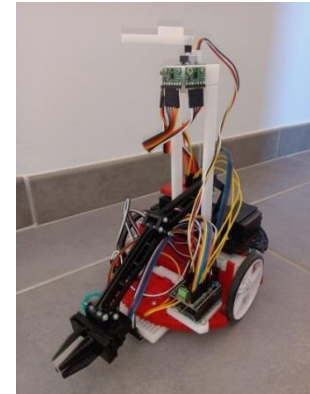
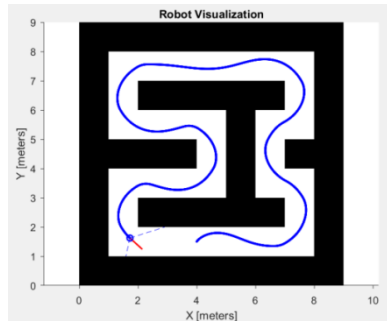


Intelligence artificielle au service de la mobilité

Projet / Activité

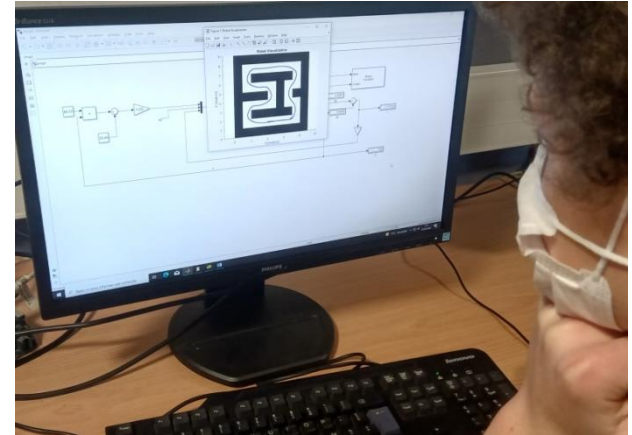


Programme et prérequis

- Le programme des Sciences de l'ingénieur indique :

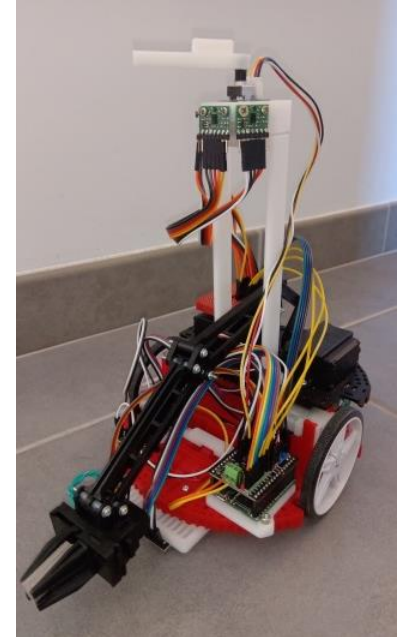
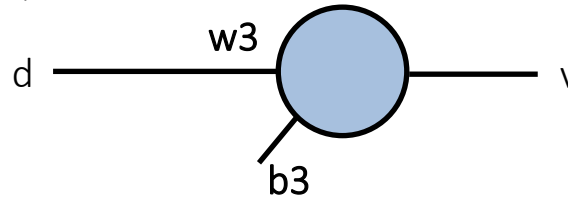
Analyser le traitement de l'information	Algorithme, programme Langage informatique Notions sur l'intelligence artificielle	T ^{ale}
---	--	------------------

- Prérequis pour ce projet / activité :
 - Notion de fonction affine
 - Notion de modélisation
 - Notions de base du langage Python (`if`, `while`, `for`, `def`) et notion de liste (`list`)
 - Notion d'algorithme



Problématique

- Modéliser puis mettre en œuvre une base robotique mobile, dotée d'intelligence artificielle, capable de se déplacer de façon autonome.
- Décrire le comportement de cette base robotique à l'aide de perceptrons (de neurones) :
- Mettre en place un apprentissage supervisé afin d'établir le comportement à adopter lors des déplacements.



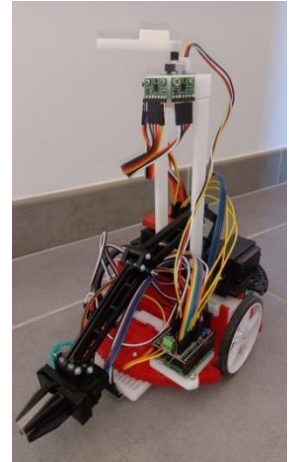
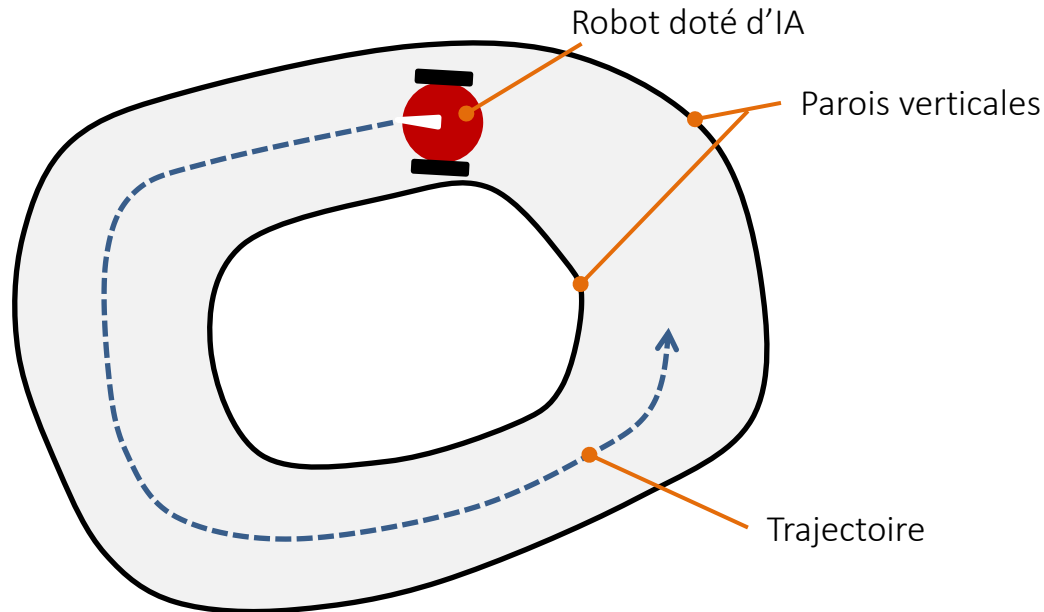
Etapes suivies par les élèves

- Alternance entre des phases de projet ■ et des phases d'activité ■ .
- Etapes suivies par les élèves :
 1. Notions de perceptron et d'apprentissage supervisé ;
 2. Evitement de collision frontale lors d'un mouvement de translation ;
 3. Positionnement angulaire dans une direction ne présentant pas d'obstacle ;
 4. Déplacement afin d'effectuer la totalité du parcours.
- Chaque étape comporte une phase d'analyse, de modélisation puis d'expérimentation.



