

TD1 : INTRODUCTION À LA MODÉLISATION DE RÉSEAUX DE NEURONES À IMPULSION

INTRODUCTION

Ce TD vous propose une introduction à la modélisation de neurones à impulsion en robotique neuronale [1]. Deux modèles sont étudiés : le modèle explicatif de Hodgkin-Huxley et le modèle phénoménologique Intègre-et-tire avec fuite (*leaky integrate-and-fire* ou LIF). Les objectifs de cette formation sont :

- Comprendre la dynamique de production du potentiel d'action.
- Comprendre la modélisation cinématique de la locomotion d'un robot humanoïde disposant d'une base mobile (le robot PR2 et pris comme cas d'étude).
- Comprendre le mécanisme d'excitation des neurones via une stimulation imposée ou des synapses.
- Programmer une tâche de navigation réactive en simulation sur Webots avec le robot **E-Puck** (développé par GCtronic et EPFL), contrôlée par un réseau de neurones à impulsion LIF.

Le travail est organisé en deux parties. Dans la Partie I, deux cahiers Jupyter Notebook vous sont proposés, décrivant quelques scénarios de simulation. La partie II vous propose d'étudier le simulateur Webots (python).

PARTIE I : ETUDE DES MODÈLES

Fichier « TD1-A_LIF.ipynb »

Ce cahier de travail vous propose une introduction à la modélisation de neurones à impulsion selon le modèle intègre-et-tire avec fuites (Leaky integrate-and-fire ou LIF).

Fichier « TD1-B_MHH.ipynb »

Ce cahier de travail vous propose d'étudier le modèle de Hodkin-Huxley via la simulation de la dynamique du potentiel d'action.

PARTIE II : SIMULATION DE LA NAVIGATION REACTIVE (WEBOTS)

Webots est un simulateur open-source de robotique utilisé dans l'industrie, la recherche et l'enseignement. Pour l'élaboration de ce TD veuillez installer la version *28 Jun 2023 - R2023b*.

Il est conseillé de suivre les quatre premiers [tutoriaux](#) afin de comprendre le fonctionnement de Webots. Vous allez programmer un contrôleur qui implémente un modèle de réseau de neurones à impulsion LIF pour la régulation de la navigation réactive (évitement d'obstacle). Un bon point de départ serait le [tutoriel 4](#), qui vous propose une introduction à la classe **Controller**, prenant comme cas d'étude le robot mobile **E-puck**.

La simulation que vous allez réaliser s'inspire du fonctionnement des véhicules de Brainternberg (voir Fig. 1). Dans son ouvrage de référence [2], Valentino Brainternberg modélise le monde animal de manière minimalistique et constructive, depuis les comportements réactifs les plus simples (comme la phototaxie) jusqu'aux véhicules les plus élémentaires.

Architecture proposée

Plusieurs architectures sont possibles pour la mise en place d'un contrôleur qui régule un comportement d'évitement d'obstacles. Une proposition vous est faite en Fig. 2, dans laquelle six neurones sensoriels encodent les données des capteurs de position PS du robot, tandis que trois neurones moteurs régulent les trois comportements : tourner à droite, tourner à gauche, ou aller vers l'avant.

