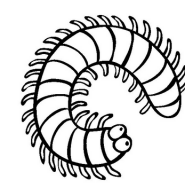


Locomotion : simulation du déplacement d'un mille-pattes

Nous attestons que nous sommes les auteurs du présent travail et que tout ce qui a été empruntée est attribué à sa source et proprement référencé
Auteurs: Soline Bernard, Antoine Marion, Victor Yon Groupe : P2 Temps de travail: 45h/personne Date: 15/10/2020



 github.com/victor-yon/gei723

Objectif

Implémenter une simulation de marche d'un hexapode sous Python à l'aide de l'outil Brian2.
Simuler le déplacement d'un hexapode en ligne droite et en cercle, puis l'étendre à une solution de plus de 10 pattes.
Simuler un déplacement sur la droite s'il est touché à gauche et inversement.
Simuler une accélération s'il est touché par l'arrière.
Simuler un recul s'il heurte un objet de face.

Modélisation

Chaque patte est assimilée à un simple segment de droite pouvant se déplacer de l'avant en arrière, avec ou sans contact avec le sol. [1]
Notre système se décompose en quatre parties :

- Un CPG (central pattern generator) permettant d'instaurer un démarche périodique, qui peut varier en vitesse s'il est stimulé.
- Une partie contrôlant la marche arrière/marche avant
- Une partie permettant de tourner à gauche ou à droite
- La partie mouvement de la patte.

Des capteurs sensoriels, symbolisés par des triangles sur le schéma, permettent de modifier les paramètres du système en fonction de l'environnement de l'hexapode.
Pour assurer la stabilité de l'hexapode, les pattes paires gauches et impaire droites ont les mêmes mouvements, et les pattes impaires gauches et paires droites également. Il s'agit de la marche en tripode[2]
Le schéma ne représente qu'une seule patte. Chaque patte est reliée à l'oscillateur et reçoit des informations décalées d'une demie période en fonction de sa parité.

Le CPG

Le CPG permet d'imposer le rythme à notre réseau de neurone. Il s'agit d'un oscillateur qui, une fois déclenché par le neurone déclencheur, oscille entre 2 neurones à chaque 5 décharges. Cette modélisation se rapproche d'un système nerveux centralisé, et permet de synchroniser toutes les pattes à partir d'une même génération de décharges.

La patte

La patte en elle-même est composée de 4 neurones moteurs, chacun donnant l'ordre d'une des 4 instructions suivantes : avancer la patte, la reculer, la lever, la baisser (ou appuyer sur le sol si la patte touche déjà la sol).
Le mouvement se décompose en 2 étapes.
1) La patte se lève en avançant : les neurones **up** et **forward** sont alors **activés** et les neurones **down** et **back** sont **inhibés**.
2) La patte se baisse en reculant : les neurones **down** et **back** sont alors **activés** et les neurones **up** et **forward** sont **inhibés**.
Les 2 parties sont liées à l'oscillation du CPG. Ainsi, quand le 1er neurone du CPG déclenche, la patte se lève en avançant, puis quand le 2e neurone du CPG déclenche, la patte se baisse en reculant.

Neurone Back et Forward

Comme le CPG oscille toute les 5 décharges, nous considérons que la patte avance ou recule de 20% de son amplitude à chaque fois qu'elle reçoit une décharge.

Neurone Up et Down

Pour la lever de la patte, la première décharge est assimilée à 100% de la montée, puis les suivantes à un maintient en position levé.
Pour le neurone **down**, la première décharge est assimilée à 100% de descente de la patte, puis les suivantes sont considérées comme étant une poussée au sol. Couplé au neurone back ou forward, cela permet d'avancer ou de reculer.

Le mouvement des pattes est directement calqué sur les décharges de l'oscillateur. Chaque neurone de l'oscillateur est relié à la moitié du mouvement des pattes. Les liaisons s'inversent pour les pattes paires et impaires. Ainsi, si la vitesse de l'oscillateur augmente, la vitesse du mouvement des pattes augmente directement. La représentation globale du fonctionnement d'une des pattes illustre ce mécanisme.

