

Locomotion d'un hexapode

"Nous attestons que nous sommes les auteurs du présent travail et que tout ce qui a été emprunté est attribué à sa source et proprement référencé."

Yahya Ettahi (25h), Yasmine Wahbi (23h), Sidi Yaya Arnaud Yarga (26h)

GEI723 - Problème I - Automne 2020

INTRODUCTION

Conception d'un réseau de neurones à décharges non autonome permettant de générer les commandes de déplacement pour un hexapode. Puis, généraliser pour un nombre supérieur de pattes (100 pattes dans notre étude). Enfin, mettre en place un système semi-autonome.

CONTEXTE DE TRAVAIL

L'étude de la locomotion des insectes démontre la présence d'un système centralisé et d'un système décentralisé. Le système centralisé permettant d'imprimer un mouvement général tandis que le système décentralisé permet à chaque patte de réagir indépendamment. Ces deux approches ont fait l'objet d'étude dans la littérature [1] [2] et permettent d'obtenir des résultats différents. Dans le cadre de notre étude nous avons opté pour un système centralisé qui permet d'avoir un réseau plus simple même si la l'adaptabilité est plus faible [2]. De plus, cette architecture offre de meilleures performances car on utilise un seul générateur central contrairement au système décentralisé qui en nécessite un pour chaque patte. Pour la réalisation de ce projet, nous avons utilisé la bibliothèque brian2 de python ainsi que l'environnement Jupyter notebook.

OBJECTIFS

- Déplacement en ligne droite
- Déplacement en cercle
- Extension de la solution au mille pattes
- Déplacement à gauche (à droite) s'il heurte un obstacle sur son flan droit (gauche)
- Recul s'il heurte un obstacle de face
- Doit pouvoir accélérer s'il est heurté par derrière.

DESCRIPTION DU SYSTÈME

Une patte comprend : 3 neurones moteurs (mouvement avant, mouvement élévation de la patte, mouvement arrière), 2 neurones connecteurs; un pour

la commande avant et l'autre connecteur pour la commande arrière. D'autre part, on a les neurones commandes composés de quatre neurones commandes (commande avant, commande arrière, commande droite, commande gauche). La figure 1 (dans l'annexe) représente clairement cette architecture. Pour que l'hexapode détecte les obstacles, nous avons mis en place 4 capteurs. (voir la figure 2 dans l'annexe).

Equations :

Nous avons utilisé deux équations qui sont:

```
eqsPrincipale = ""
dv/dt = ((s*int(v > 0 or s>0))+ l - v)/tauV :1
ds/dt = -s/tauS :1
l :1
tauV : second
tauS : second ""
.....
```

```
eqsSecondaire = ""
dv/dt = (l - v)/tau :1
tau = 10*ms : second
l :1 ""
```

L'équation secondaire est celle d'un neurone LIF, Elle est utilisée pour nos neurones capteurs et connecteurs. Sa proximité avec la biologie et sa simplicité sont les raisons pour lesquelles nous l'avons préféré au neurone IF et Izhikevich. L'équation principale est celle d'un LIF avec ajout d'une variable s qui représente l'activité synaptique. Cette variable permet d'effectuer une inhibition mutuelle plus efficace d'une part, et nous permet d'engendrer une excitation plus durable (même après la décharge) d'autre part. Cette équation est utilisée pour les neurones commandes et les neurones moteurs. Ceci est inspiré d'un exemple[3] développée par Brian2

CONDITIONS EXPÉRIMENTALES

Les valeurs des paramètres utilisés pour ce problème ont été obtenues de manière empirique. Tous nos neurones ont un seuil de 0.9, l = 0 et les synapses inhibitrices ont un poids de -1. Pour simuler l'activation des neurones commandes ou des capteurs, on met la valeur de l à 1.

Neurones moteurs:

- tauV = 10*ms
- tauS = 20*ms

Neurones connecteurs:

- tau=10*ms
- Poids des synapses vers les neurones moteurs: 1.8 si s du neurones moteur est <=0 et 0.9 sinon.

