



# **Projet de « Computational Neurosciences and Robotics »**

**Utilisation du modèle Rowat-Selverston pour  
l'interaction sociale avec le robot Nao**

**Bilal FOURKA**

**Encadré par : Mr. Patrick HÉNAFF**

**2020-2021**

## Introduction

Le sujet de ce projet est l'utilisation du modèle Rowat-Silverston de synchronisation neuronale sur le robot Nao afin qu'il puisse suivre le mouvement de salutation avec la main. Le robot doit pouvoir utiliser le modèle afin de synchroniser le mouvement de sa main avec celui d'une personne devant lui. On souhaite donc combiner les simulations sous Brian2 et les scripts de contrôle du robot Nao pour aboutir à une interaction du robot en fonction du signal qu'il reçoit en entrée.

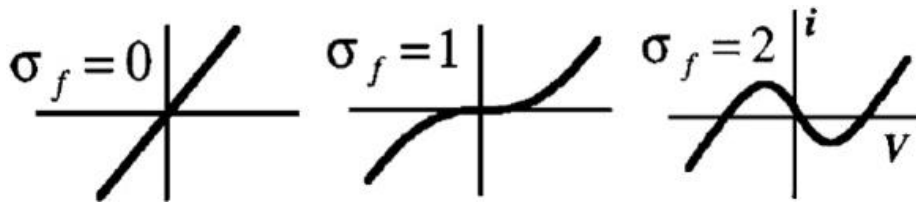
Rowat-Silverston (1993) présentent un modèle simple d'un neurone pour lequel deux groupes de courants sont identifiés : un courant rapide et un courant lent, chacun défini par une équation différentielle de premier ordre. Le courant rapide est défini par l'équation. 1 et courant lent par Eq. 2.

$$\tau_m \frac{dV}{dt} = -(F(V, \sigma_f) + q - I_{inj}) \quad (1)$$

$$\tau_s \frac{dq}{dt} = -q + q_\infty(V) \quad (2)$$

Avec  $\tau_m < \tau_s$ .  $\tau_m$ , est la constante de temps de la membrane du neurone,  $\tau_s$  est la constante de temps de l'activation des courants lents.  $I_{inj}$ , est le courant injecté,  $V$  la tension de la membrane cellulaire, et  $q$  le courant lent.

$F(V, \sigma_f)$ , défini à l'équation. 3, est une fonction courant-tension non linéaire pour le courant rapide. La figure 1 montre que la forme de courant rapide  $F(V, \sigma_f)$  peut être linéaire ( $\sigma_f = 0$ ), non linéaire mais bijective ( $\sigma_f = 1$ ) et non linéaire et non bijective ( $\sigma_f = 2$ ).  $F(V, \sigma_f)$  est une partie fondamentale du modèle RS parce que cette fonction induit des comportements différents pour le neurone (oscillant amorti, potentiels de plateau ou oscillant) suivant la valeur du gain de courant rapide,  $\sigma_f$ .



**Figure1 : Effet de  $\sigma_f$  sur la fonction  $F(V, \sigma_f)$**

$$F(V, \sigma_f) = V - A_f \tanh\left(\frac{\sigma_f}{A_f} V\right) \quad (3)$$

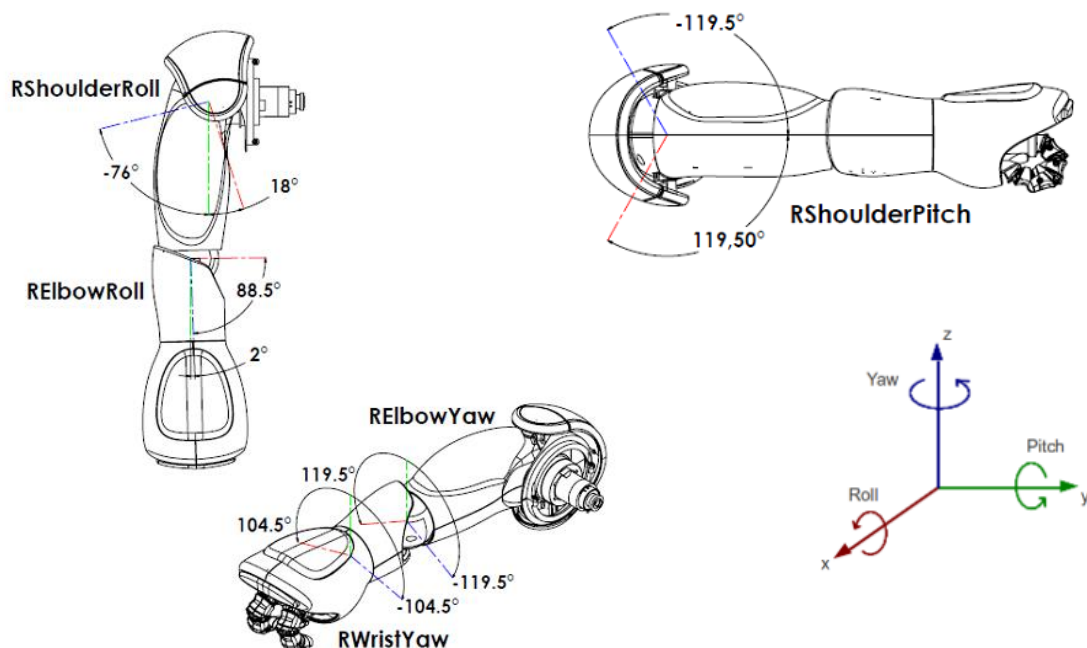
L'évaluation des performances de la solution apportée a été effectuée à travers la réalisation de différentes simulations. Le logiciel CoppeliaSim a été utilisé, avec le modèle de robot Nao intégré.

