (1.00, -0.01, 0.00)	Pz = 4926.64

Moments d'inertie: (grammes * millimètres carrés)
Sélectionnés au centre de masse et alignés aux système de coordonnées de sortie. (En utili

Lxx = 4926.61	Lxy = 2.21	Lxz = 0.00
Lyx = 2.21	Lyy = 4722.76	Lyz = 0.00
Lzx = 0.00	Lzy = 0.00	Lzz = 955.93

Moments d'inertie: (grammes * millimètres carrés)
Pris au système de coordonnées de sortie. (En utilisant la notation de tenseur positive.)

lxx = 300875.40	lxy = 2.21	lxz = 0.00
lyx = 2.21	lyy = 300671.55	lyz = 0.00
lzx = 0.00	lzy = 0.00	lzz = 955.93

figure 22 : propriétés de la roue après l'équilibrage

VII-REALISATION

I-CONTEXTUALISAION ET PROBLEMATIQUE

II-ANALYSE FONCTIONNELLE DU ROBOT

III-SUIVI DE LA TRAJECTOIRE

IV-DETECTION DES OBSTACLES

V-CHOIX DU MOTEUR ET ASSERVISSEMENT DE VITESSE

VI-EQUILIBRAGE DE LA ROUE ARRIERE

VII-REALISATION

VIII-CONCLUSION



I-CONTEXTUALISATION ET PROBLEMATIQUE

II-ANALYSE FONCTIONNELLE DU ROBOT

III-SUIVI DE LA TRAJECTOIRE

IV-DETECTION DES OBSTACLES

V-CHOIX DU MOTEUR ET ASSERVISSEMENT DE VITESSE

VI-EQUILIBRAGE DE LA ROUE ARRIERE

VII-REALISATION

VIII-CONCLUSION

CONCLUSION

**MERCI POUR VOTRE
ATTENTION**

ANNEXES

■ PROGRAMME SUIVEUR POUR 2 CAPTEURS

```
void suiveur() {
  valorinfra1 = digitalRead(infra1) ; //capteur à droite
  valorinfra2 = digitalRead(infra2) ;// capteur à gauche

  if (digitalRead(infra1) == LOW) {
    digitalWrite(motor1Pin1,LOW) ;
    digitalWrite(motor1Pin2,LOW) ;}
  else {
    digitalWrite(motor1Pin1,HIGH) ;
    digitalWrite(motor1Pin2,LOW) ;
    analogWrite(3,80) ;
    analogWrite(9,80) ; }

  if (digitalRead(infra2) == LOW) {
    digitalWrite(motor2Pin1,LOW) ;
    digitalWrite(motor2Pin2,LOW) ; }
  else {
    digitalWrite(motor2Pin1,HIGH);
    digitalWrite(motor2Pin2,LOW);
    analogWrite(3,80);
    analogWrite(9,80);
  }
}
```

■ PROGRAMME SUIVEUR POUR 3 CAPTEURS

```
void suiveur2() {
  digitalWrite(infra3) ;//capteur au centre
  digitalWrite(infra1) ;// capteur à droite
  digitalWrite(infra2) ;//capteur à gauche
  if (digitalRead(infra3) == LOW ) //capteur centre sur ligne
  {
    if (digitalRead(infra2) == LOW and digitalWrite(infra1) == HIGH) //Tourner à gauche
      digitalWrite(motor1Pin1, HIGH) ;
      digitalWrite(motor1Pin2, LOW);
      analogWrite(3, 80);
      analogWrite(9, 80);
    if (digitalRead(infra1) == LOW and digitalWrite(infra2) == HIGH) //Tourner à droite
      digitalWrite(motor2Pin1, HIGH) ;
      digitalWrite(motor2Pin2, LOW);
      analogWrite(3, 80);
      analogWrite(9, 80);
    if(digitalRead(infra1) == LOW and digitalWrite(infra2) == LOW)// Avancer
      digitalWrite(motor1Pin1, HIGH) ;
      digitalWrite(motor1Pin2, LOW);
      digitalWrite(motor2Pin1, HIGH) ;
      digitalWrite(motor2Pin2, LOW);
      analogWrite(3, 80);
      analogWrite(9, 80);}
  }
```

ANNEXES

■ PROGRAMME SUIVEUR POUR 3 CAPTEURS(suite):

```
else // capteur centre sur piste
{
  if (digitalRead(infra1) == LOW and digitalRead(infra2) == HIGH)//Tourner à droite
  digitalWrite(motor1Pin1, HIGH) ;
  digitalWrite(motor1Pin2, LOW);
  analogWrite(3, 80);
  analogWrite(9, 80);
  if (digitalRead(infra1) == HIGH and digitalRead(infra2) == LOW)//Tourner à gauche
  digitalWrite(motor2Pin1, HIGH) ;
  digitalWrite(motor2Pin2, LOW);
  analogWrite(3, 80);
  analogWrite(9, 80);
  if(digitalRead(infra2) == HIGH and digitalRead(infra1) == HIGH) // Reculer
  digitalWrite(motor1Pin1, LOW) ;
  digitalWrite(motor1Pin2, HIGH);
  digitalWrite(motor1Pin1, LOW) ;
  digitalWrite(motor1Pin2, HIGH);
  analogWrite(3, 80);
  analogWrite(9, 80);
}}
```

■ PROGRAMME PRINCIPALE ET EVITEUR D'OBSTACLES:

```
void loop() {
  {
  digitalWrite(trig,1) ;
  delayMicroseconds(10);
  digitalWrite(trig,0);
  lecteur_echo = pulseIn(echo,1) ;
  distance=lecteur_echo*0.017 ;
  Serial.print("la distance est :") ;
  Serial.print(distance);
  Serial.println("cm");
  if (distance <= 100)
  {
  obstacle() ;
  }
  else
  {
  suiveur() ;
  } } }
```

```
void obstacle() {
  digitalWrite(motor1Pin1,LOW)
  digitalWrite(motor1Pin2,LOW)
  digitalWrite(motor2Pin1,LOW)
  digitalWrite(motor2Pin2,LOW)
```