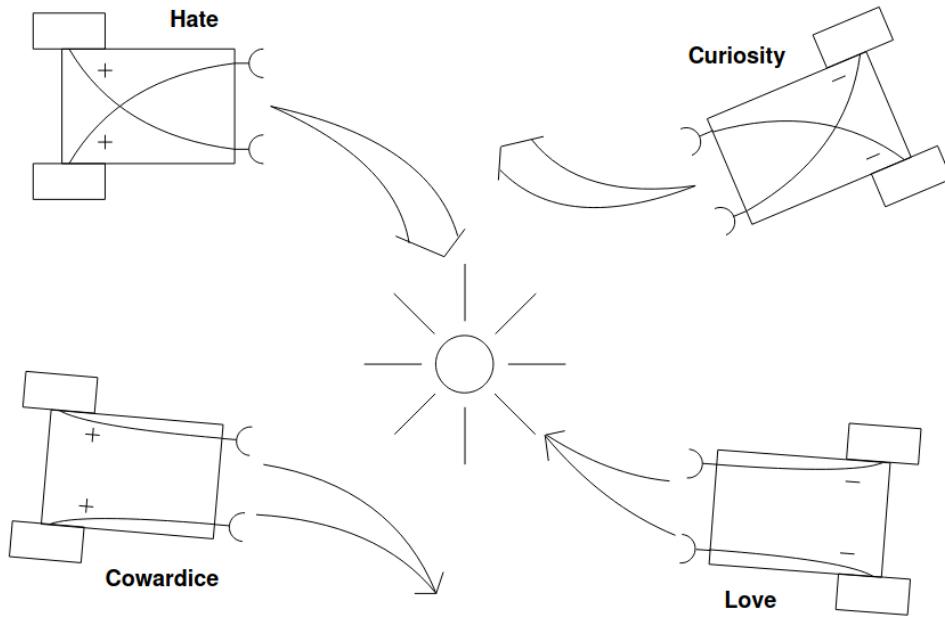


Figure 1: Quatre véhicules de Braatenberg de base illustrent leur comportement face à une source lumineuse grâce à des schémas de connexion très simples [3]. Les arcs à l'avant des véhicules sont des capteurs. Ces capteurs sont reliés aux unités de commande par des connexions croisées ou non croisées, excitatrices (+) ou inhibitrices (-). Les émotions indiquées correspondent aux termes utilisés par Braatenberg pour décrire la réaction des véhicules à la lumière. Les flèches représentent le mouvement des véhicules, leur largeur indiquant la vitesse.

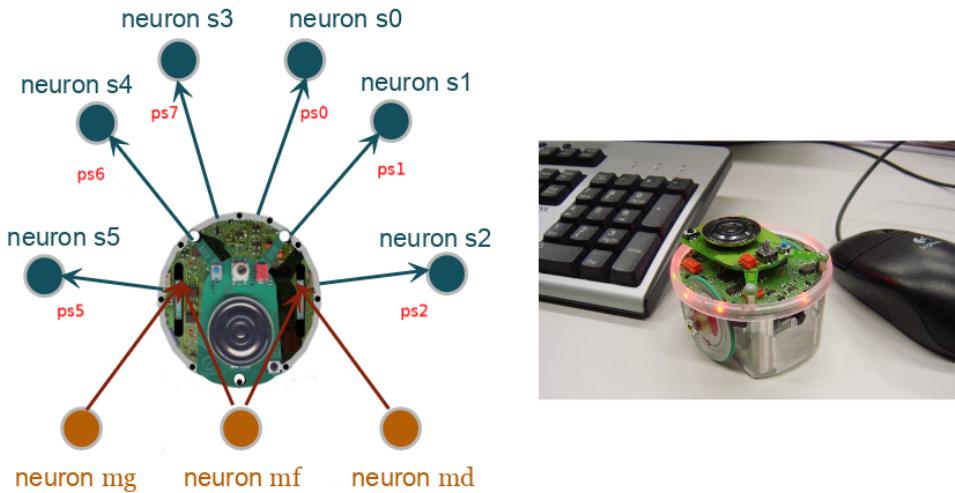


Figure 2: Exemple d'architecture de réseau de neurones à impulsion LIP pour la régulation d'un comportement d'évitement d'obstacle. Le robot E-puck dispose de plusieurs capteurs infrarouges qui permettent l'estimation de la distance à des obstacles situés à quelques centimètres autour de lui. Les relevés de chaque capteur sont mis à l'échelle pour stimuler un neurone sensoriel LIF respectif (en vert). L'activité des neurones sensoriels stimulent les motoneurones (en orange) pour contrôler le déplacement du robot vers l'avant (neuron mf) et le virage à gauche (neuron mg) ou à droite (neuron md), via des synapses excitatrices ou inhibitrices, compte tenu de la disposition des obstacles dans l'espace proximale du robot. Il est également possible d'établir des connexions synaptiques entre les neurones sensoriels afin de favoriser la compétition ou la coopération en réponse à certaines situations rencontrées dans l'environnement.