## I. Description de l'algorithme implémenté

L'implémentation de l'algorithme Rowat-Selverston de synchronisation utilisé lors du cours a été conservé. Celle de l'algorithme du contrôle du robot a, quant à elle, été codée en Python, en s'inspirant du script python du robot serpent. Enfin, l'interface Python-CoppeliaSim fournie a également été réutilisée.

Un certain nombre de modifications ont été nécessaires au bon fonctionnement de la solution. Puisqu'on ne possède pas d'un algorithme de détection du mouvement de salutation, il fallait travailler avec un signal sinusoïdal afin d'imiter le mouvement d'une main lors de la salutation. Ce signal doit être en conformité avec les spécifications du robot.

### A. Les spécifications du robot Nao (bras droit)



**Figure2 : Spécifications du robot Nao**

Le bras du robot Nao contient plusieurs articulations qu'on peut contrôler. RShoulderPitch est l'articulation responsable du mouvement de tout le bras autour de l'axe 'y'.

Range (degrees)	Range (radians)
-119.5 to 119.5	-2.0857 to 2.0857

RshoulderRoll quant à elle est l'articulation responsable du mouvement de tout le bras autour de l'axe 'x'.

<b>Range (degrees)</b>	<b>Range (radians)</b>
-76 to 18	-1.3265 to 0.3142

Ce sont deux articulations importantes pour le démarrage de la salutation. En plus, RelbowYaw est une articulation qui permet de faire une rotation au niveau de l'avant-bras du robot.

<b>Range (degrees)</b>	<b>Range (radians)</b>
-119.5 to 119.5	-2.0857 to 2.0857

Cependant, la partie la plus intéressante dans notre projet est RElbowRoll qui est responsable du mouvement autour de l'axe 'x'. Cette articulation sera celle qui imite le mouvement de salutation humaine.

<b>Range (degrees)</b>	<b>Range (radians)</b>
2 to 88.5	0.0349 to 1.5446

## B. Le contrôle du robot Nao :

Le programme de contrôle du robot commence par charger la scène du robot en se connectant avec CoppeliaSim a travers l'adresse IP 127.0.0.1 et le port 19997. Ce sont les configurations par défaut sous Vrep. Une fois que le programme soit connecté au robot, le programme lit les articulations qui existe dans le robot. Pour ce faire, il faut saisir le nom de chaque articulation ainsi que l'id du client (Robot) :

```
res, v1_motor = vrep.simxGetObjectHandle(client_id, 'RShoulderPitch3',  
vrep.simx_opmode_one-shot_wait)
```

Ensuite, le programme python exécute une boucle « while », où on modifie la valeur des degrés données aux articulations à chaque itération. Pour cela on crée une liste contenant les commandes :

```
command_h = [1.5, 0, 0, 0]  
command_v = [-1.5,-1.5,0,0]
```

La mise à jour de la valeur de l'articulation est effectuée avec la commande suivante :

```
vrep.simxSetJointTargetPosition(client_id, v1_motor, command_v[0],  
vrep.simx_opmode_one-shot_wait)
```

## C. L'utilisation de l'algorithme de Rowat-Selverston :

Le programme crée une simulation avec la bibliothèque Brian2. Cette simulation prend en entrée un signal sinusoïdal. Puisque l'articulation RelbowRoll varie

entre 0.0349 et 1.5446 radian, le signal sinusoïdal doit être entre ces deux valeurs. Pour cela le signal utilisé est :

**stimulus = TimedArray([0,0,40],dt = 10\*ms)**

**i\_inj = 0.8\*sin(30\*Hz\*stimulus(t)\*t)**

Le résultat de la simulation sera appliqué à l'articulation RelbowRoll avec une échelle de 0.03 à 0.8 en ajoutant la constante 0.8.