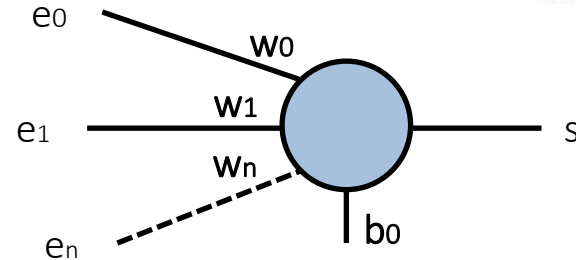
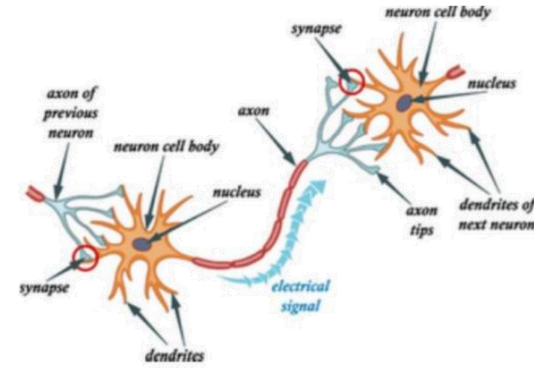
Performances attendues

- Parcourir, à l'issue d'une phase d'apprentissage, un tracé donné sans rentrer en collision avec une paroi verticale :



Partie 1 - Notion d'intelligence artificielle

- L'IA consiste à « mimer » les phénomènes biologiques (jumeau biologique).
- Notions :
 - d'entrées / sortie d'un neurone (perceptron) ;
 - de poids et d'entrée pondérée ;
 - de biais ;
 - de relation d'entrées / sortie d'un perceptron.



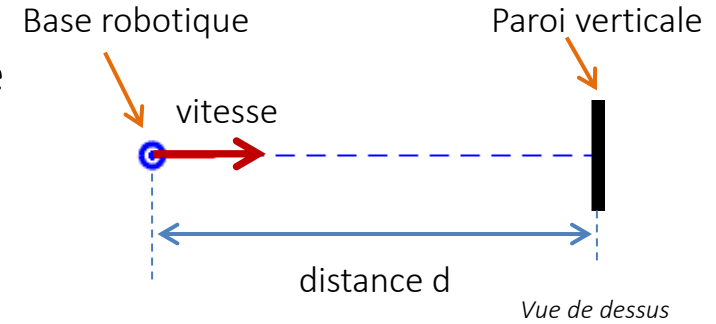
$$s = w_0 \times e_0 + w_1 \times e_1 + \dots + w_n \times e_n + b_0$$

w_0, w_1, w_n : poids

b_0 : biais

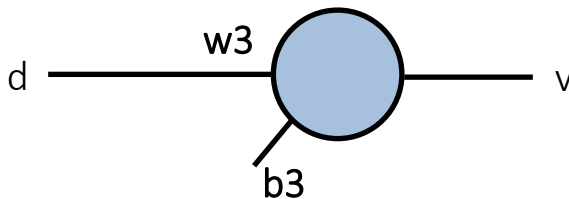
Partie 2 - Evitement de collision frontale

- Objectif pour l'élève dans cette partie :
 - mettre en place un perceptron pour permettre un déplacement en translation en direction d'une paroi mais sans rentrer en collision.



- Solution proposée à l'élève :


- mesurer la distance d à l'obstacle à l'aide d'un capteur de distance ;
- adapter la vitesse en fonction de la distance d selon une fonction affine :



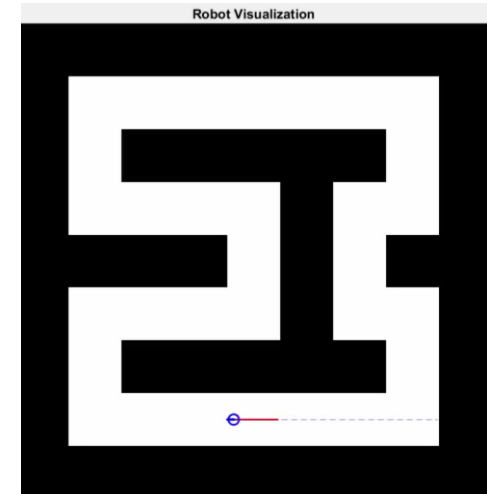
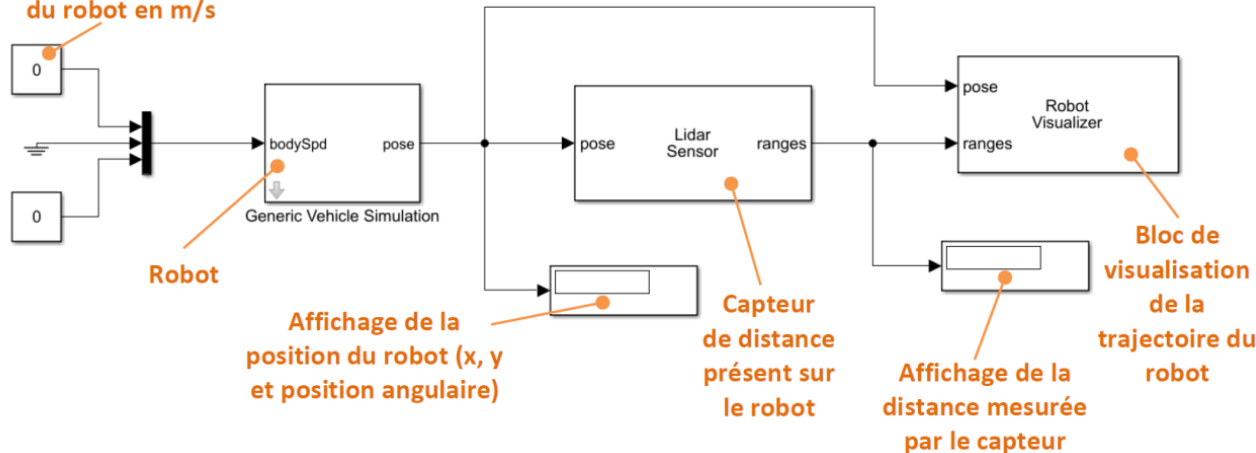
$$v = w_3 \times d + b_3$$

w_3 : poids b_3 : biais

Partie 2 - Evitement de collision frontale

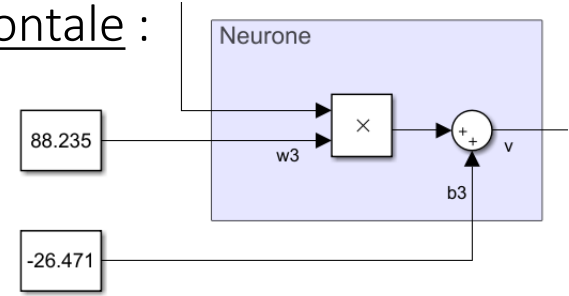
- Modélisation 1 par l'élève :
 - modéliser, sur Matlab / Simulink, le comportement du robot avec un module spécifique :  Mobile Robotics Simulation Toolbox Version 2_1.mltbx

