



HAL
open science

Etude sur l'intérêt des modèles biologiques de réseaux de neurones pour la synthèse de rythmes locomoteurs adaptatifs

Thierry Hoinville, Patrick Henaff, Stéphane Delaplace

► To cite this version:

Thierry Hoinville, Patrick Henaff, Stéphane Delaplace. Etude sur l'intérêt des modèles biologiques de réseaux de neurones pour la synthèse de rythmes locomoteurs adaptatifs. *Journal Européen des Systèmes Automatisés (JESA)*, 2007, Vol. 41 (N°3/4), pp.413-435. hal-00519907

HAL Id: hal-00519907

<https://hal.science/hal-00519907v1>

Submitted on 21 Sep 2010

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Etude sur l'intérêt des modèles biologiques de réseaux de neurones pour la synthèse de rythmes locomoteurs adaptatifs.

Thierry Hoinville — Patrick Hénaff — Stéphane Delaplace

Laboratoire d'Ingénierie des Systèmes de Versailles (LISV)
Université de Versailles Saint Quentin
10-12 Av de l'Europe, 78140 Vélizy, France
thierry.hoinvillef@liris.uvsq.fr

RÉSUMÉ. Cet article présente les résultats d'une étude qui vise à accroître la capacité d'adaptation des robots à pattes. Le contrôleur est constitué d'un réseau de neurones, synthétisé par algorithme évolutionniste, auxquels des mécanismes adaptatifs inspirés de la biologie sont incorporés. Nous comparons les performances des contrôleurs suivant trois modèles de neurones à complexité biologique croissante. Sur l'exemple de la commande du déplacement d'une structure monopode à trois degrés de liberté, nous montrons l'intérêt des modèles à dynamique interne dont les paramètres sont calculés à partir de mécanismes inspirés de la biologie plutôt que d'être calculés par évolution artificielle. Nous incorporons ensuite ces mécanismes dans un contrôleur neuronal qui commande la vitesse du robot monopode glissant sur un rail et soumis à des perturbations externes. Nous montrons que ces mécanismes permettent l'adaptation en ligne du contrôleur donc de l'allure du cycle de locomotion et qu'ils maintiennent la performance en vitesse du robot.

ABSTRACT. This article presents a preliminary study for work that aims at increasing the capacity of adaptation of legged robots. The controller consists of one neural networks, synthesised by evolutionary algorithm, to which adaptive mechanisms inspired of biology are implemented. We compare the performances controllers according to three models of neurons with increasing biological complexity. We thus show the interest of the dynamic models of neurons for which one parameter must be calculated from mechanisms inspired by biology rather than to be advance. We incorporate then these mechanisms in a neural controller of a robot made up of one leg of 3 dof pushing a body that is sliding on a rail. We show that these mechanisms allow the on line adaptation of the controller and the cycle of locomotion which maintain the performance of speed of the robot.

MOTS-CLÉS: robotique évolutionniste, locomotion, adaptation, contrôle neuronal,

KEYWORDS: evolving neural controller, locomotion, adaptive behaviour

1. Introduction

La robotique évolutionniste est basée sur la constatation que dans la nature, l'évolution a permis d'aboutir à des êtres plus ou moins complexes, très bien adaptés à leur environnement et aux incidents qu'ils peuvent rencontrer. Elle essaye donc de reproduire ce que la nature a su faire. Cependant, les travaux les plus significatifs (Doncieux, 2003)(Gallagher *et al.*, 1996)(Ijspeert *et al.*, 1998)(Meyer *et al.*, 2002)(Nolfi, 1997) montrent qu'utiliser la synthèse évolutionniste pour créer totalement à partir d'un seul neurone qui se développe, un contrôleur neuronal adaptatif apte à faire face à des situations complexes, reste encore un objectif difficile à atteindre. Ainsi dans (Ijspeert, 2001), l'auteur utilise l'évolution artificielle de façon incrémentale pour construire un contrôleur neuronal capable de commander les mouvements d'une salamandre simulée quelque soit l'environnement dans lequel elle évolue.

