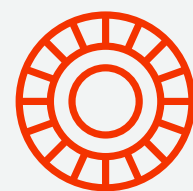




Evitement d'obstacles pour le robot e-puck en utilisant un réseau de neurones artificiels.



Réalisé par : ARGUI Imane – CHAKRAA Hamza.
Encadré par : Mr.CHERUBINI Andrea.

Plan

01

Objectif du
projet

Évitement d'obstacles
d'un robot mobile par
un réseau de
neurones artificiels.

02

Matériel
utilisé

Robot mobile e-
puck.
Python.
CoppeliaSim.

03

Réseau de
neurones

Réseau de
perceptron.
Algorithme de Hebb.
Algorithme de
descente de
gradient.

04

Résultats et
conclusion

Résultats de
simulation.
Remarques et
perspectives.



Objectif du projet

APPROCHE

RÉACTIVE

SIMPLE

- ✓ PAS DE LOCALISATION.
- ✓ PAS DE CARTOGRAPHIE.
- ✓ PAS D'OBJECTIFS À ATTEINDRE.

OBSTACLES

STATIQUES.

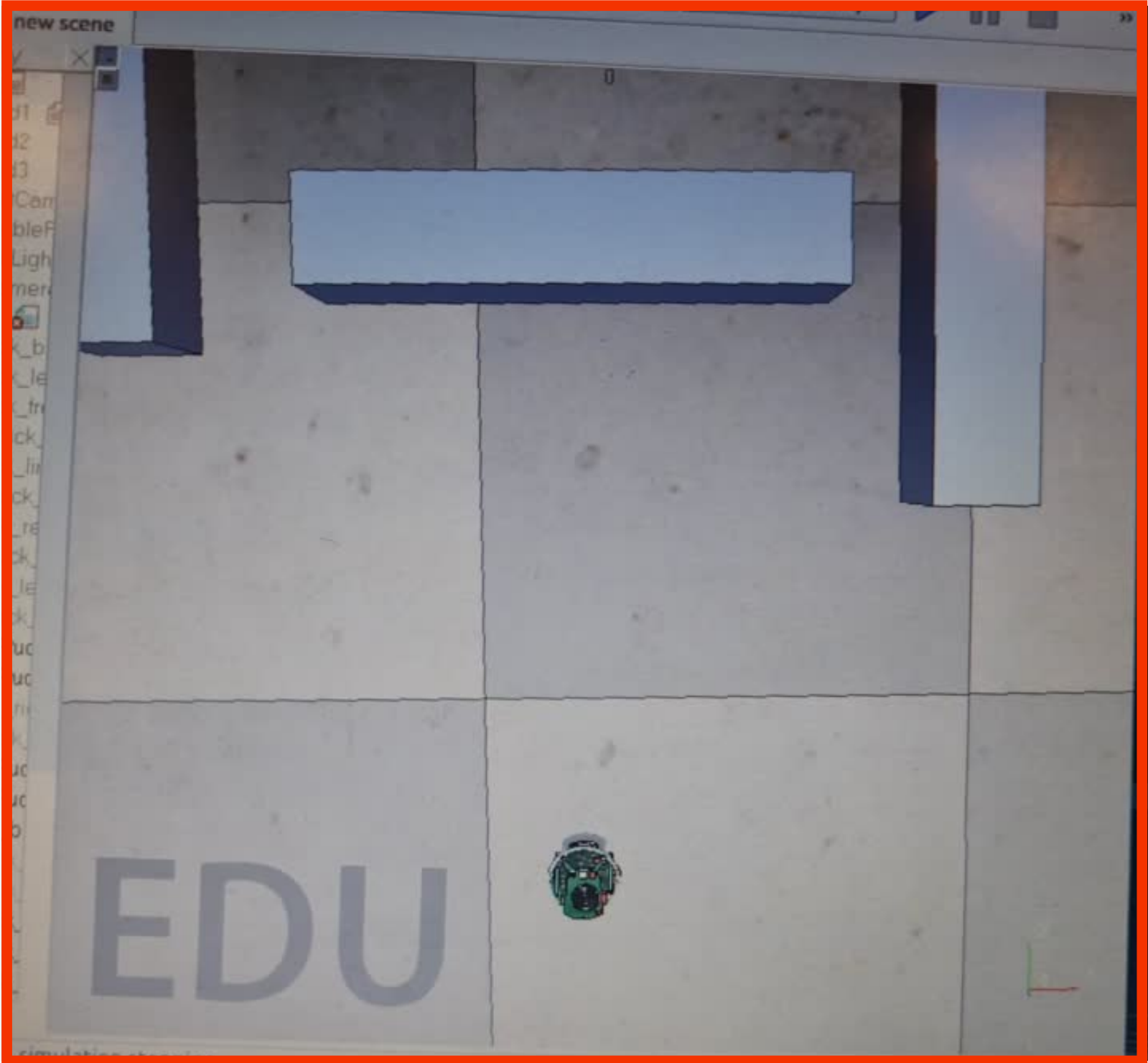
A DÉCOUVRIR EN
TEMPS RÉEL.

APPLICATIONS

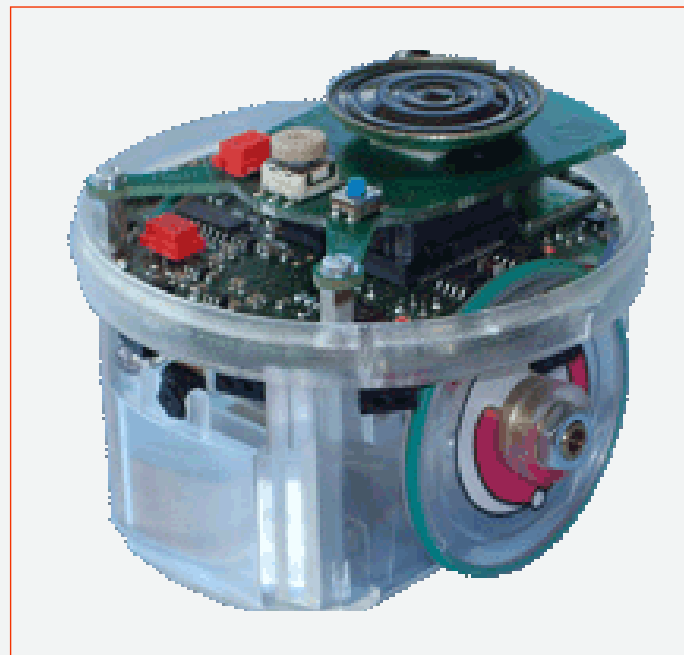
- ✓ ASPIRATEURS.
- ✓ ROBOTS SUR CARTE.
- ✓ EXPLORATION (MER, PERSEVERANCE).