Vitesse de déplacement
du robot en m/s

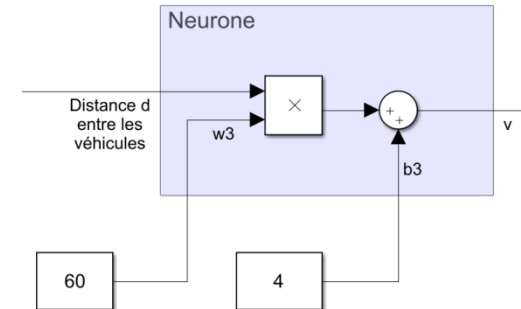
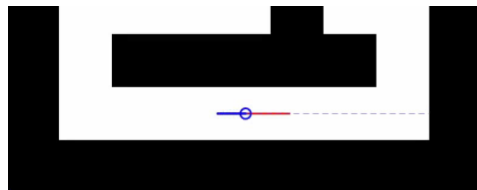


Partie 2 - Evitement de collision frontale

- Modélisation 2 par l'élève :
 - ajouter le perceptron d'évitement de collision frontale :



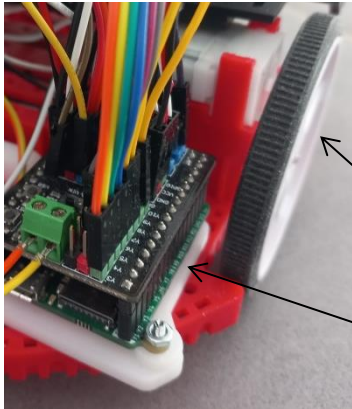
- étudier qualitativement les conséquences de la modification du poids et du biais :



Partie 2 - Evitement de collision frontale



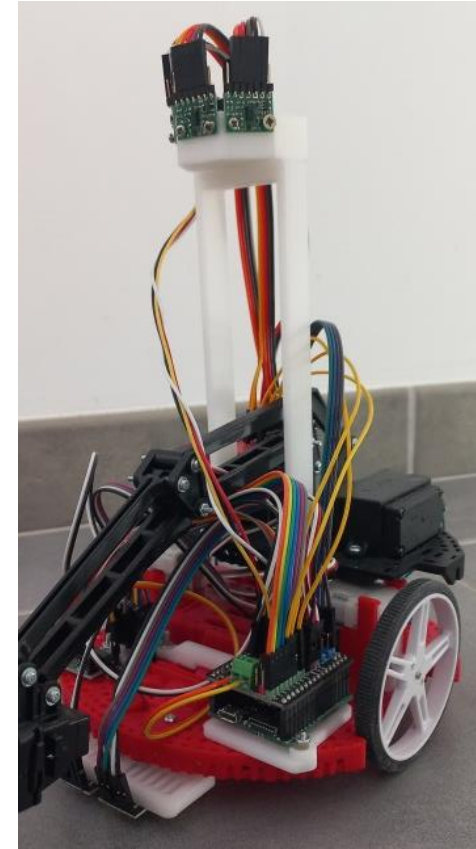
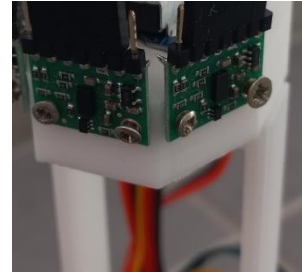
- Expérimentation par l'élève :
 - mettre en place le perceptron d'évitement de collision dans le programme Python de la base robotique ;
 - utiliser un capteur de distance :



Motorisation

Pyboard

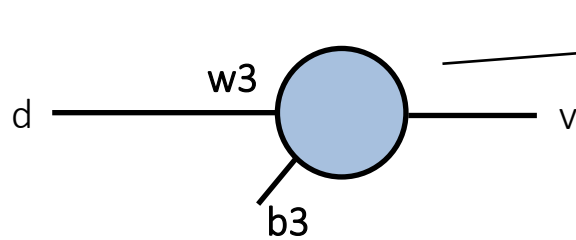
Capteur de distance →
(les capteurs gauche et droite ne sont pas utilisés dans cette partie)



Partie 2 - Evitement de collision frontale



- mettre en place un perceptron en python (programme à compléter) :



$$v = w_3 \times d + b_3 \quad \text{avec :} \quad w_3 = 88,23$$
$$b_3 = -26,47$$

- commander la motorisation de la base robotique de façon appropriée :

```
tof1_centre.start()
```

```
while True:
```

```
    distance = tof1_centre.read()/1000
```

```
    w3 = 88.23
```

```
    b3 = -26.47
```

```
    vitesse = w3 * distance + b3
```

```
    if vitesse >= 0:
```

```
        Mot_A_Sens.high()
```

```
        Mot_B_Sens.high()
```

```
    else:
```

```
        vitesse = vitesse * -1
```

```
        Mot_A_Sens.low()
```

```
        Mot_B_Sens.low()
```

```
    Mot_A_Vitesse.pulse_width_percent(int(vitesse))
```

```
    Mot_B_Vitesse.pulse_width_percent(int(vitesse))
```

```
    time.sleep_ms(10)
```

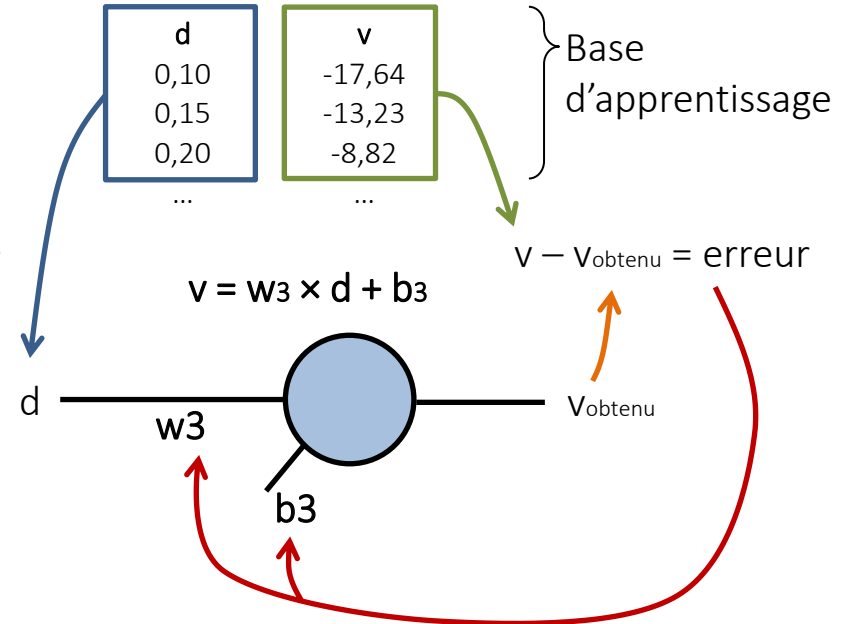
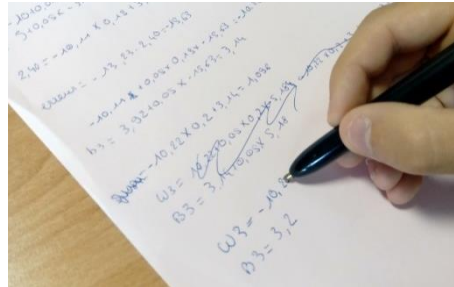
Partie 2 - Evitement de collision frontale