Neurones commandes:

- tauV=20*ms
- tauS = 20*ms
- Poids des synapses vers les neurones connecteurs: 1

Neurones capteurs:

- tau = 10*ms
- Poids des synapses vers les neurones commandes: 1.3 pour le capteur arrière (accélération) et 1 pour tous les autres.

RÉSULTATS ET INTERPRÉTATIONS

Dans les graphes des figures 3, 4, 5, et 7, l'axe des ordonnées représente l'indice des neurones, l'axe des abscisses représente le temps, les lignes verticales en pointillés représentent les instants de décharges. Par ailleurs, pour les "petits triangles" se sont mis en place pour montrer la direction du déplacement. On constate donc que les pattes réagissent (mouvement avant ou arrière) les unes après les autres en raison du délai introduit dans les synapses. Pour la figure 5 qui concerne le déplacement en cercle, on remarque que les pattes situées du côté gauche décharge 2 fois plus. Cela induit un mouvement de rotation comme indiqué dans [2]. De plus, la figure 6 dévoile clairement le nombre de décharges de la patte 1 du côté gauche (2 fois) contrairement à celle du droit (la patte 0) qui décharge une seule fois et dans le même délai. La figure 4 montre que la généralisation aux 100 pattes fonctionnent correctement et prouve le même comportement des pattes comme décrit précédemment. Ces résultats sont conformes à ceux trouvés dans la littérature, notamment la démarche en vague cependant les aspects mécaniques (exemple : angles de la patte) ne sont pas modélisés.

REMERCIEMENTS

Nos sincères remerciements à l'équipe qui a mis à notre disposition l'accès à la bibliothèque Brian2

RÉFÉRENCES

[1]: 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 : <https://doi.org/10.1162/neco.1992.4.3.356>

[2]: M. OTIS, "Développement d'un système de propulsion pour un biomimétique hexapode avec un ionomère perfluorosulfonique," Master's thesis, Université de Sherbrooke, 2004

[3]: <https://brian2.readthedocs.io/en/stable/examples/CUBA.html>

[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.

CONCLUSION

A partir des résultats trouvés, on peut conclure que la synchronisation entre les pattes, les délais de décharge et les inhibitions entre les neurones ont une influence forte sur la rapidité du déplacement. Ainsi, l'implémentation de quelques inhibitions entre des pattes adjacentes pourra optimiser la vitesse du déplacement[4].