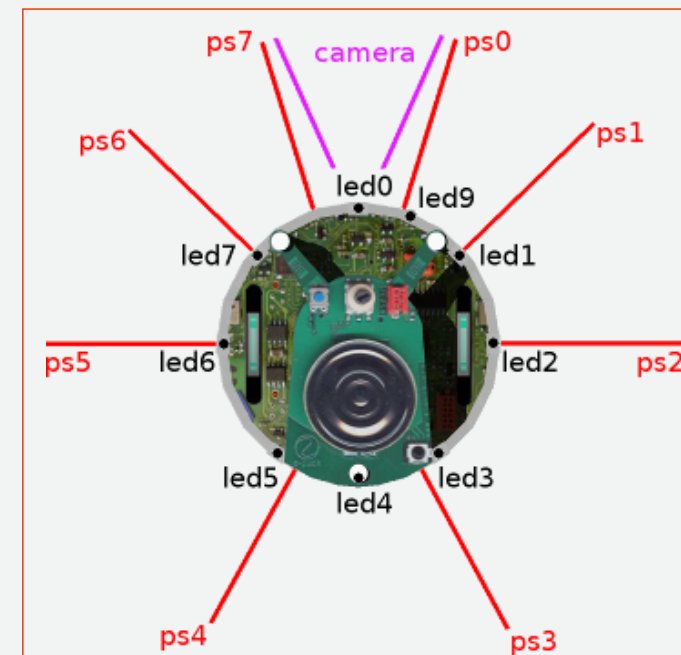
Résultat souhaité



Le robot e-puck

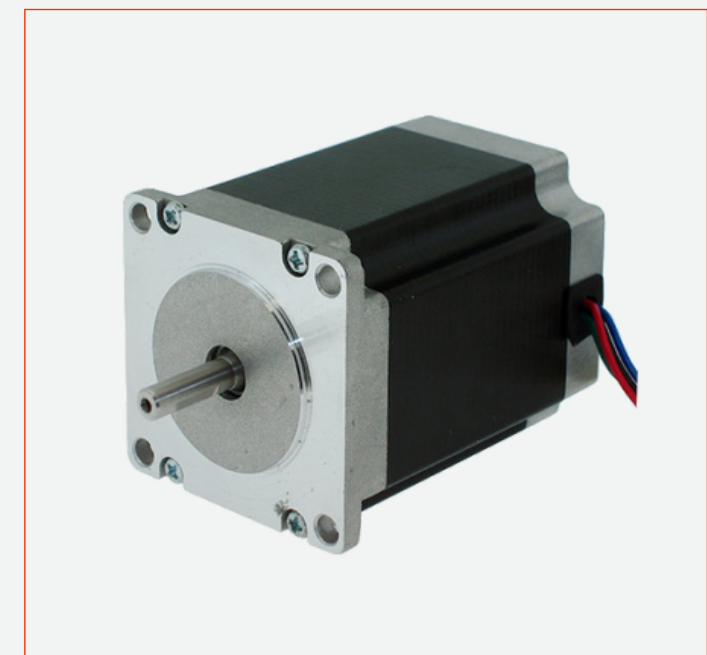


ROBOT MOBILE À ROUES.



8 CAPTEURS INFRAROUGES.

Distance de détection : 0.2 à 4 mm.
Les données des capteurs : Les entrées du système.

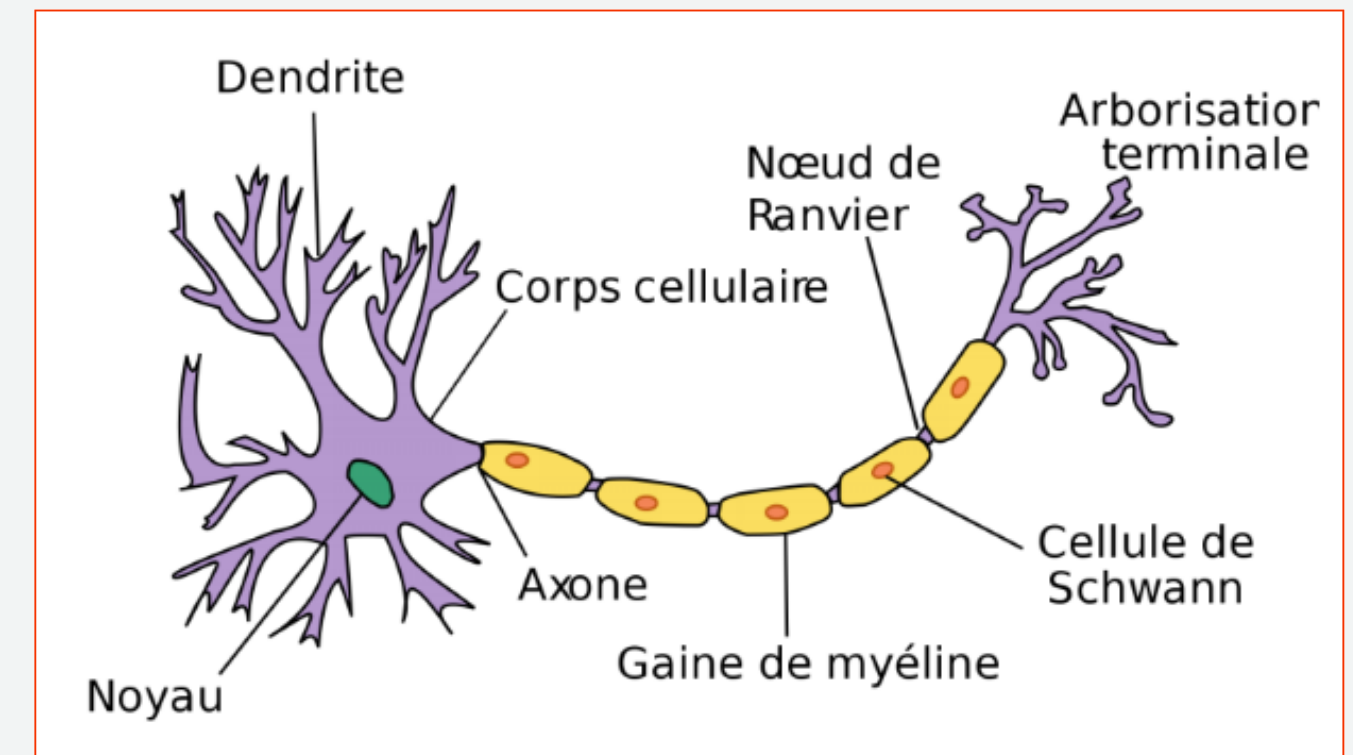


2 MOTEURS PAS À PAS.

La vitesse des roues : La sortie du système.