Remerciements à Brian2, GitHub et StackOverflow

Type de neurone

On choisit d'utiliser des neurones de type intégration et décharge sans courant de fuite pour le CPG car ils sont plus simples à manipuler et moins lourds en équation. De plus, c'est une approche communément employée ayant fait ses preuves dans la littérature [1], [3]. Ils permettent de relier plus facilement, via des équations, les temps de décharges, et les délais synaptiques pour obtenir un oscillateur. De plus, il n'est pas nécessaire d'avoir un modèle qui se rapproche de la réalité. Nous avons également utilisé des neurones de type intégration et décharge avec courant de fuite pour la gestion des système de rotations et de sens de la marche. Voici les équations permettant de modéliser les neurones à décharges sans fuite (à gauche), puis avec fuite (à droite).

$$\frac{dv}{dt} = I/\tau \qquad \frac{dv}{dt} = \frac{I - v}{\tau}$$

Marche arrière/Marche avant

Le sens de la marche est géré par 1 capteur externe (frontal). Lorsque ce capteur n'est pas stimulé, le sens naturel de la marche est la marche avant. Lorsqu'il est stimulé, il envoie une décharge qui permet d'inverser les seuils de déclenchement des neurones **Up** and **Down** de tous les neurones. Ainsi, **Up** sera déclenché en même temps que **Back**, et **Down** en même temps que **Forward**. Le système ira alors en marche arrière.

Tourner

Pour tourner, nous utilisons 2 capteurs sensoriels, un à droite et un à gauche. Lorque l'un de deux est stimulé, le neurone inhibe grâce à un système de seuil les impulsions envoyées par l'oscillateur aux neurones **Forward** et **Back** des pattes du coté opposé. Ainsi, ces pattes auront des mouvements moins amples d'un coté que de l'autre, ce qui aura pour effet de faire tourner le système.
Si les 2 capteurs sont stimulés en même temps, cela aura pour effet de faire ralentir la fréquence de marche globale du système. Jusqu'à l'arrêt complet une fois la valeur maximale atteinte.

Variation de la vitesse de marche

La vitesse de marche varie à partir d'un neurone récepteur, V. Lorsque ce neurone est stimulé, l'intensité envoyée au CPG augmente, ce qui a pour effet de diminuer la constante de temps des équations utilisées. Ainsi, chaque neurone du CPG déclenche plus rapidement et la fréquence globale du système et donc de la marche est augmentée.

Programmation des capteurs

Les entrées des capteurs sont initialisés avant la simulation. Ce sont des constantes que l'on déclare au début du code. Ceci fait en sorte que l'on observe un comportement constant pour une même simulation.