Modelisation des synapses

Vous allez programmer le modèle de synapse exponentiel où le courant I_{syn} (en ampères) correspond au produit des conductances (en siemens) et potentiels synaptiques, tel que

$$I_{\text{syn}}(t) = \sum_i g_{\text{syn}i}(t) (E_{\text{syn}i} - V(t)) \quad (1)$$

$$\dot{g}_{\text{syn}i}(t) = \frac{-g_{\text{syn}i}(t)}{\tau_{\text{syn}i}} \quad (2)$$

où $g_{\text{syn}i}$, $E_{\text{syn}i}$, et $\tau_{\text{syn}i}$ désignent respectivement la conductance, le potentiel de repos et la constante de temps de la synapse. Souvent, $E_{\text{syn}i} = 0$ mV pour une synapse excitatrice et $E_{\text{syn}i} = -70$ mV pour une synapse inhibitrice. Lors de la production d'une impulsion dans le neurone pré-synaptique i , $g_{\text{syn}i}$ est modifié dans le neurone post-synaptique, tel que

$$g_{\text{syn}i}(t) = g_{\text{syn}i}(t-1) + w_{\text{syn}i} \quad (3)$$

dont $w_{\text{syn}i}$ correspond au poids synaptique.

Voici le modèle LIF en incluant la stimulation par courant synaptique

$$C_m \dot{V}(t) = I_s(t) + I_{\text{syn}}(t) - I_{\text{fuite}}(t) \quad (4)$$

Compte tenu de l'architecture proposée en Fig. 1, à l'aide des modèles de synapse (excitatrice ou inhibitrice) que vous devez paramétrier pour contrôler le mouvement d'évitement d'obstacles, une consigne en vitesse de rotation \dot{r} des roues peut être définie tel que

$$\dot{r}^{[x]}(t) = (1 - \lambda) \dot{r}^{[x]}(t-1) + \lambda \left(\sum_{w=0}^{w=W-1} [\delta(t-w-T^{[x]}) + \delta(t-w-T^f)] \right) \quad (5)$$

où $x = \{g, d\}$ désigne les roues gauche (g) et droite (d), $\delta()$ est la fonction de Dirac, T^g , T^d et T^f dénotent les temps d'impulsion respectivement des motoneurones mf, mg ou md, λ est un facteur d'interpolation linéaire, et W représente la taille de la fenêtre glissante selon hypothèse du taux d'activation (*firing-rate*).

LE TRAVAIL À RENDRE

Pour la partie I, répondez aux questions et exercices proposés directement sur les cahiers de travail. Vous devez donc soumettre les fichiers « TD1-A_LIF.ipynb » et « TD1-B_MHH.ipynb » contenant vos réponses.

Pour la partie II, préparez un fichier nommé « contrôleurLIF.py » contenant le programme en langage python pour la régulation du comportement d'évitement d'obstacles. Préparer également un compte rendu en format PDF pour justifier vos choix effectués pour la mise en place du modèle et la présentation de vos résultats à l'aide de quelques figures. Vous avez donc à rendre 4 fichiers sur Arche.

CONDITIONNEMENT

Travail individuel à rendre au plus tard le **vendredi 9 janvier 2026 avant minuit**.

Code d'honneur

Il est essentiel d'agir de manière éthique. L'utilisation d'outils d'IA générative n'est pas autorisée pour l'élaboration de ce travail. Ce document est protégé par le droit d'auteur ; veuillez ne pas le soumettre à ChatGPT ni à des plateformes similaires.

References

- [1] Zhenshan Bing, Claus Meschede, Florian Röhrbein, Kai Huang, and Alois C Knoll. A survey of robotics control based on learning-inspired spiking neural networks. *Frontiers in neurorobotics*, 12:35, 2018.
- [2] Valentino Braitenberg. *Vehicles: Experiments in synthetic psychology*. MIT press, 1986.
- [3] Gordon Wyeth. Training a vision guided mobile robot. *Autonomous Robots*, 5(3):381–394, 1998.