- faire des essais réels sur la base robotique :

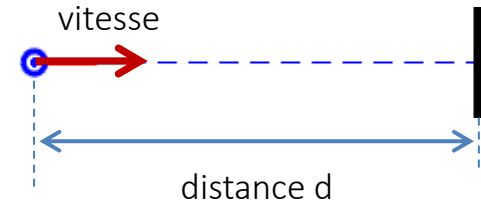


Partie 2_b : Apprentissage supervisé

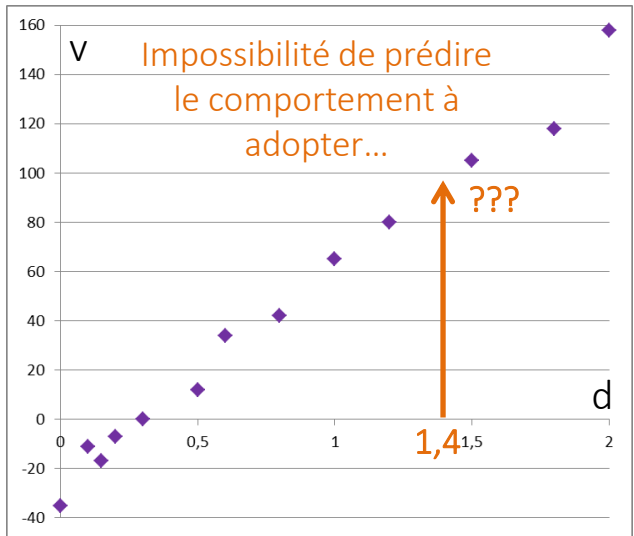
- Analyse par l'élève :
 - mettre en place l'apprentissage supervisé du perceptron d'évitement de collision frontale.
 - découvrir un algorithme de recherche itérative d'une droite ajustée aux couples (d, v) présents dans la base d'apprentissage.



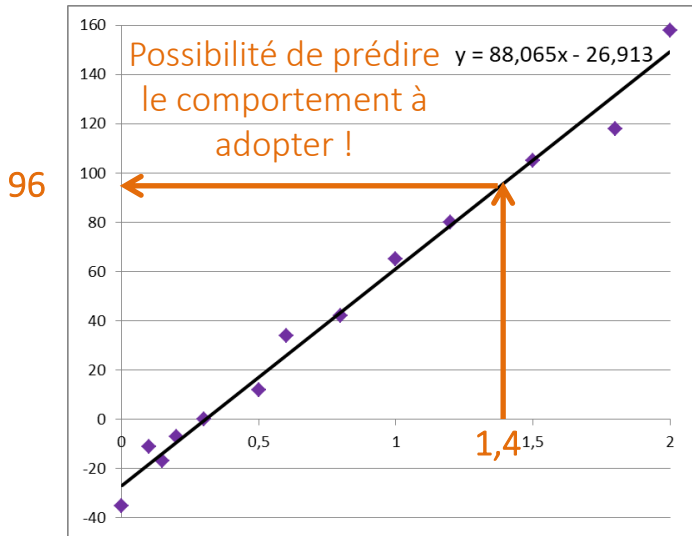
Partie 2_b : Apprentissage supervisé



- Analyse par l'élève :
 - déterminer les valeurs du poids w_3 et du biais b_3 en recherchant la droite qui s'ajuste le mieux possible aux couples (d, v) .



AVANT l'apprentissage



APRES l'apprentissage

La base robotique va apprendre à prédire la vitesse à laquelle elle devra se déplacer pour une distance d donnée.

Partie 2_b : Apprentissage supervisé

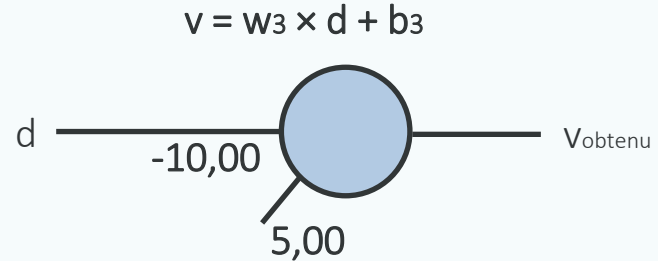
- Extrait de l'activité :

Le principe de l'apprentissage supervisé est le suivant.

On choisit arbitrairement la valeur du poids w_3 et du biais b_3 :

$$w_3 = -10,00 \text{ et } b_3 = 5,00$$

d	v
0,10	-17,64
0,15	-13,23
0,20	-8,82
...	...



Partie 2_b : Apprentissage supervisé

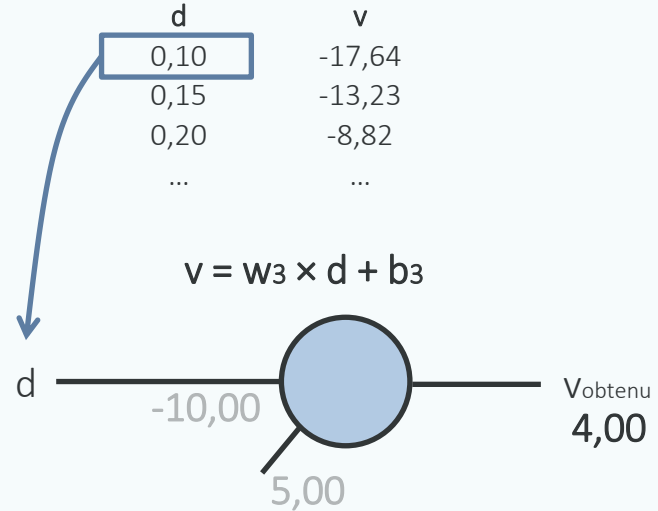
- Extrait de l'activité :

1- On soumet le perceptron au premier couple (distance / vitesse) de la base d'apprentissage.

Puis on calcule la valeur v associée :

$$V_{\text{obtenu}} = -10,00 \times d + 5,00 = 4,00$$

On parle ici de propagation avant.