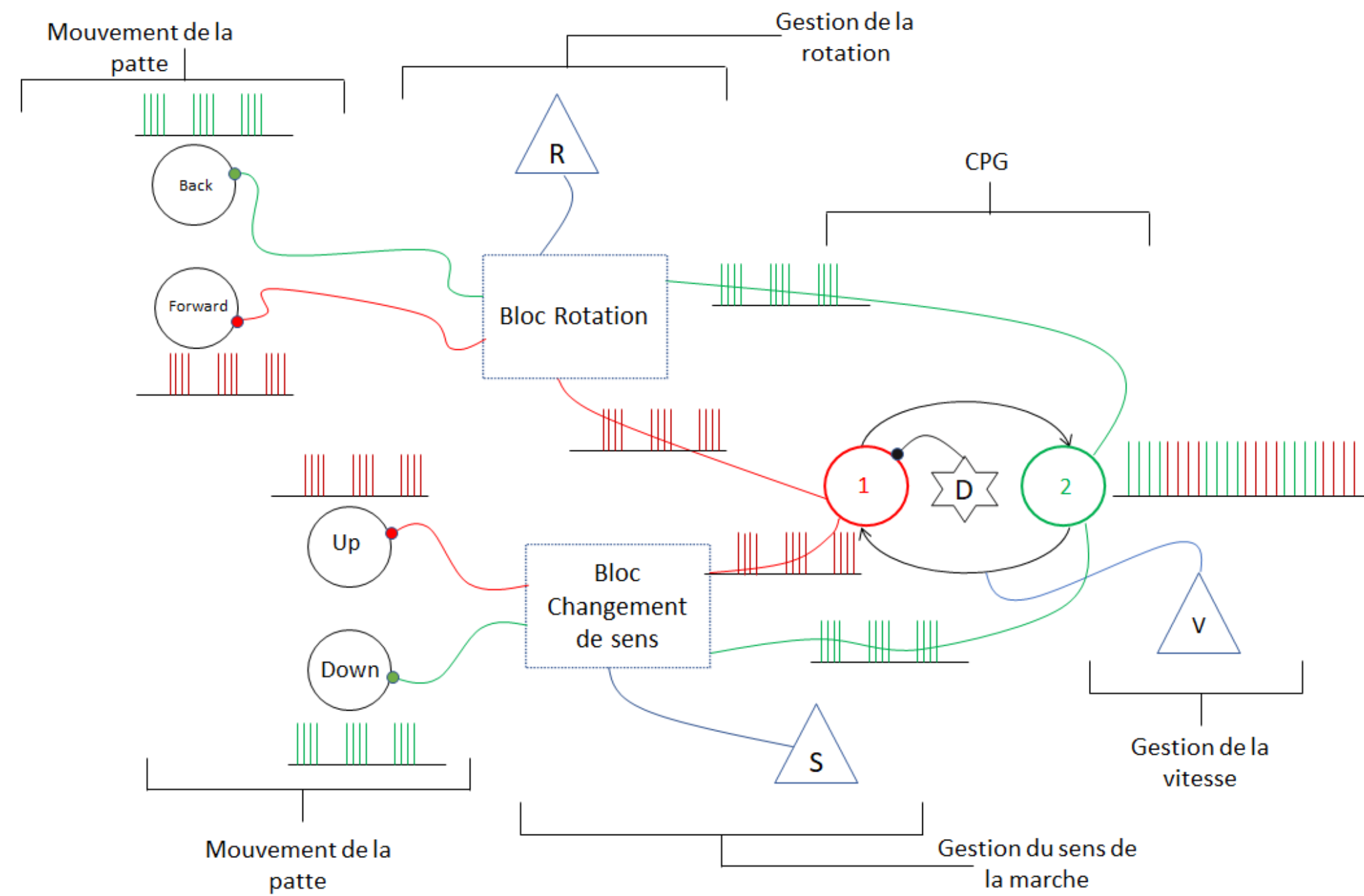
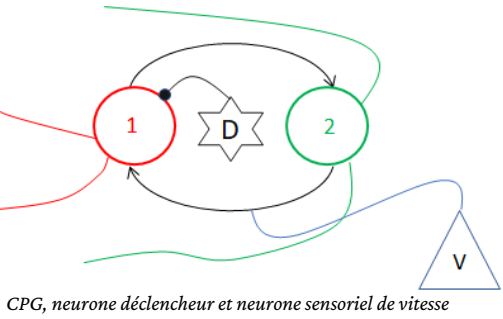
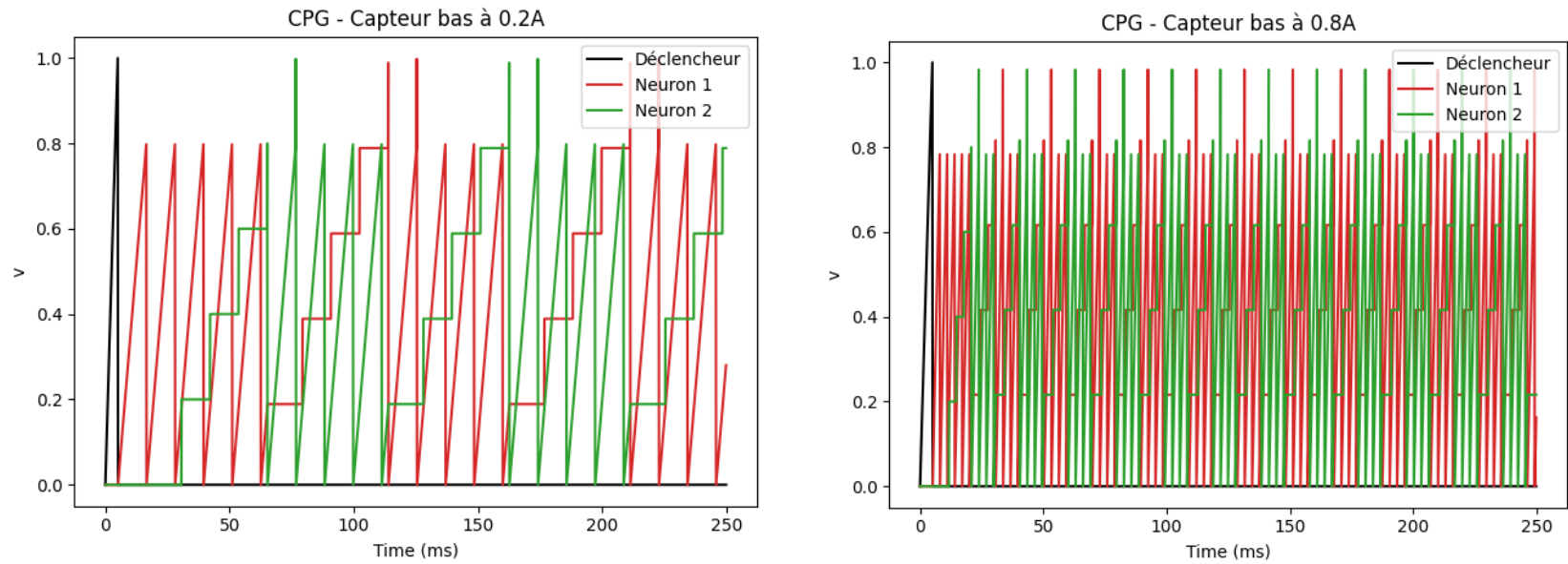