Le système nerveux

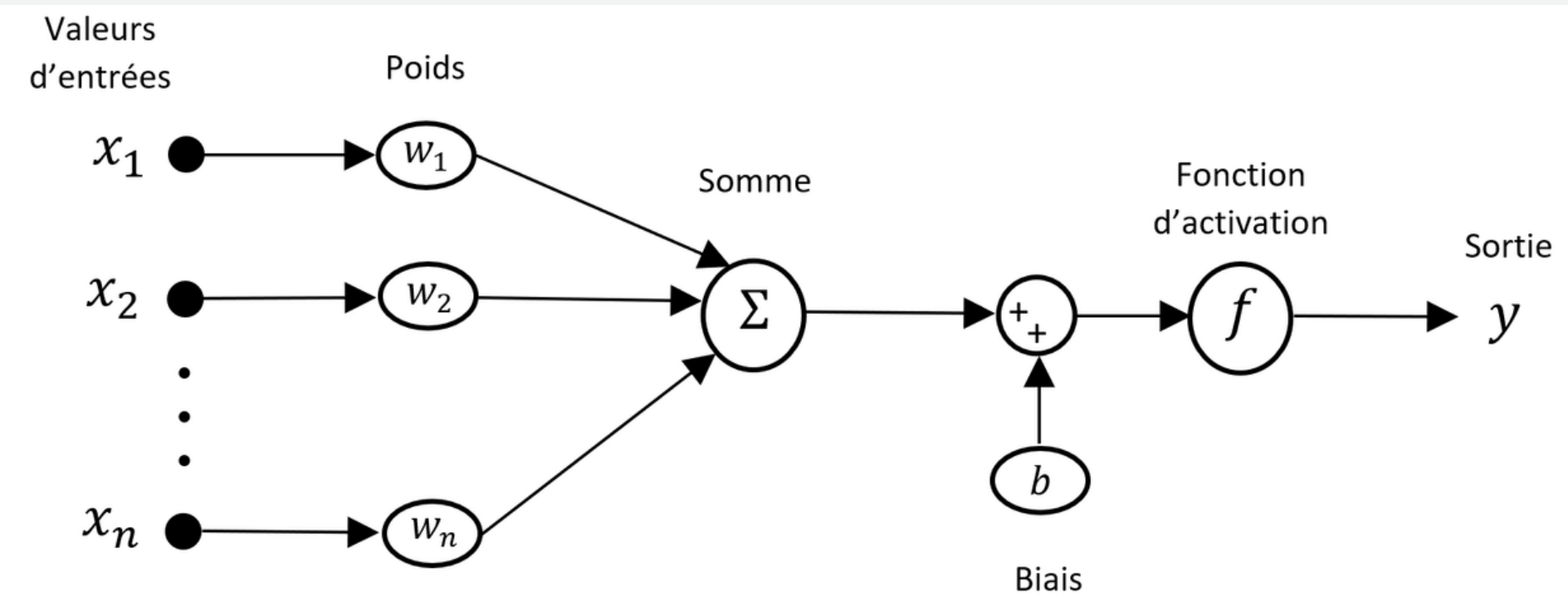
- Principales cellules : les neurones.
- Axones connectées aux dendrites.
- Impulsions envoyées :
 - Entrée : Dendrite.
 - Sortie : Axone.



Structure d'un neurone.