Le code source du projet est disponible sur le dépôt github suivant : <https://github.com/fourkaBilal1/neuroscience-robotics>

## I. Analyse du comportement observé

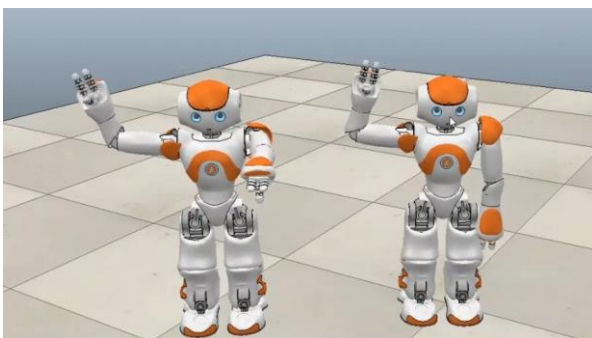
Afin d'analyser le comportement du robot, une série de tests a été effectuée sur le robot Nao, les valeurs des paramètres du modèle étant modifiés. On s'attend à ce que ces derniers aient un impact sur la performance du robot selon le type du test et sa complexité.

Pour contourner l'absence de la salutation de l'humain, un deuxième robot sera utilisé pour qu'il reproduit le I\_inj. Cela nous permet de comparer les deux robots et voir la performance de synchronisation du premier (celui à gauche avec le bras tendu vers l'avant) vis-à-vis du deuxième (celui à droite avec le bras gauche en bas).

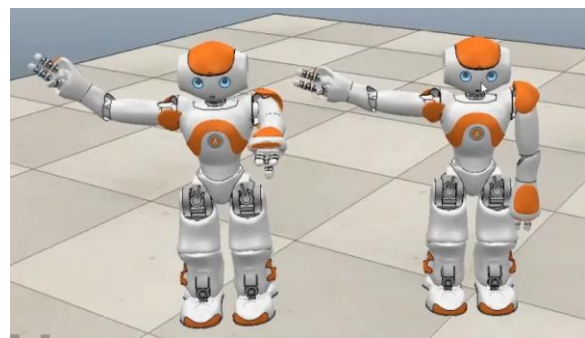
### A. Test 1 :

Dans ce premier test le robot qui synchronise démarre avec un mouvement de salutation aléatoire (Initialisation aléatoire de v et q). Le deuxième robot-humain démarre après une vingtaine de millisecondes avec un mouvement sinusoïdale :