Schéma globale du système neuronale lié au déplacement d'un mille-pattes avec représentation des décharges provenant du CPG. Chaque bloc est détaillé à la page suivante.

Central Pattern Generator (CPG)

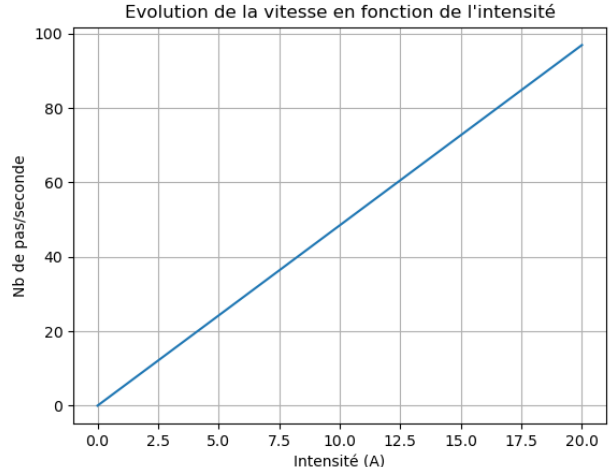
Le CPG est déclenché par le neurone déclencheur **D**. Ce neurone charge, et déclenche après 5ms, le temps que l'information arrive jusqu'au CPG. Il est lié au neurone **1** de l'oscillateur. Après 5 décharges générées selon le modèle intégration et décharge sans fuite, c'est au tour du **second** neurone de déclencher 5 fois grâce à une liaison synaptique entre les 2 neurones. La vitesse est générée par un second neurone moteur, qui influe sur l'intensité arrivant dans le CPG. Plus l'intensité sera grande, plus la décharge arrivera rapidement car le temps de décharge t_d sera diminuée. On peut ainsi augmenter ou diminuer la vitesse du système.