Réseau de perceptron

- ✓ LE PREMIER RÉSEAU DE NEURONES.
- ✓ UN NEURONE FORMEL.
- ✓ UN RÉSEAU MONOCOUCHE.
- ✓ $Y = W.X + B$

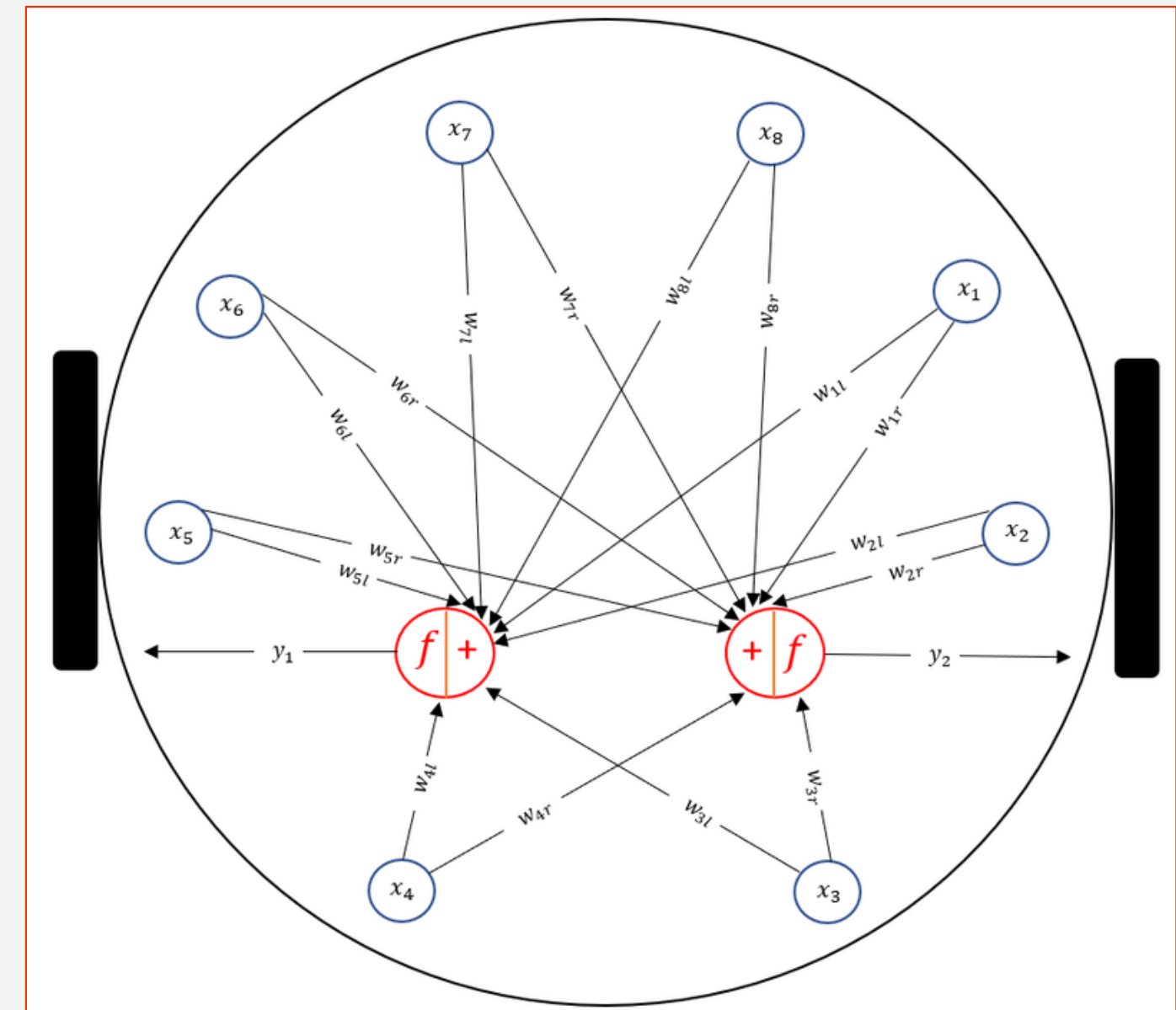


Réseau de perceptron

Réseau de perceptron pour le robot e-puck.

- y_1 : Vitesse de la roue gauche.
- y_2 : Vitesse de la roue droite.
- $x_1 \dots x_8$: données des capteurs (booléennes).

$$\begin{cases} y_1 = b + \sum_{i=1}^8 x_i w_{il} \\ y_2 = b + \sum_{i=1}^8 x_i w_{ir} \end{cases}$$



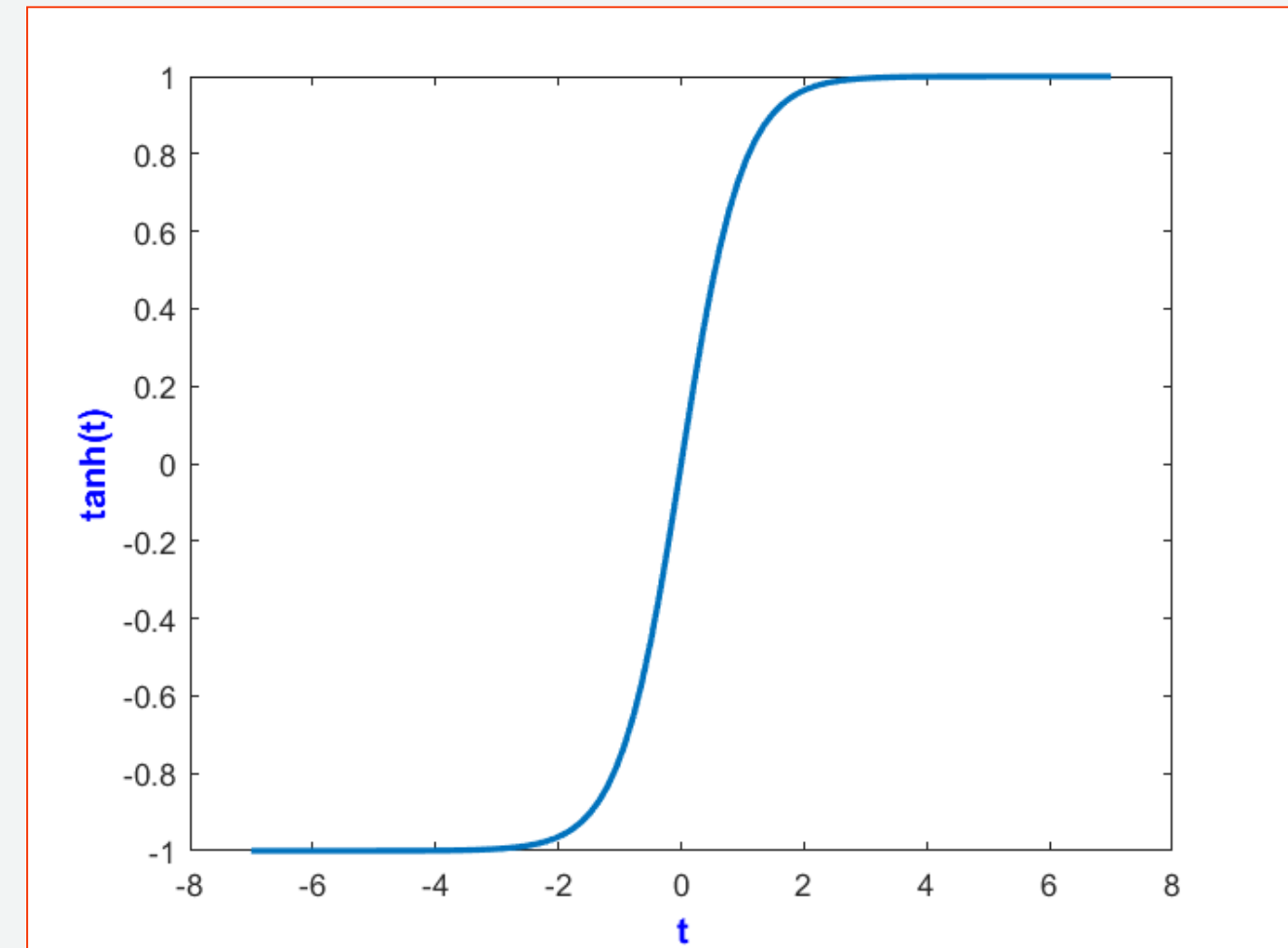
Réseau de perceptron appliqué au robot e-puck.

Réseau de perceptron pour l'epuck

Fonction d'activation

But : Modélisation du traitement des informations reçues.

- Fonction utilisée : La tangente hyperbolique.
- $y \leftarrow \tanh(y)$.
- Saturation de la sortie $[-1,1]$.



Courbe de la tangente hyperbolique.