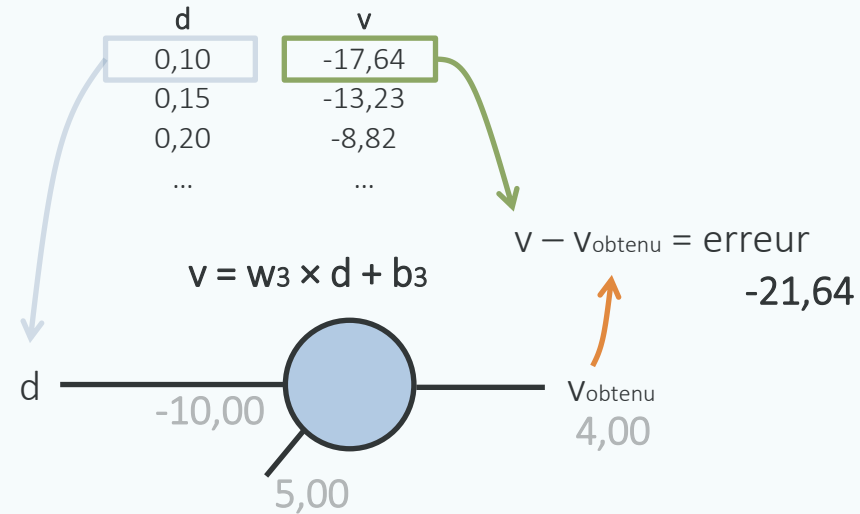


Partie 2_b : Apprentissage supervisé

- Extrait de l'activité :

2- On calcule l'erreur entre la sortie v_{obtenu} du neurone et la valeur attendue v présente dans la base d'apprentissage :

$$\text{erreur} = -17,64 - 4,00 = -21,64$$



Partie 2_b : Apprentissage supervisé

- Extrait de l'activité :

3- On recalcule une nouvelle valeur de w_3 et de b_3 à l'aide des relations suivantes :

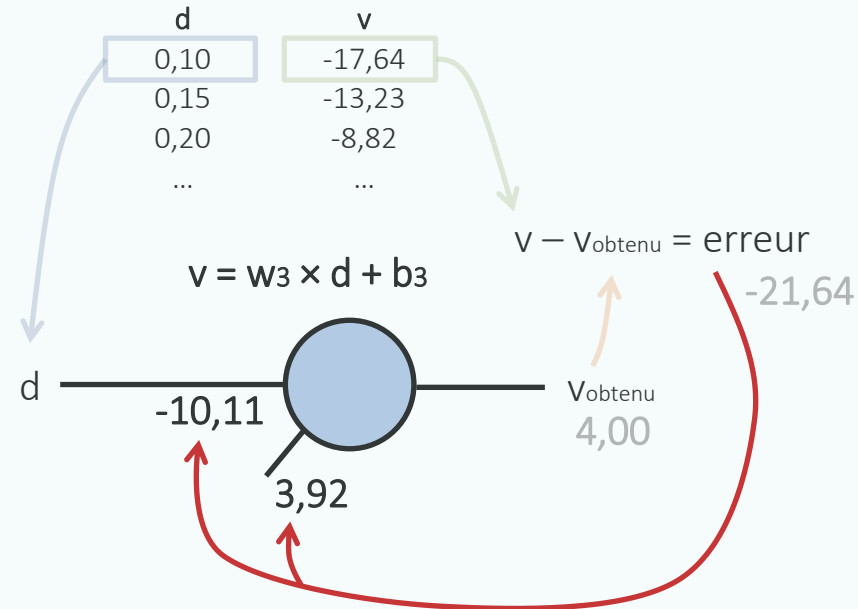
$$w_3 = w_3 + 0,05 \times \text{distance} \times \text{erreur}$$

$$b_3 = b_3 + 0,05 \times \text{erreur}$$

$$w_3 = -10,00 + 0,05 \times 0,10 \times -21,64 = \underline{\underline{-10,11}}$$

$$b_3 = 5,00 + 0,05 \times -21,64 = \underline{\underline{3,92}}$$

On parle ici de rétro-propagation.



Partie 2_b : Apprentissage supervisé

- Analyse par l'élève :
 - utiliser un tableur afin d'itérer à plusieurs reprises sur la base d'apprentissage :

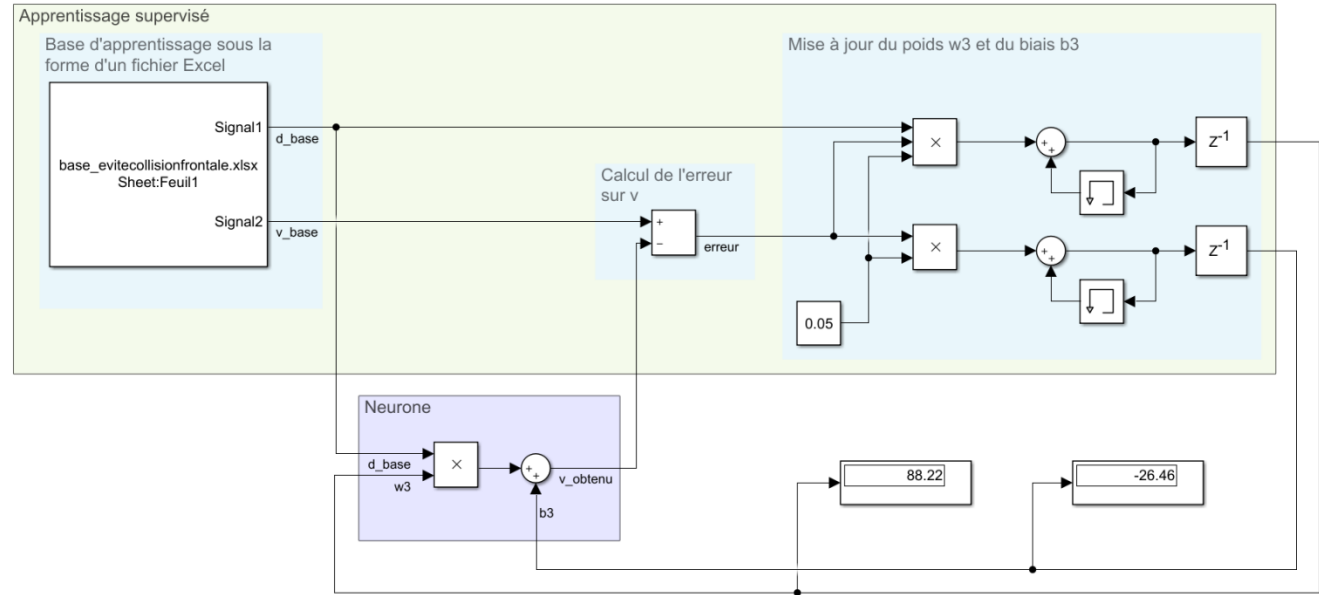
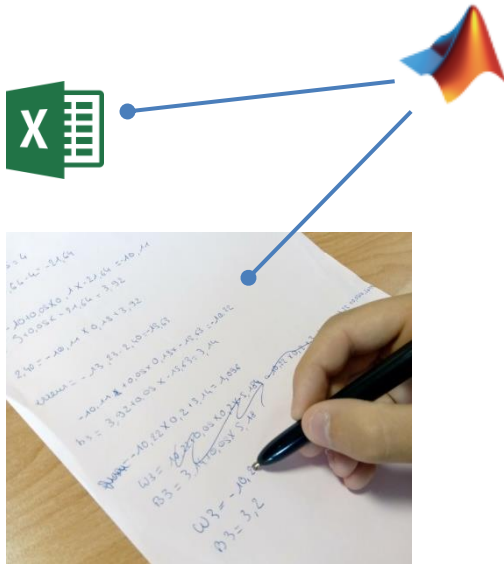
base d'apprentissage		itération	Vobtenu	erreur	w3	b3
d	v				-10	5
0,1	-17,6471	1	4	-21,6471	-10,10824	3,917645
0,15	-13,2353	2	2,4014097	-15,63671	-10,22551	3,1358095
0,2	-8,82353	3	1,0907074	-9,914237	-10,32465	2,6400976
0,3	0	4	-0,457298	0,4572983	-10,31779	2,6629626
0,5	17,64706	5	-2,495934	20,142994	-9,814219	3,6701123
0,6	26,47059	6	-2,218419	28,689009	-8,953549	5,1045627
0,8	44,11765	7	-2,058276	46,175926	-7,106512	7,413359
1	61,76471	8	0,3068475	61,457863	-4,033618	10,486252
1,2	79,41176	9	5,6459101	73,76585	0,3923326	14,174545
1,5	105,8824	10	14,763044	91,119356	7,2262843	18,730512
1,8	132,3529	11	31,737824	100,61508	16,281641	23,761266
2	150	12	56,324549	93,675451	25,649186	28,445039
...
		735	-8,802036	-0,021494	88,21364	-26,44588



- déterminer le nombre d'itérations nécessaires afin que l'erreur sur la valeur de v en sortie du perceptron soit « acceptable ».