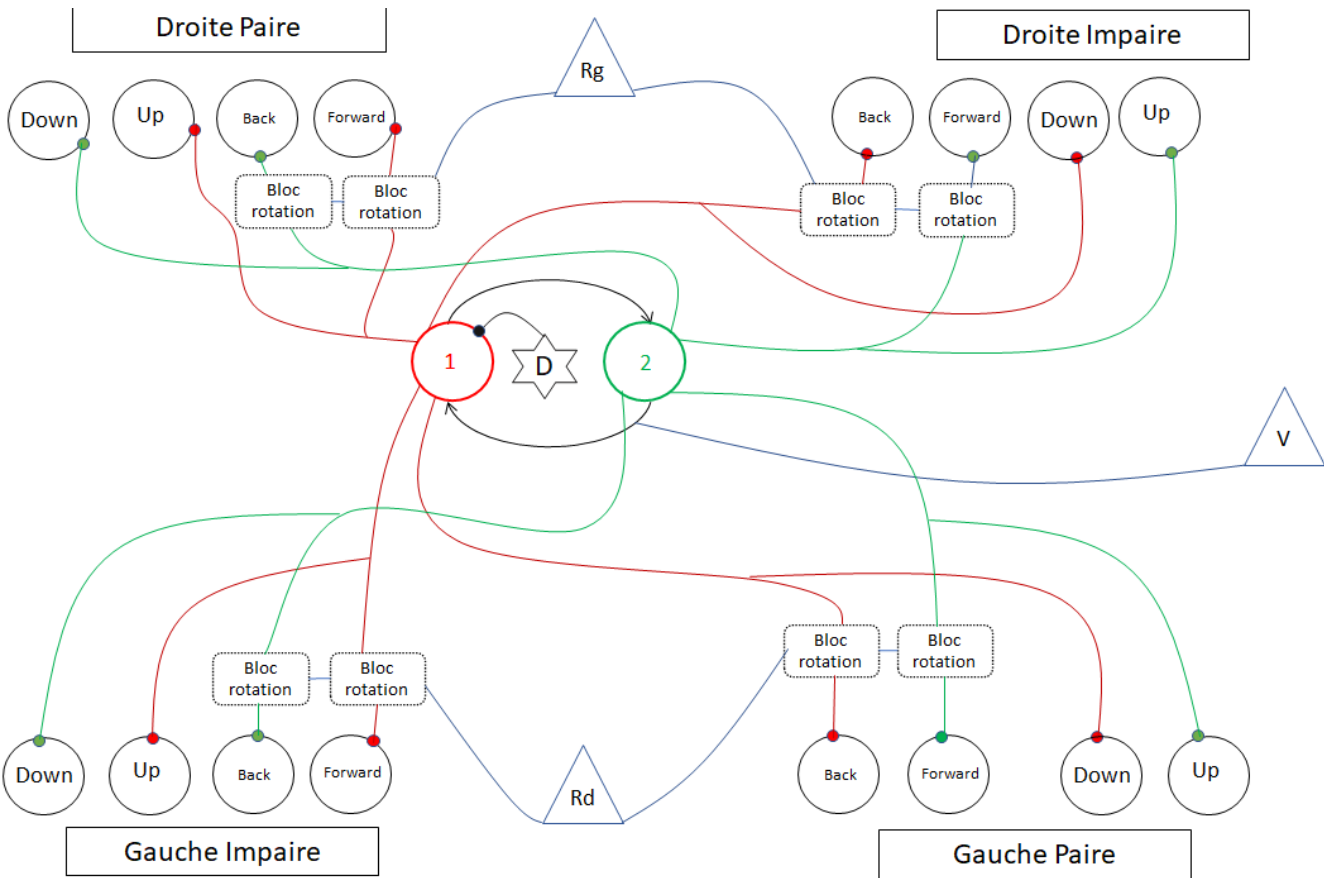
CPG, neurone déclencheur et neurone sensoriel de vitesse

$$t_d = \frac{\text{seuil} * \tau}{I(\text{vitesse})}$$

Les synapses sont définies pour qu'à chaque décharge, le second neurone voit son poids augmenter d'un quart de sa valeur seuil après un léger délai. Ce délai permettra de ne pas donner d'information contradictoire pour le mouvement de la patte.



Liaison selon la parité et le côté des pattes



Réseau de neurones de l'hexapode depuis le CPG vers les neurones moteurs selon la parité des pattes

Par souci de simplification, nous n'avons pas représenté les neurones initiant les changements du sens de la marche et nous avons représenté par des blocs les mécanismes pour tourner.

Références:

[1] M. OTIS, “Developpement d’un système de propulsion pour un biomicrorobot hexapode avec un ionomère erfluorosulfonique,” Master’s thesis, Universite de Sherbrooke, 2004. II-A

[2] H. Cruse, T. Kindermann, M. Schumm, J. Dean, and J. Schmitz, “Walknet’ A biologically inspired network to control six-legged ”walking,” Neural Networks, vol. 11, no. 7, 8, pp. 1435–1447, 1998. [Online]. Available : <http://www.sciencedirect.com/science/article/pii/S0893608098000677> II-A

[3] R. D. Beer, H. J. Chiel, R. D. Quinn, K. S. Espenschied, and P. Larsson, “A distributed neural network architecture for hexapod robot locomotion,” Neural Computation, vol. 4, pp. 356–365, 1992. [Online]. Available : <http://dx.doi.org/10.1162/neco.1992.4.3.356> II-A

[4] N. Porcino, “Hexapod gait control by a neural network,” in 1990 IJCNN International Joint Conference on Neural Networks, June 1990, pp. 189–194 vol. 1 II-A