Réseau de perceptron pour l'epuck

Schéma de commande

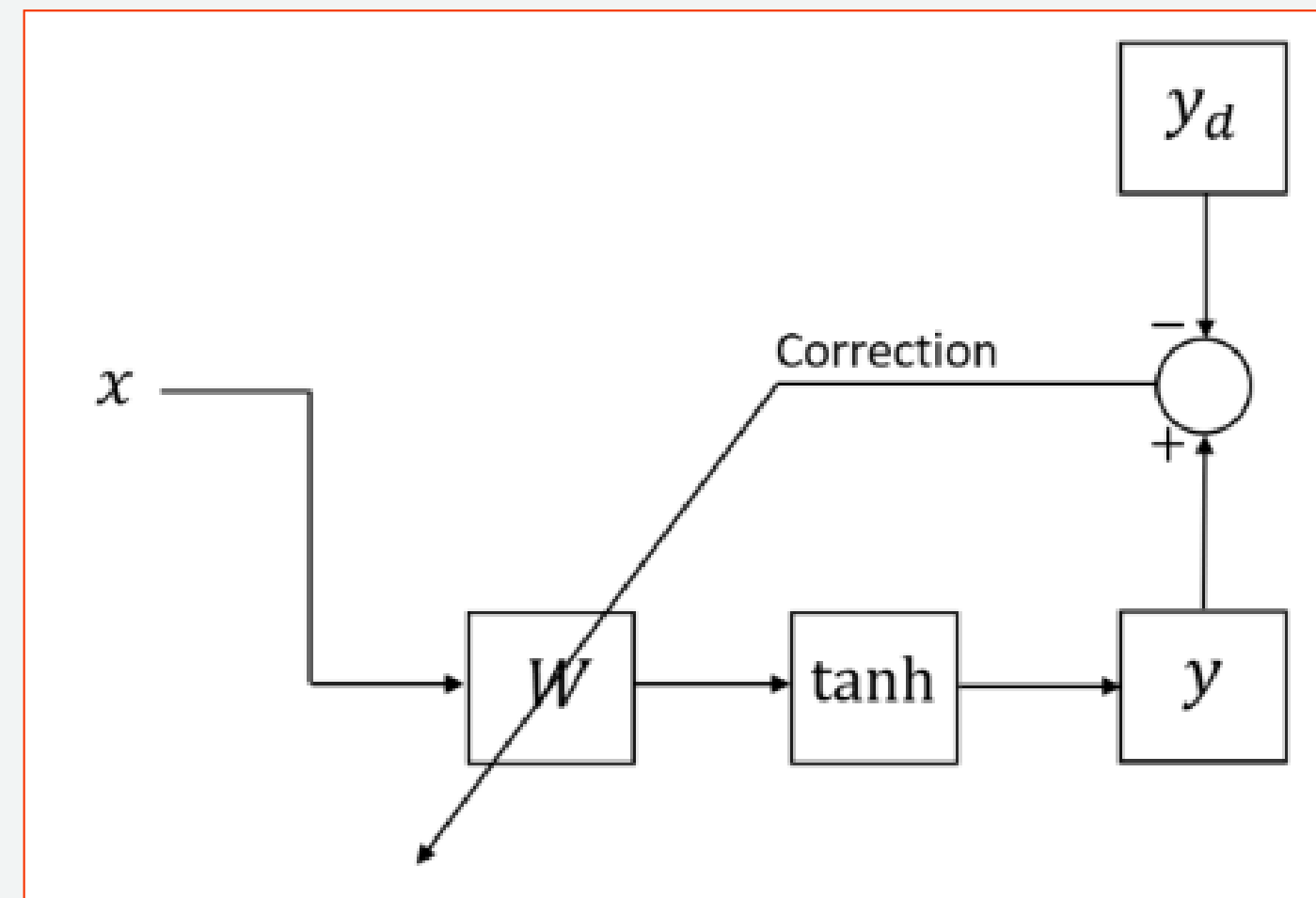


Schéma de commande

Types d'apprentissage

Apprentissage par renforcement

Le robot reçoit une prime ou une punition suivant le choix qu'il fait.

Apprentissage non supervisé

La machine ne dispose pas de modèle. Elle apprend par elle-même.

Apprentissage supervisé

C'est le superviseur qui donne la sortie souhaitée correspondante à une entrée du réseau de neurones