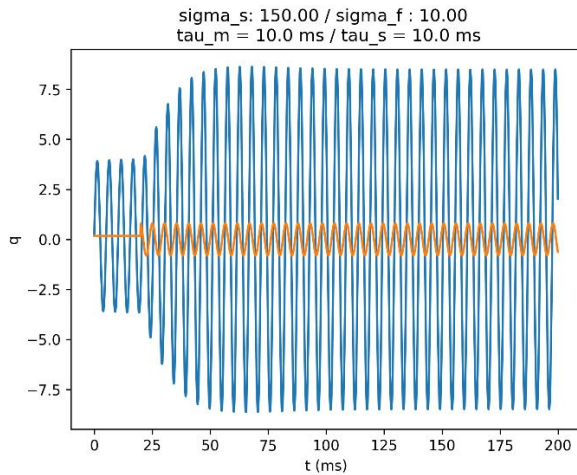
**i\_inj = 0.8\*sin(30\*Hz\*40\*t)**



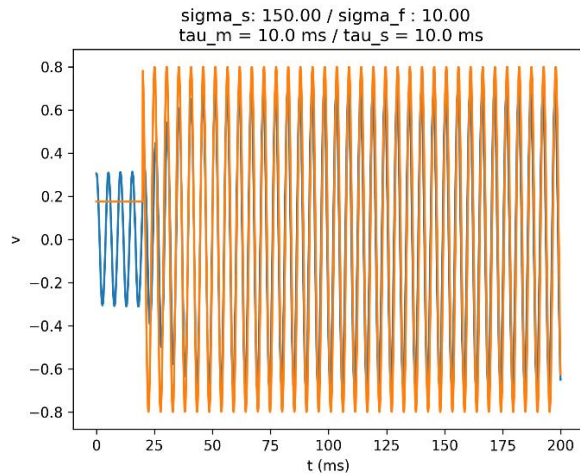
**Figure 3 : Début de simulation**



**Figure 4 : Fin de simulation**



**Figure 5 : Courant q et I<sub>inj</sub>**



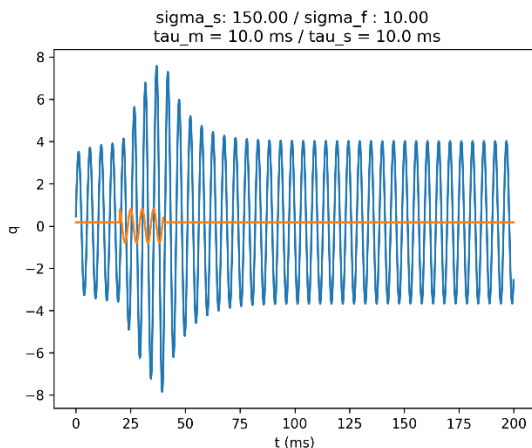
**Figure 6 : Courant v et I<sub>inj</sub>**

## Analyse :

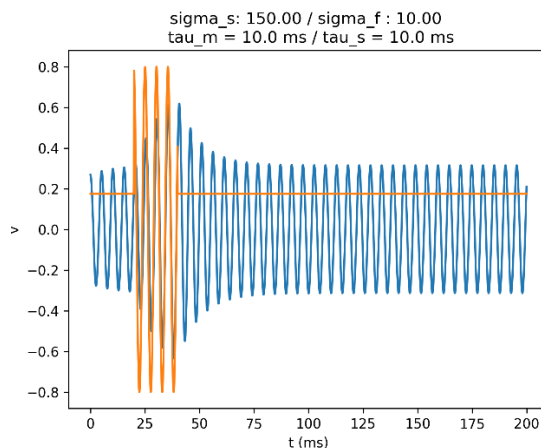
On constate sur la vidéo et sur les graphes (figure 5 et 6) que le courant v est le plus adapté dans ce cas et celui qui sera utilisé pour la détection du courant. Aussi, on constate qu'au début de la simulation, les deux robots faisaient la salutation avec une différence de phase et d'amplitude. Ce qui le montre la figure 3. A la fin de la simulation (voir la figure 4), le premier robot arrive à se synchroniser avec le deuxième en phase et en amplitude ce qui signifie que ces paramètres les meilleurs paramètres pour résoudre ce problème de synchronisation.

## B.Test 2 :

Dans ce deuxième test, nous allons procéder de la même façon en effectuant une modification. Le courant d'entrée I<sub>inj</sub> sera coupée après 20 millisecondes de son démarrage. On veut tester la possibilité d'arrêt du mouvement si le robot se synchronise bien avec le deuxième robot-humain.



**Figure 7 : Courant q et I<sub>inj</sub>**



**Figure 8 : Courant v et I<sub>inj</sub>**

## Analyse :

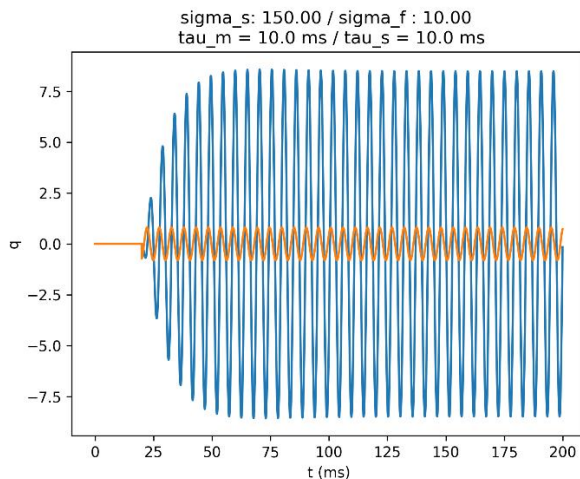
Sur les figures 7 et 8, on constate qu'on démarrant d'un mouvement aléatoire, le premier robot se synchronise très vite au mouvement du deuxième comme dans le premier test. Mais, le robot continue son mouvement de salutation même après l'arrêt du deuxième robot. On peut dire que ces paramètres fonctionnent pour arriver à la synchronisation mais jamais à l'arrêt du premier robot. C'est comme si le robot a sa propre phase et amplitude et à chaque fois un courant  $I_{inj}$  est présent il se synchronise avec lui.