Tourner

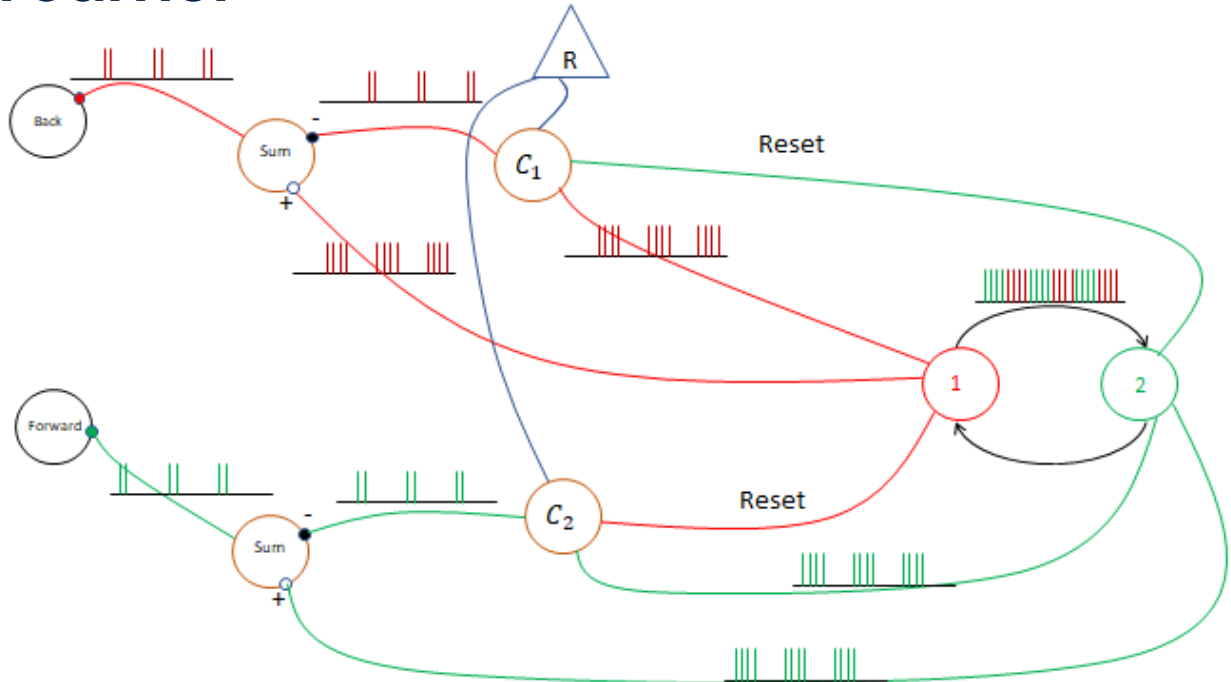