Apprentissage supervisé

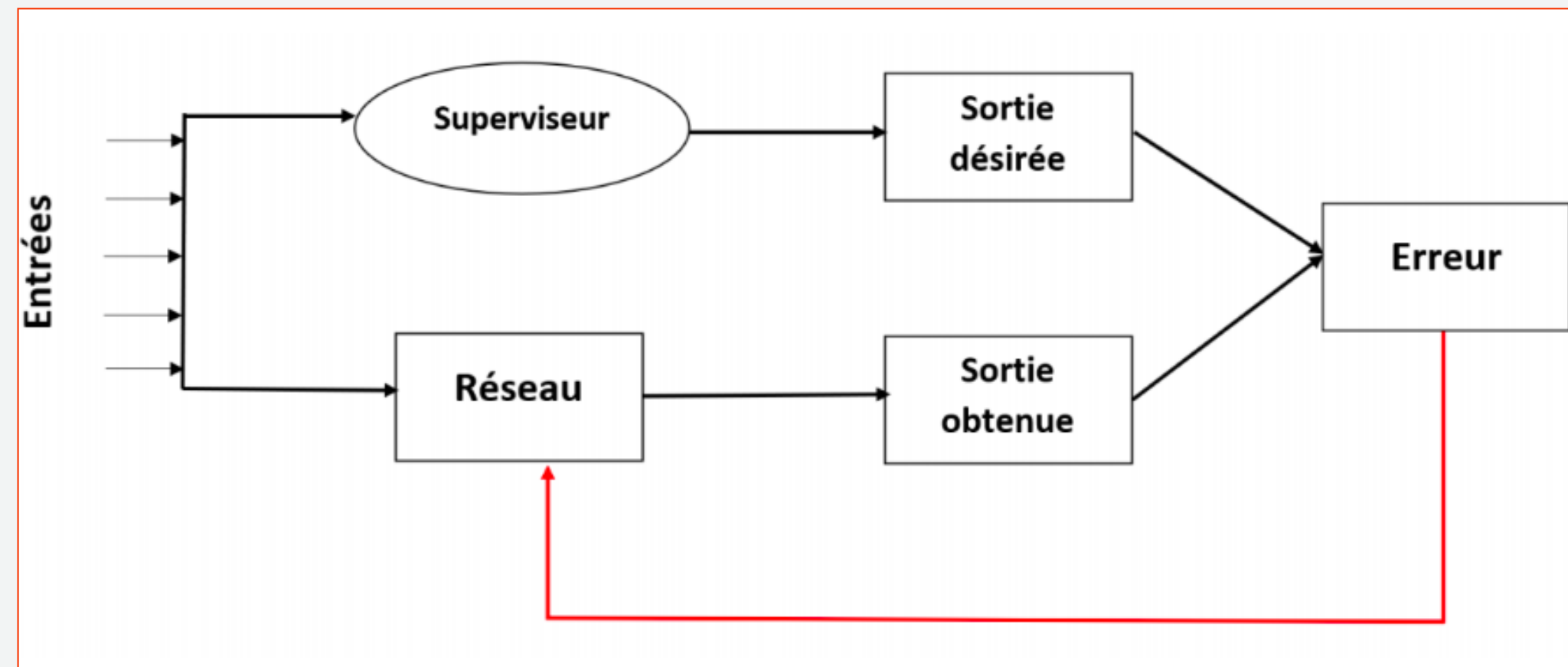


Schéma d'apprentissage supervisé.

Apprentissage supervisé

Détection d'obstacles

Aucun

Capteur 5, 6 et 7

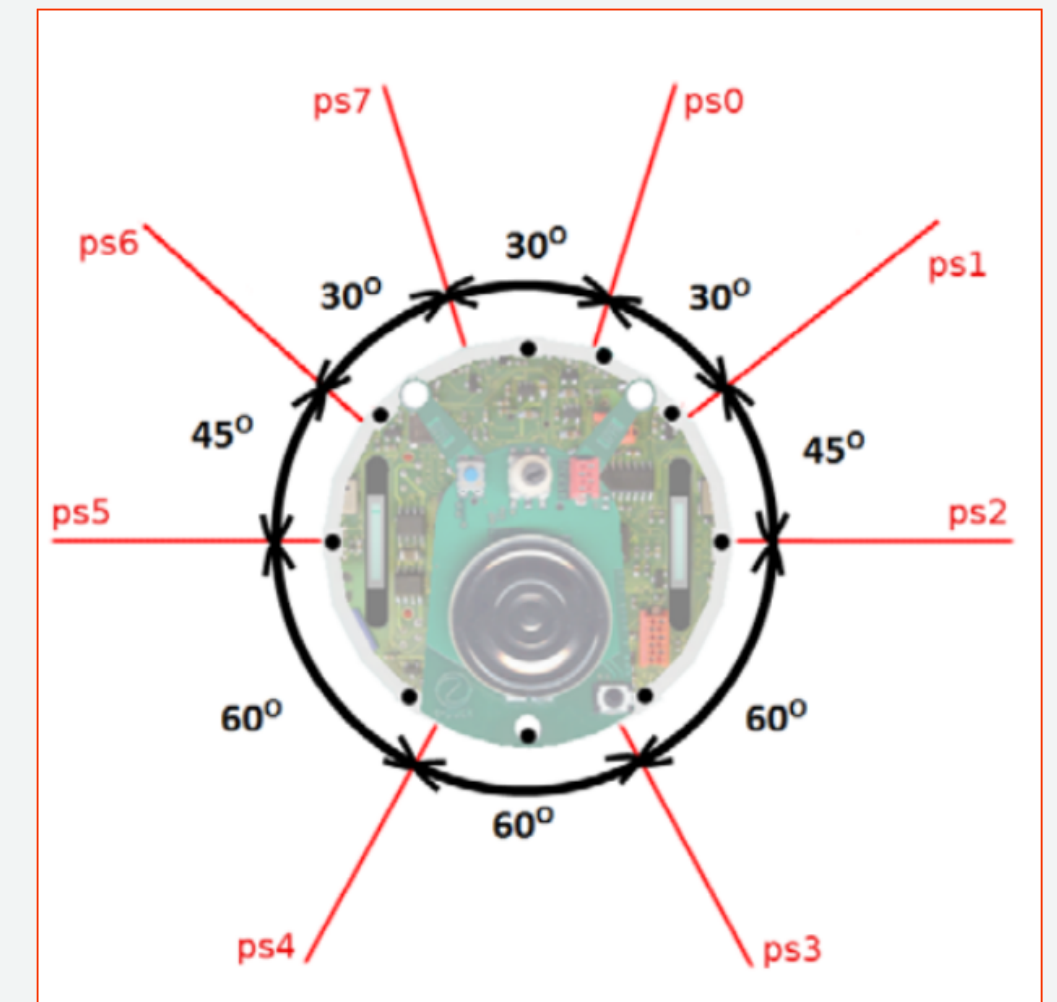
Capteur 0 et 1

Réaction du robot

Le robot va tout droit.

Le robot tourne à droite.

Le robot tourne à gauche.



Capteurs du robot e-puck

Algorithmes utilisés

Algorithme de Hebb.

—
TECHNIQUE D'APPRENTISSAGE
SIMPLE POUR UN ANN.

Algorithme de descente de gradient.

—
ALGORITHME STOCHASTIQUE.

Algorithme de Hebb

- Mise à jour automatique des poids.
- Pas de mise à jour de biais.

$$\begin{cases} w_{jl} \leftarrow w_{jl} + \alpha y_1 x_j \\ w_{jr} \leftarrow w_{jr} + \alpha y_2 x_j \end{cases}$$

L'erreur de sortie

$$\begin{cases} y_1 \leftarrow y_{1d} - y_1 \\ y_2 \leftarrow y_{2d} - y_2 \end{cases}$$

Algorithme avec descente de gradient

Démarche
globale pour
l'apprentissage
de la machine

❁ 1. Descente du gradient.

Ajuster les paramètres pour diminuer l'erreur.

❁ 2. Fonction Backpropagation.

Calcul de la dérivée de l'erreur.

❁ 3. Biais.

Mise à jour automatique du biais.