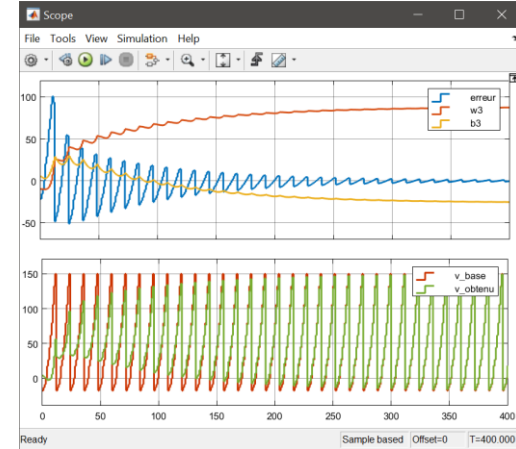
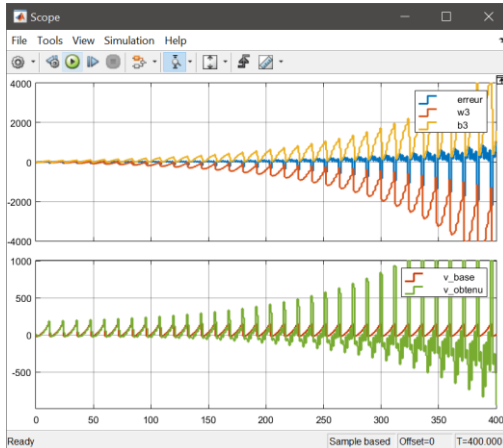
Partie 2_b : Apprentissage supervisé

- Modélisation par l'élève :
 - mettre en place un modèle d'apprentissage, sur Matlab / Simulink, afin de valider les résultats issus de l'analyse théorique et du tableur :



Partie 2_b : Apprentissage supervisé

- Analyse par l'élève des compromis entre :
 - le nombre d'itérations et l'erreur après l'apprentissage ;
 - le nombre d'itérations nécessaires et le temps de calcul lors de l'apprentissage.



- Analyse par l'élève :
 - de l'influence de la valeur de la constante d'apprentissage sur le temps et la qualité de l'apprentissage.

Partie 2_b : Apprentissage supervisé

- Expérimentation par l'élève :
 - mettre en place l'apprentissage supervisé en python sur la base robotique
(programme à compléter)

base_apprentissage =

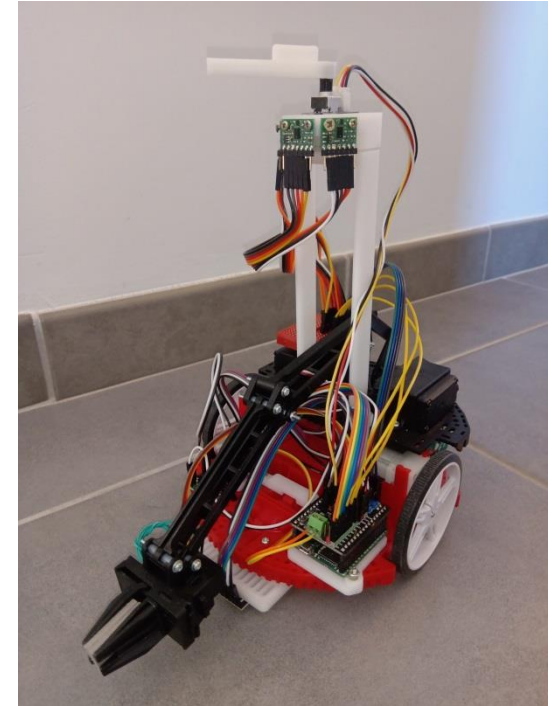
```
[[0.1, -17.64],  
 [0.15, -13.23],  
 [0.1, -8.82],  
 [0.3, 0],  
 [0.5, 17.64],  
 [0.6, 26.47],  
 [0.8, 44.11],  
 [1, 61.76],  
 [1.2, 79.71],  
 [1.5, 105.88],  
 [1.8, 132.35],  
 [2, 150]]
```

```
def apprentissage_supervisé(it, w3, b3, mu):  
    for i in range(it//len(base_apprentissage)):  
        for ligne_base in range(len(base_apprentissage)):  
            # Propagation avant  
            v_obtenu = w3 * base_apprentissage[ligne_base][0] + b3  
  
            # Calcul de l'erreur  
            erreur = base_apprentissage[ligne_base][1] - v_obtenu  
  
            # Rétropropagation des gradients d'erreur  
            w3 = w3 + mu * base_apprentissage[ligne_base][0] * erreur  
            b3 = b3 + mu * erreur  
  
    return w3, b3
```



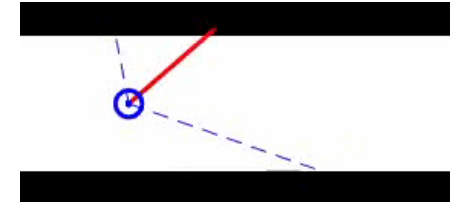
Partie 2_b : Apprentissage supervisé

- Expérimentation par l'élève
 - faire des essais sur la base robotique avec les valeurs de w_3 et b_3 issues de l'apprentissage supervisé :

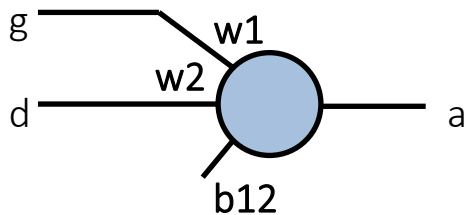


Partie 3 : Positionnement angulaire

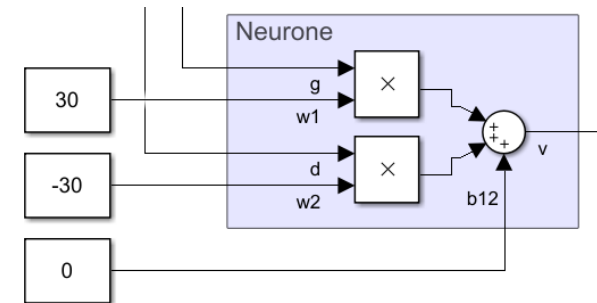
- Objectif pour l'élève dans cette partie :
 - permettre un positionnement angulaire dans une direction dénuée d'obstacle.