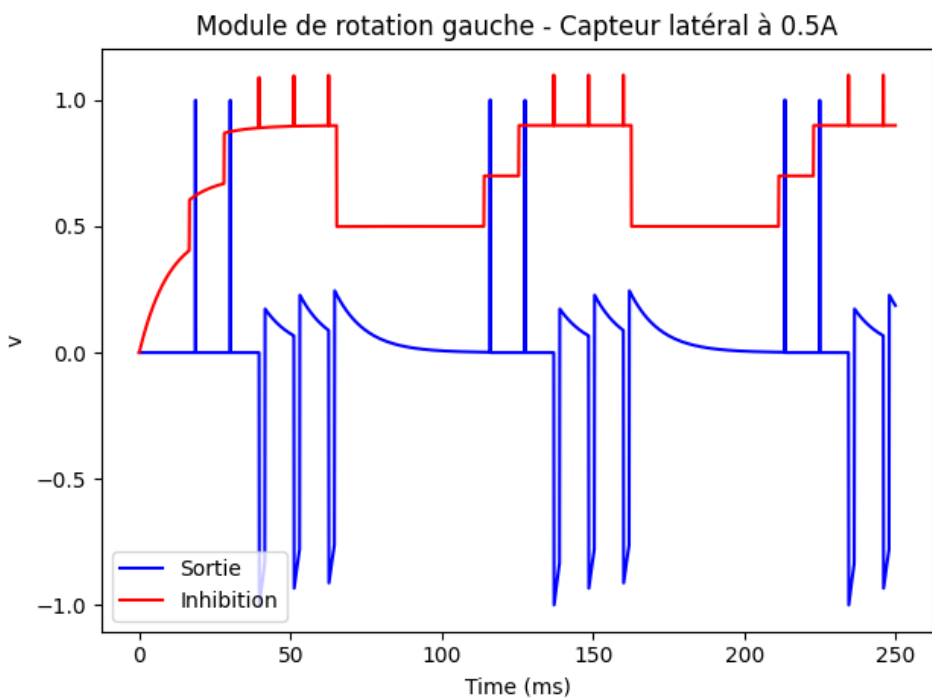
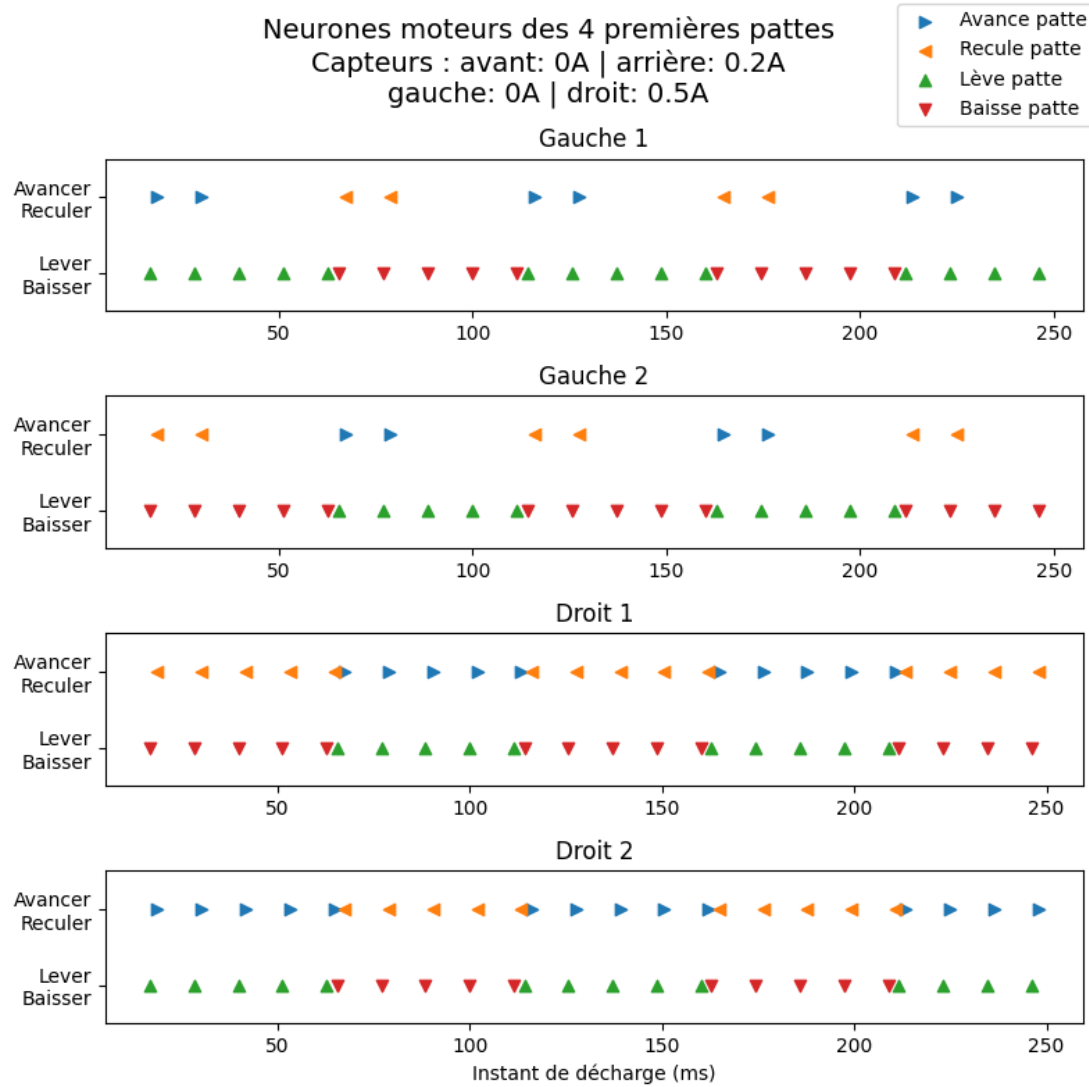