Algorithme avec descente de gradient

La descente du gradient

- Méthode d'optimisation non linéaire intuitive

La formule générale

$$W \leftarrow W - l_r \frac{\partial E}{\partial W}$$

L'erreur moyenne quadratique

$$E = \sum \frac{1}{2} (y_d - y)^2$$

Algorithme avec descente de gradient

La fonction backpropagation

- La règle de dérivation des fonctions composées.

$$\frac{\partial E}{\partial W} = \frac{\partial E}{\partial f} \frac{\partial f}{\partial y} \frac{\partial y}{\partial W}$$

$$\begin{cases} \frac{\partial E}{\partial f} = y - y_d \\ \frac{\partial f}{\partial y} = 1 - \tanh y \\ \frac{\partial y}{\partial W} = x \end{cases}$$

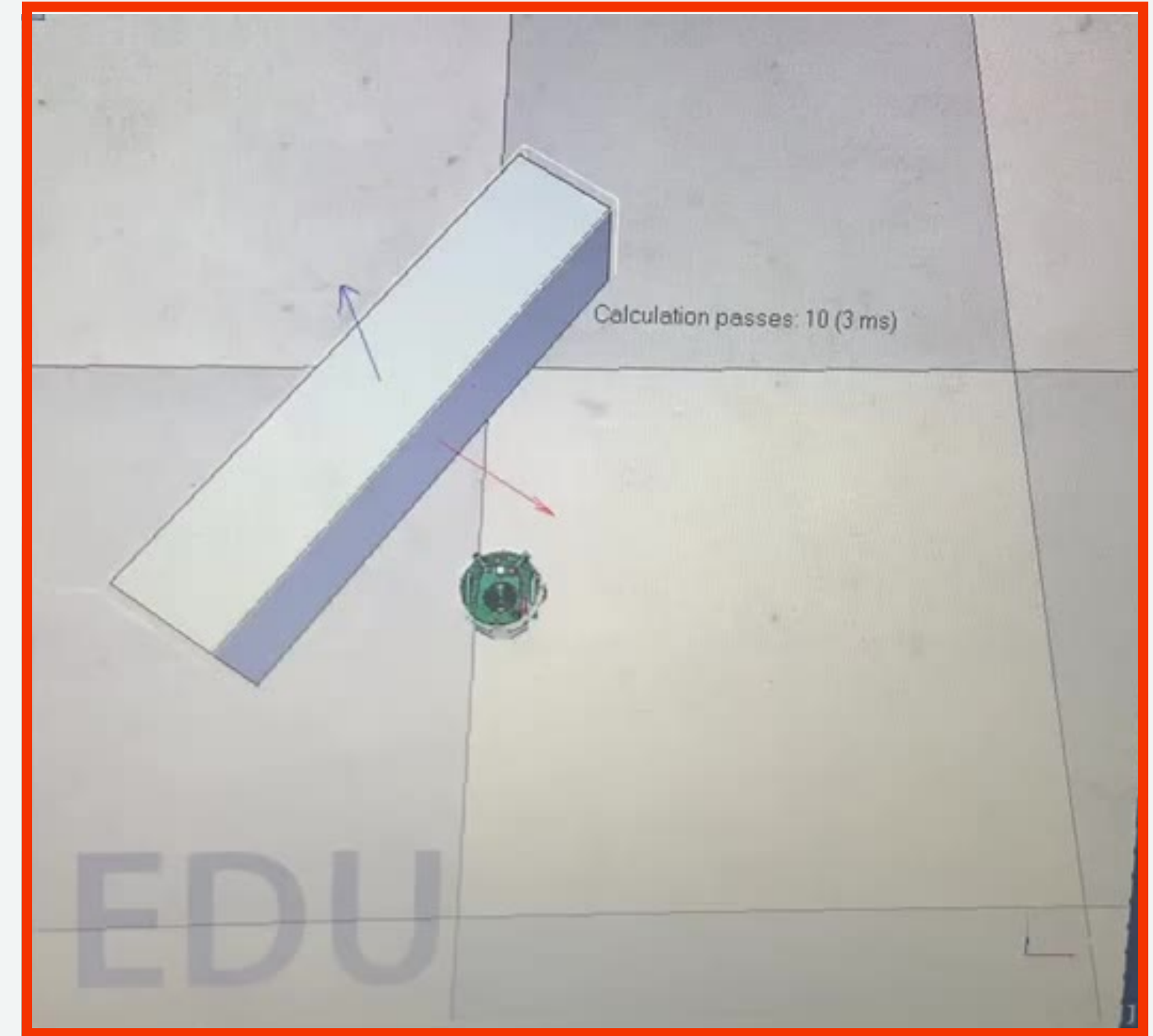
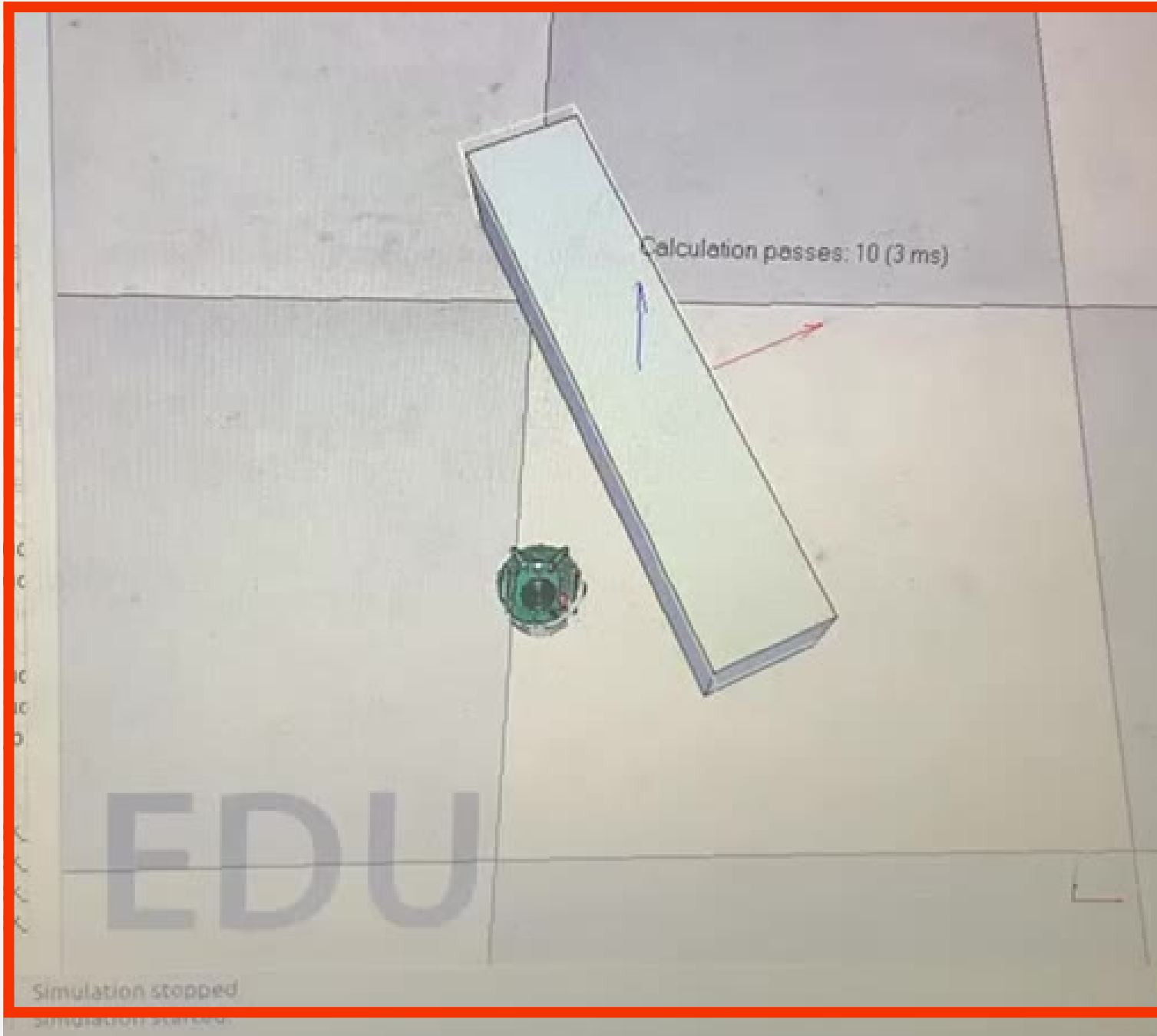
Mise à jour du biais