- Solution proposée à l'élève :
 - utiliser deux capteurs de distance gauche et droit positionnés à 60° ;
 - utiliser un perceptron qui délivre sur sa sortie la vitesse angulaire du robot.



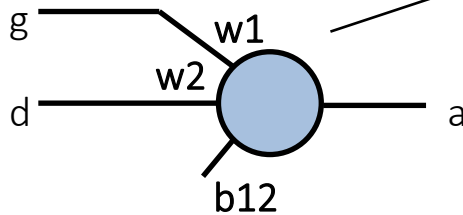
$$a = w_1 \times g + w_2 \times d + b_{12}$$



Partie 3 : Positionnement angulaire



- Expérimentation par l'élève :
 - mettre en place le perceptron de positionnement angulaire en python :



```
tof1_gauche.start()  
tof1_droit.start()
```

```
while True:  
    gauche = tof1_gauche.read()/1000  
    droit = tof1_droit.read()/1000
```

```
w1 = -30  
w2 = 30  
b12 = 0  
a = w1 * gauche + w2 * droit + b12
```

```
if a > 30:  
    a = 30  
if a < -30:  
    a = -30
```

- commander la motorisation de la base robotique de façon appropriée :

```
Mot_A_Sens.high()  
Mot_B_Sens.high()  
Mot_A_Vitesse.pulse_width_percent(40 + int(a*2/3))  
Mot_B_Vitesse.pulse_width_percent(40 - int(a*2/3))  
  
time.sleep_ms(10)
```

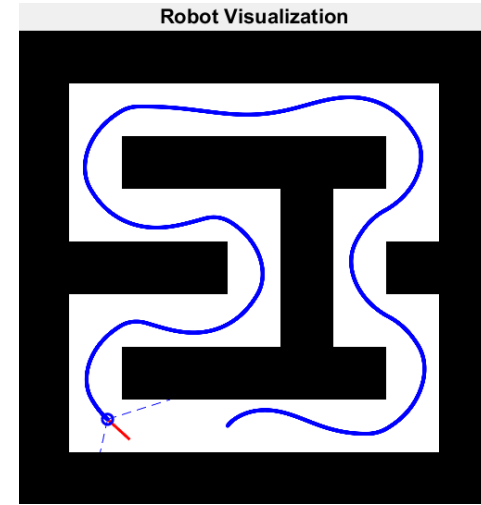
Partie 3 : Positionnement angulaire

- faire des essais réels sur la base robotique :

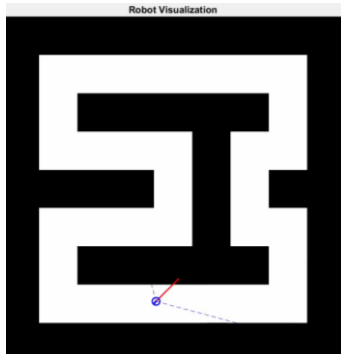


Partie 4 : Déplacement autonome

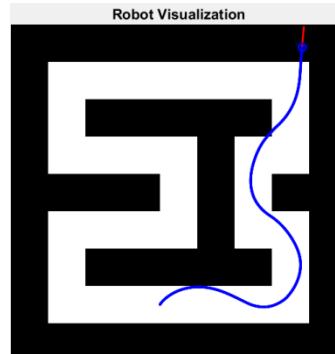
- Modélisation par l'élève :
 - analyser la modification empirique :
 - de la valeur des poids et / ou du biais ;
 - de la vitesse de déplacement.
 - mettre en évidence la nécessité d'un perceptron en plus pour gagner du temps.



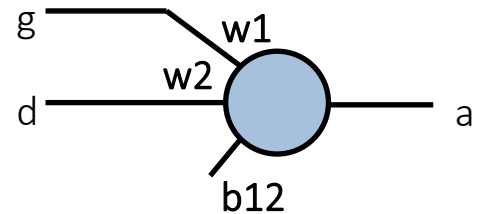
$$\begin{aligned}w_1 &= 30 \\w_2 &= -30 \\b_{12} &= 0 \\v &= 0,1\text{m/s}\end{aligned}$$



$$\begin{aligned}w_1 &= 60 \\w_2 &= -30 \\b_{12} &= 0 \\v &= 0,1\text{m/s}\end{aligned}$$

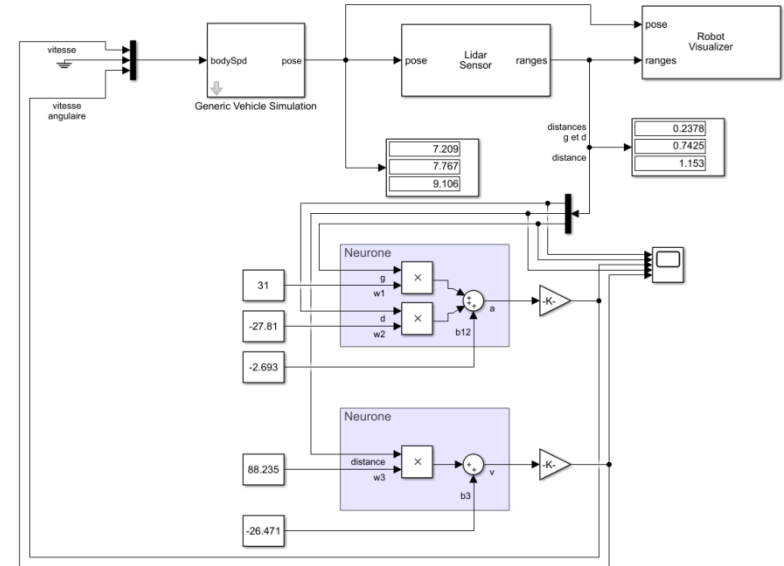
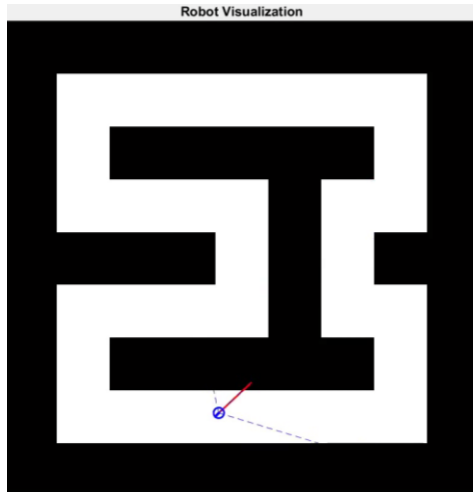


$$\begin{aligned}w_1 &= 30 \\w_2 &= -30 \\b_{12} &= 0 \\v &= 0,2\text{m/s}\end{aligned}$$



Partie 4 : Déplacement autonome

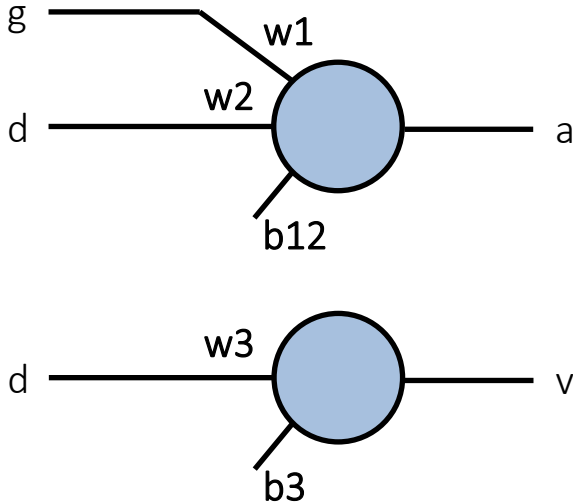
- Modélisation élève :
 - mettre en place les deux perceptrons pour essayer de réduire le temps de parcours :



- Prolongement à prévoir : procéder à un apprentissage supervisé commun au deux perceptrons.