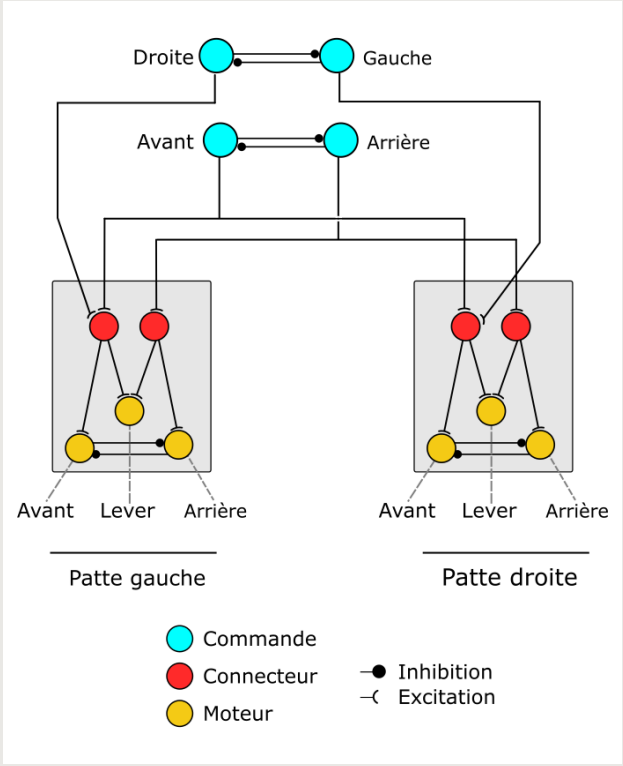


Figure 1 : architecture détaillée d'une patte

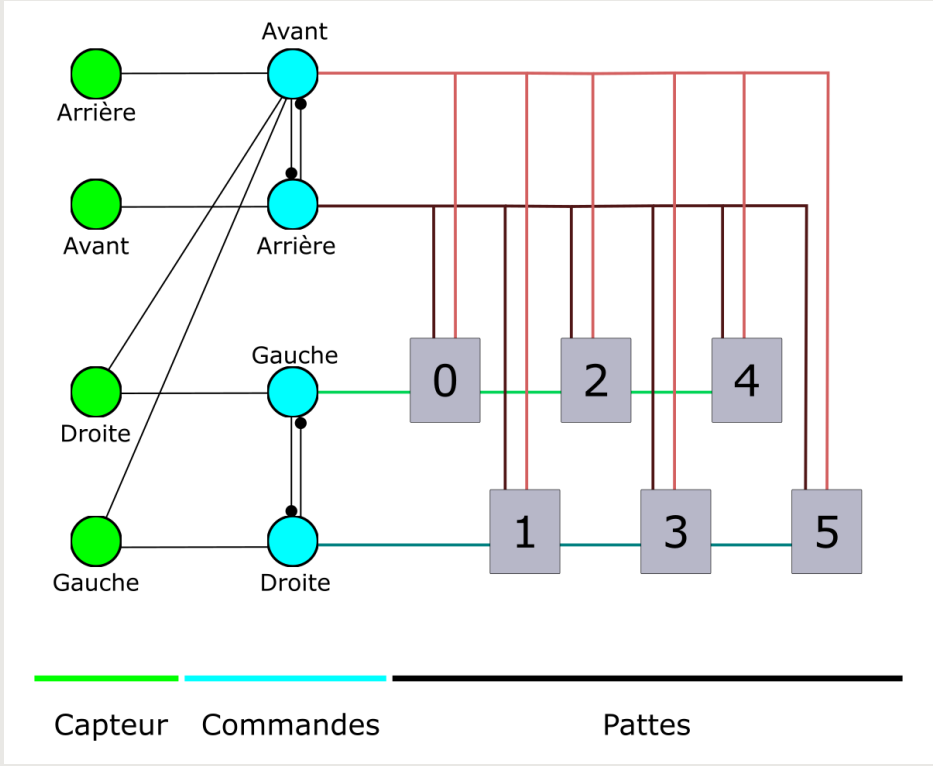


Figure 2 : architecture général

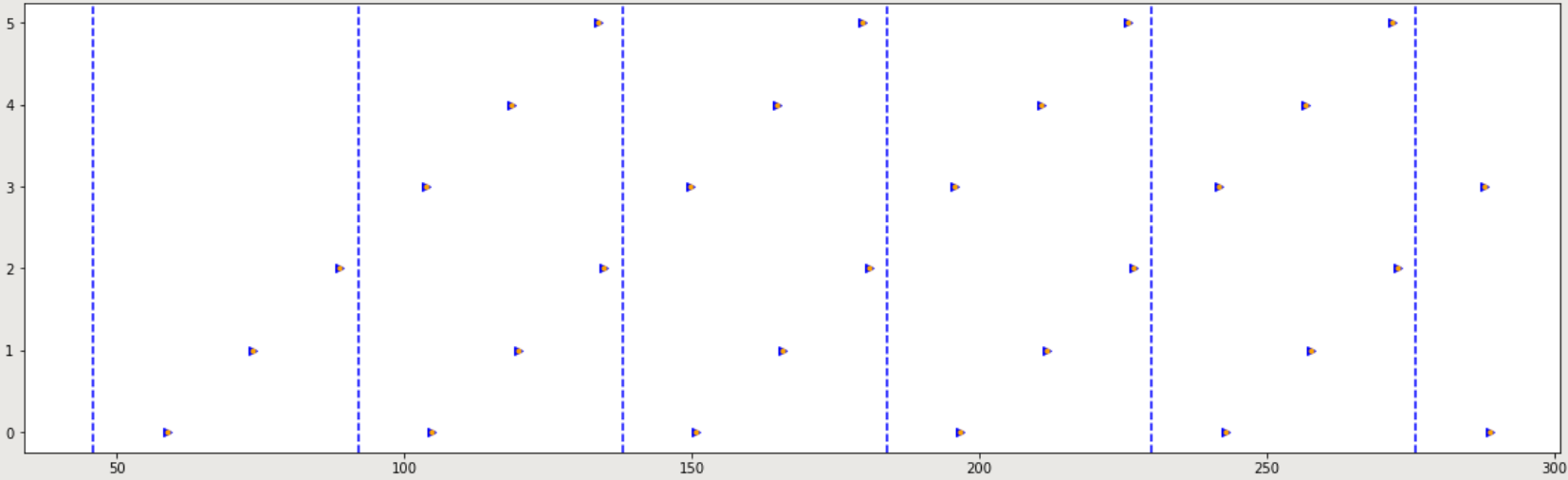


Figure 3 : résultat du scénario 