## C.Test 3 :

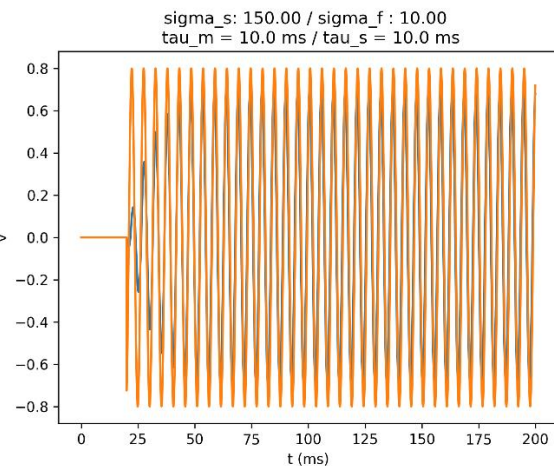
Dans ce troisième test nous allons essayer d'utiliser la même configuration du premier test en changeant l'entrée aléatoire du premier robot afin de voir un mouvement qui se déclenche avec le début du mouvement chez le deuxième robot-humain. Donc nous allons enlever les deux lignes suivantes du code :

**#G.v = 0.5\*rand()**

**#G.q = 0.5\*rand()**



**Figure 9 : Courant q et  $I_{inj}$**



**Figure 10 : Courant v et  $I_{inj}$**

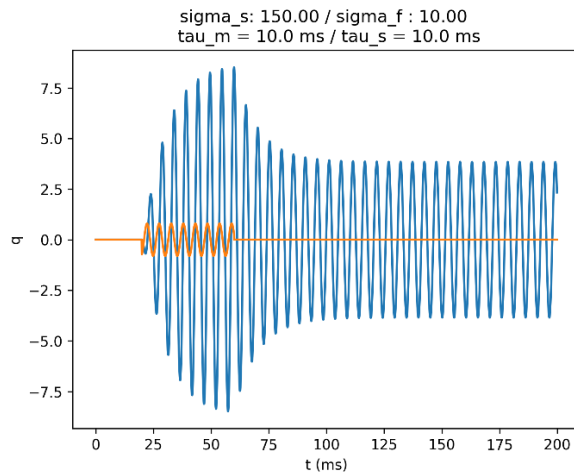
## Analyse :

Sur les figures 9 et 10, on observe que la seule différence qui existe vis-à-vis le premier test se manifeste dans le début de la simulation. Le premier robot, étant à l'arrêt jusqu'à l'arrivée du signal  $I_{inj}$ , prend plus de temps afin de se synchroniser avec le deuxième robot-humain. Le comportement devient plus humain en enlevant l'initialisation aléatoire de v et q.

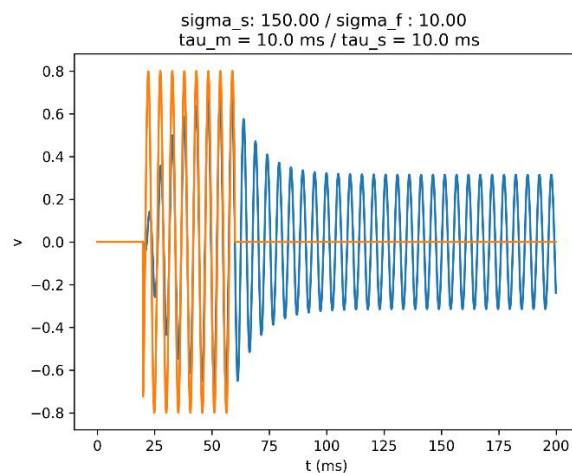


## D. Test 4 :

L'objectif de ce quatrième test est d'améliorer l'arrêt du mouvement et de comparer l'étape permanente du robot lorsque le deuxième robot s'arrête après 40 millisecondes de son déclenchement en supprimant l'initialisation aléatoire de  $q$  et  $v$ . Puisque



**Figure 11 : Courant  $q$  et  $I_{inj}$**



**Figure 12 : Courant  $v$  et  $I_{inj}$**

## Analyse :

Sur les figures 11 et 12, on remarque que le phénomène observé dans le test 3 au début de la simulation se reproduit. Mais, en arrêtant le deuxième robot-humain, le premier robot ne s'arrête pas et continue à faire le mouvement avec une amplitude plus basse.