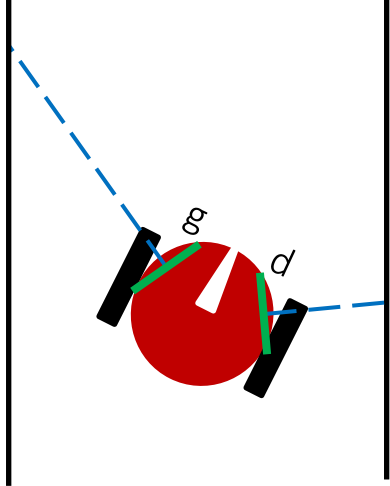
Partie 4 : Déplacement autonome

- Expérimentation par l'élève :
 - faire des essais sur la base robotique avec l'implémentation des deux perceptrons :

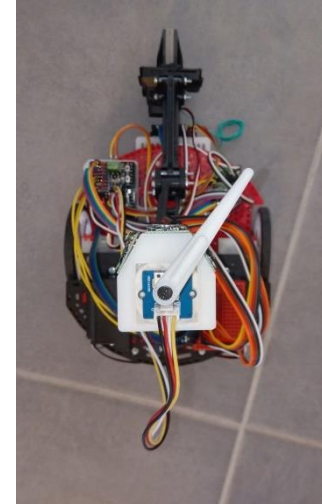


Partie 5 : Apprentissage in situ

- Expérimentation par l'élève :
 - mettre en place un apprentissage in situ afin de concevoir la base d'apprentissage puis d'exécuter l'apprentissage supervisé directement sur la base robotique.

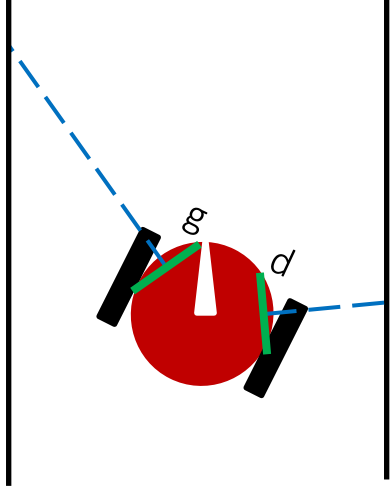


- utiliser un potentiomètre dotée d'une aiguille afin de pointer la direction à atteindre pour une position angulaire donnée.

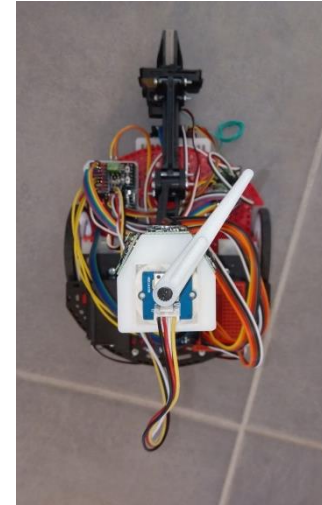


Partie 5 : Apprentissage in situ

- Expérimentation par l'élève :
 - mettre en place un apprentissage in situ afin de concevoir la base d'apprentissage puis d'exécuter l'apprentissage supervisé directement sur la base robotique.

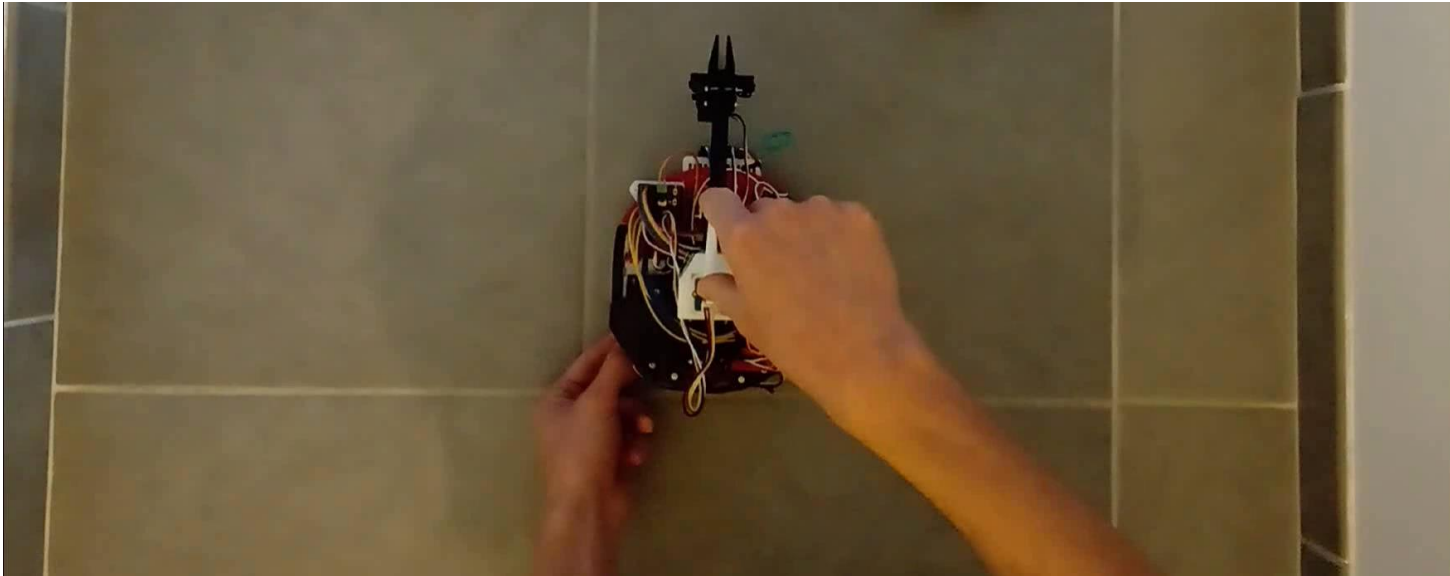


- utiliser un potentiomètre dotée d'une aiguille afin de pointer la direction à atteindre pour une position angulaire donnée.



Partie 5 : Apprentissage in situ

- Expérimentation par l'élève :
 - faire des essais de l'apprentissage in situ :



En espérant que ces ressources
vous seront utiles.

Merci pour votre écoute !