'déplacement en avant' pour un hexapode montrant l'instant de décharge de chaque patte

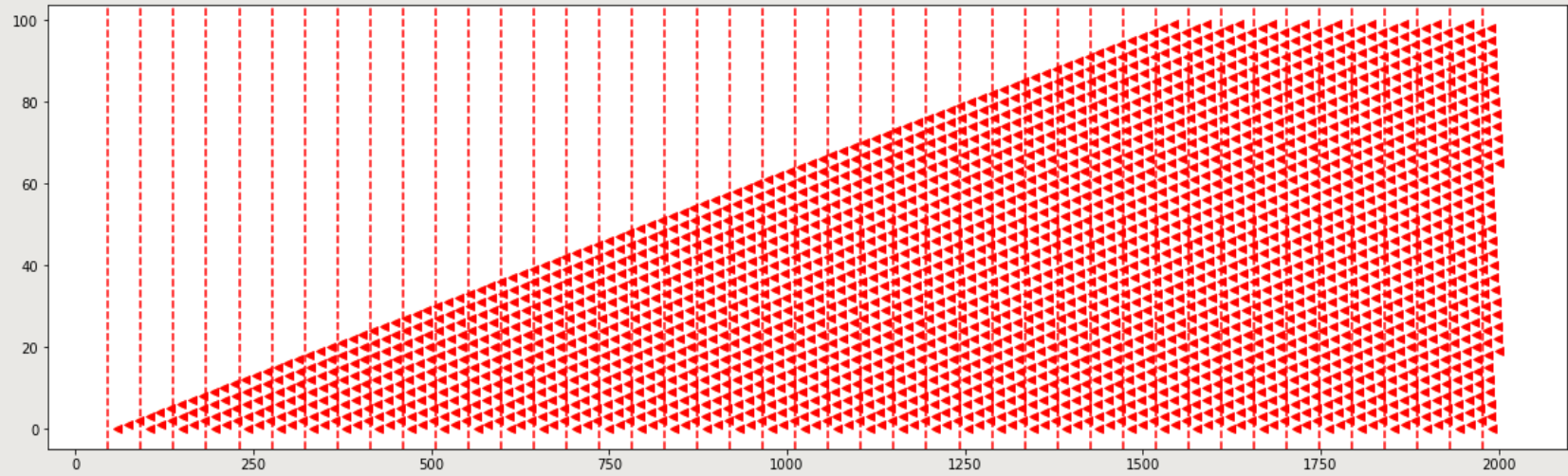


Figure 4 : résultat du scénario 'déplacement en arrière' d'un 100 pattes montrant l'instant de décharge de chaque patte