## E. Test 5 :

Dans ce dernier test, Nous comprenons très bien qu'il ne faut pas initialiser les valeurs de  $q$  et  $v$  afin d'avoir un signal qui démarre avec le déclenchement de mouvement chez le robot-humain. En plus, le test 4 montre qu'on est proche d'un mouvement réaliste et qu'il faut travailler l'arrêt du premier robot lorsque la valeur de  $I_{inj}$  est nulle. Des changements dans le modèle de Rowat-Silverston peuvent avoir lieu.

## Analyse :

En se basant sur les travaux dirigés faits au cours, il était indispensable de réduire la valeur de  $\sigma_f$  afin d'avoir un pseudo arrêt du mouvement du premier robot. Encore, une remarque était que le signal de  $q$  donne de meilleurs résultats que celui de  $v$ . Donc, il fallait



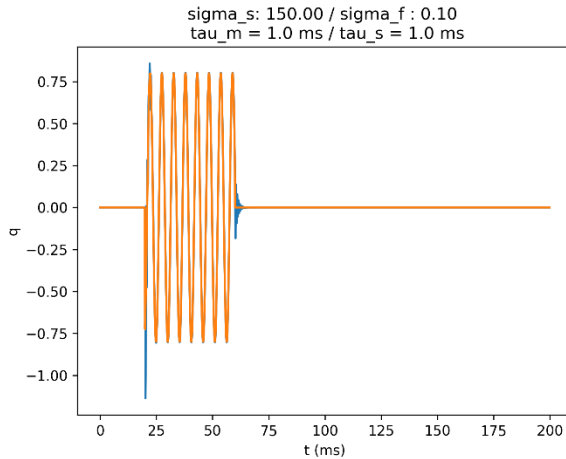
réduire la valeur du  $\tau_m$  afin de favoriser une plus grande valeur de  $q$ . A la fin, ce les modifications suivantes qu'il fallait faire sur le modèle et sur le script.

**$A_f = 1$**

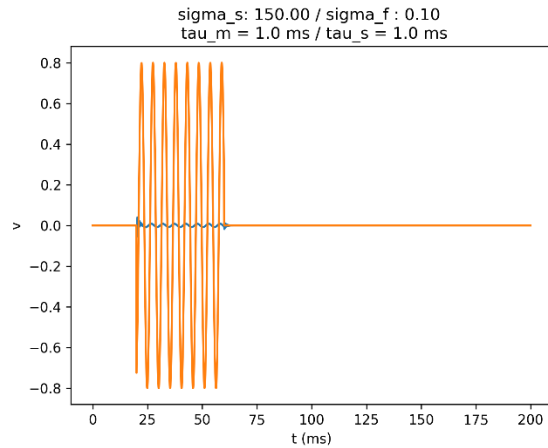
**$\sigma_f = 0.1$**

**$\tau_m = 1 \text{ ms}$**

**$b = (\text{trace.q}[0]).\text{tolist}()$**



**Figure 13 : Courant  $q$  et  $I_{inj}$**



**Figure 14 : Courant  $v$  et  $I_{inj}$**

Sur ces deux figures, on observe que le courant  $q$  fonctionne très bien pour un mouvement humain qui démarre avec le déclenchement du mouvement par un robot-humain. L'arrêt du premier robot s'effectue d'une manière très fluide et très réaliste. La vidéo de simulation du test 5 montre ce phénomène. On peut dire qu'on a réussi à construire, à la base du modèle Rowat-Selverston, une simulation de robot Nao entrain de synchroniser sa salutation avec un humain qui lui envoie un salut d'une façon humaine.

L'ensemble des vidéos des simulations sont disponibles sur le drive suivant : [https://drive.google.com/drive/folders/1B6oWXmUj9qh\\_Fyscwu40tYzFwylCt\\_gy?usp=sharing](https://drive.google.com/drive/folders/1B6oWXmUj9qh_Fyscwu40tYzFwylCt_gy?usp=sharing)

## Conclusion

Ce projet a été très enrichissant, car il m'a permis de découvrir le domaine de la Neuroscience et ses contraintes, en prenant part à un projet concret. Ce dernier m'a également fait prendre conscience que le domaine de la neuroscience nécessite une analyse très détaillée et des connaissances dans la biologie afin de comprendre les phénomènes observés dans les comportements biologiques des êtres vivants, dont la source est parfois difficile à cerner.