Schéma détaillant la gestion de la rotation

Lorsque le neurone sensoriel **R** est stimulé, il génère un courant d'intensité proportionnelle à la stimulation. Cela à pour effet de générer un certain nombre de décharges fournies par l'oscillateur par l'intermédiaire des neurones **C1** et **C2**. Ces décharges vont être reliées à un neurone **Sum** par l'intermédiaire d'une synapse inhibitrice, alors que les décharges provenant directement de l'oscillateur seront reliées au neurone **Sum** par une synapse excitatrice. Ainsi, seulement une partie des décharges est transmise aux neurones **Back** et **Forward**. La patte n'avancera et ne reculera alors que d'une certaine amplitude en fonction du nombre de décharges reçues (1 décharge = 20% du mouvement).

Le graphique ci-dessous illustre les séquences de décharges des neurones moteurs lorsque le capteur sensoriel à **droite** de l'hexapode est excité. Les décharges de ce capteur amène une diminution du nombre de décharges vers les neurones motrices **Forward** et **Backward** de **gauche** tout en gardant inchangé les séquences pour la patte droite, ce qui tend à faire tourner l'hexapode vers la gauche



Sens de la marche

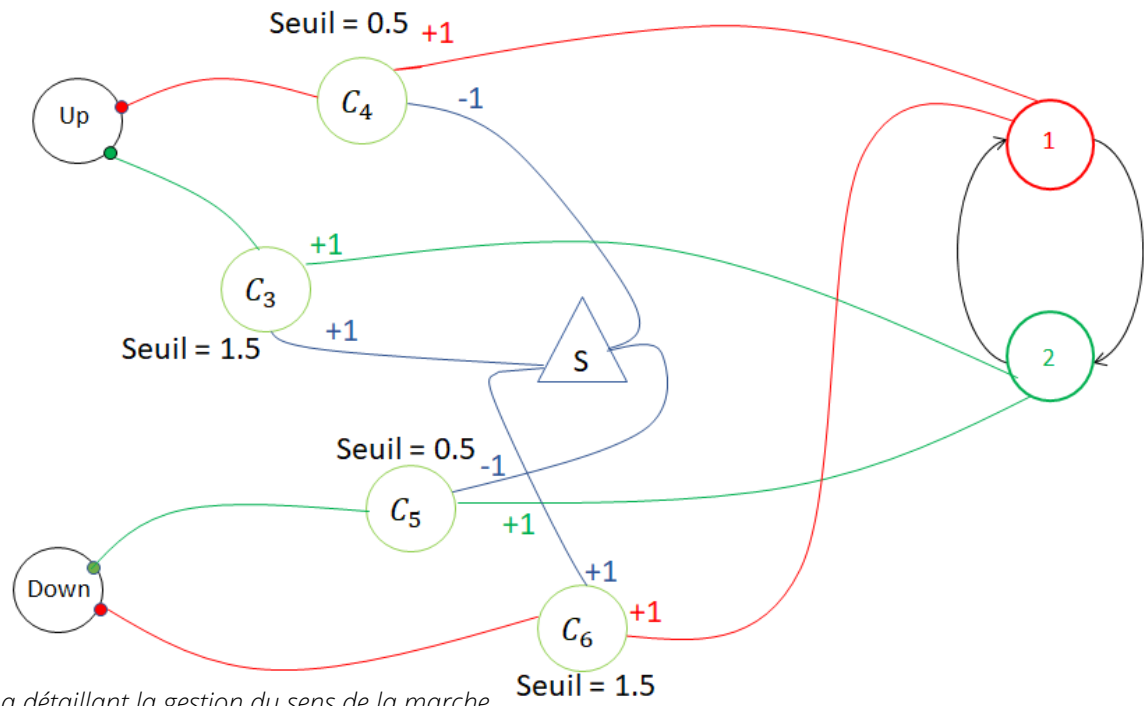


Schéma détaillant la gestion du sens de la marche

Le sens de la marche est dicté par le neurone sensoriel **S**. S'il n'est pas stimulé, le système va en marche avant. Le *neurone 1* envoie une décharge d'intensité 1 sur **C4**, ce qui est plus grand que sa valeur seuil, la décharge continue de circuler jusqu'au neurone *Up*, alors qu'elle est bloquée par **C6** dont le seuil est 1,5. Inversement quand c'est le *neurone 2* qui décharge. Comme *Up* est activée en même temps que *Forward* et *Down* en même temps que *Back*, on a un **mouvement vers l'avant**. Si **S** est stimulé, lorsque le **neurone 1** décharge, **S** envoie une décharge inhibitrice de -1 sur **C4** et excitatrice de +1 sur **C6**. La somme des décharge étant inférieure à la valeur seuil sur **C4**, la décharge est bloquée, alors qu'elle dépasse 1.5 sur C6. Le neurone *Down* est alors activé en même temps que le neurone *Forward* par l'intermédiaire du *neurone 1* du CPG. Il en est ainsi avec **Up** et **Back** pour que le système recule tel qu'illustré par les séquences de décharges sur le graphique ci-dessous.