ANNEXE

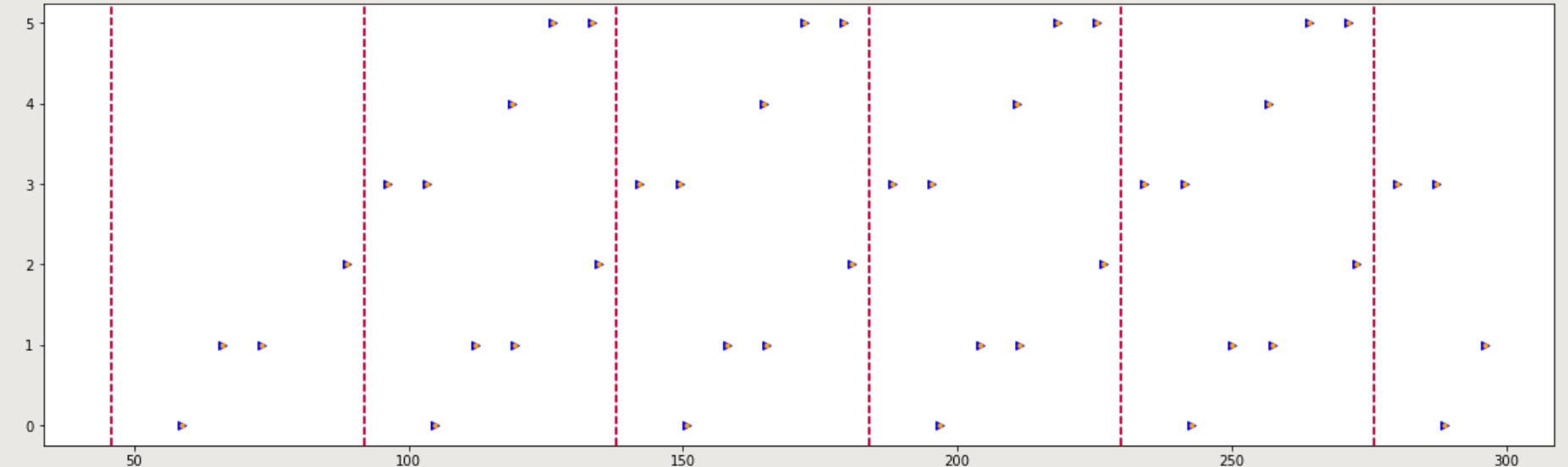


Figure 5 : résultat du scénario 'déplacement avant et droite (cercle)' d'un hexapode montrant la différence entre le nombre de décharge des deux côtés (2 décharges du côté gauche et une décharge du côté droit)