- Méthode de descente
- Biais indépendant de l'entrée x .

$$b \leftarrow b - l_r \frac{\partial E}{\partial y}$$

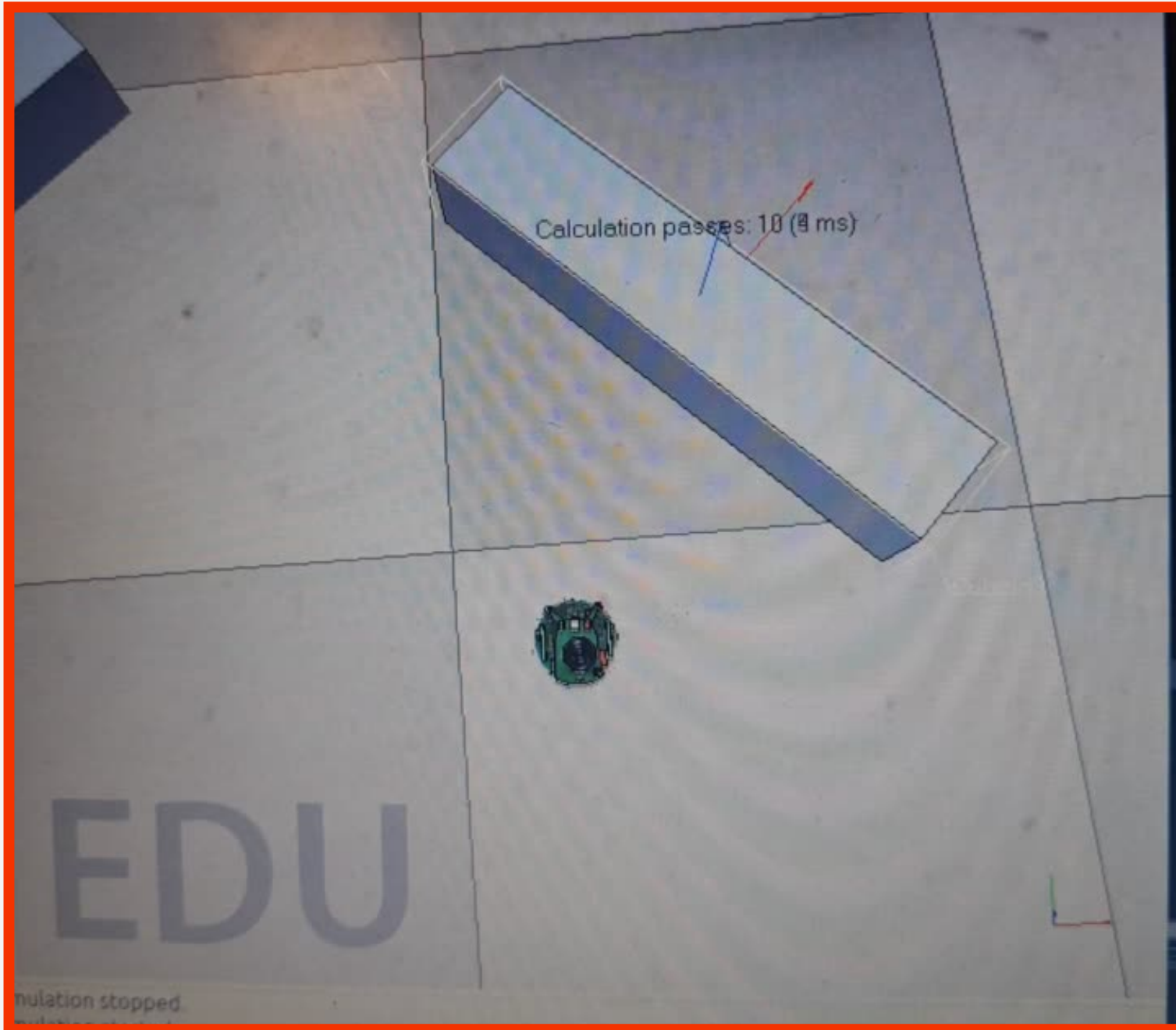
Simulation

Algorithme de Hebb



Simulation

Algorithme de descente du gradient



Conclusion

Axe principal

- L'évitement d'obstacles d'un robot mobile dans un environnement incertain.
- Biais piloté => Système plus robuste.

Perspectives

- Utilisation de librairies pour des réseaux plus complexes (Pytorch/Tensorflow...).
- L'utilisation d'un réseau de neurones convolutifs (CNN).