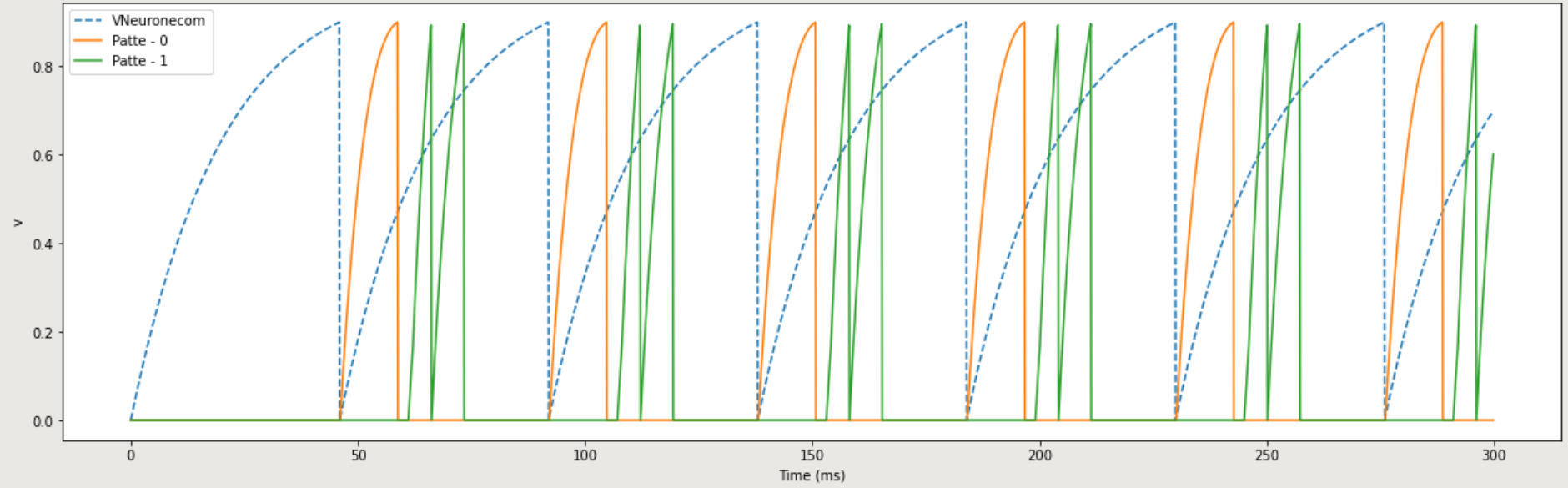


Figure 6 : Evolution du potentiel de neurone Commande d'avant montrant aussi la différence entre le nombre de décharges des pattes gauches (la patte 1 comme exemple) et droites (la patte 0 comme exemple)

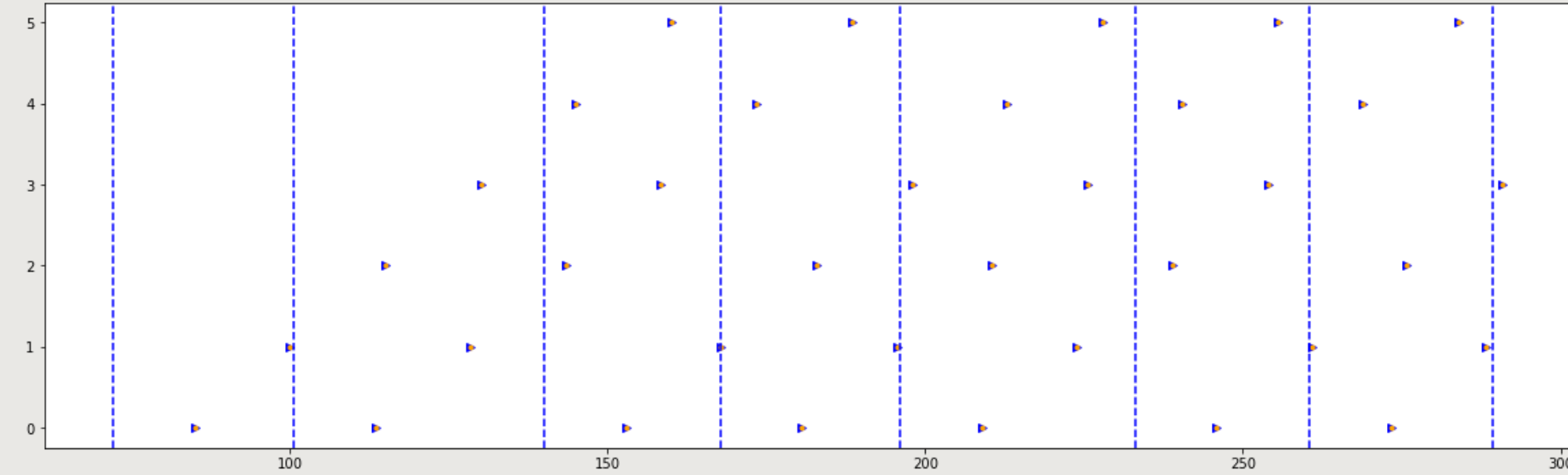


Figure 7 : résultat du scénario 'Accélération' après avoir heurté un obstacle arrière et qui montre les instants de décharge plus proche des uns aux autres