La synthèse évolutionniste consiste à faire évoluer automatiquement différents éléments de la structure d'un réseau de neurones (connexions, les paramètres des neurones, types de synapses, règles d'adaptions...). C'est une technique d'optimisation avantageuse car elle nous permet d'explorer automatiquement l'ensemble de l'espace de recherche et d'identifier des solutions efficaces qui n'auraient pas été envisagées par une synthèse manuelle. Néanmoins, la synthèse évolutionniste est limitée par la dimension de l'espace de recherche qui s'accroît très vite en fonction de la complexité du problème à résoudre et du nombre de paramètres du réseau de neurones, le temps de calcul devenant alors trop prohibitif. L'analyse du problème et son codage génétique sont donc des étapes importantes en robotique évolutionniste. Dans cet article, nous n'étudierons pas dans les détails les choix que nous avons faits pour la synthèse évolutionniste car nous l'utilisons comme un outil d'optimisation et ce n'est donc pas notre sujet de recherche.

Notre travail s'est inscrit dans le projet IRON¹ soutenu par l'ex-programme ROBEA² du CNRS. Ce projet avait pour objectif scientifique de comprendre le fonctionnement, la synergie et la robustesse de certains mécanismes adaptatifs empruntés à la biologie en les intégrant dans un contrôleur neuronal d'un système artificiel (un robot à pattes) qui peut être confronté aux mêmes situations que celles que peut rencontrer un animal dans la nature, c'est à dire qu'il doit assurer sa «survie» pour accomplir sa mission sans l'aide de l'homme dans un milieu plus ou moins imprévisible et hostile. Ainsi les dommages ou les dysfonctionnements envisagés sur le robot concernent sa structure mécanique, ses actionneurs, ses capteurs ou son architecture de contrôle.

Cet article présente les résultats issus de la première phase du projet IRON qui concerne l'étude des modèles biologiques de neurones pour la synthèse d'allures de

¹ Implémentation RObotique de Neuro-contrôleurs adaptatifs, en partenariat avec l'AnimatLab (<http://animatlab.lip6.fr/>)

² www.laas.fr/robea

robots à pattes. Dans la première partie, nous étudions la robustesse des contrôleurs neuronaux en comparant leurs performances lorsqu'ils doivent commander le déplacement d'un corps cubique posé sur le sol et doté d'une seule patte à trois degrés de liberté. En faisant évoluer artificiellement trois types de réseaux de neurones à complexité biologique croissante et dont les paramètres sont calculés ou pas par évolution, nous mettons en évidence l'efficacité de deux mécanismes homéostatiques observés en biologie. Dans la seconde partie, nous montrons comment incorporer ces mécanismes dans le modèle de réseau le plus performant afin qu'il devienne adaptatif. Les résultats présentés dans la troisième partie montrent l'efficacité des modèles proposés pour adapter le rythme du cycle de locomotion aux perturbations s'exerçant sur le robot.

2. Intérêt des modèles biologiques adaptés à la locomotion

2.1. Méthodologie de synthèse de trois types de contrôleur

La synthèse évolutionniste nous permet d'implémenter automatiquement des mécanismes adaptatifs dans les réseaux de neurones en explorant l'ensemble de l'espace de recherche. Nous présentons dans cette section une étude comparative sur les performances de 3 types de contrôleurs pour lesquels la complexité biologique est accrue par l'apport de mécanismes inspirés de la biologie et dont les paramètres peuvent être calculés par l'évolution.

Le problème que nous posons est celui de la commande du déplacement, par la poussée, d'un « robot cubique » doté d'une seule patte. Ce robot (**Figure 1**) et son contrôleur peuvent être vu comme l'un des six constituants d'un robot hexapode comme cela est souvent le cas dans les travaux qui cherchent à s'inspirer de la nature pour construire des contrôleurs de robots à pattes (Shiniki *et al.*, 2006)(Von Twickel *et al.*, 2006).

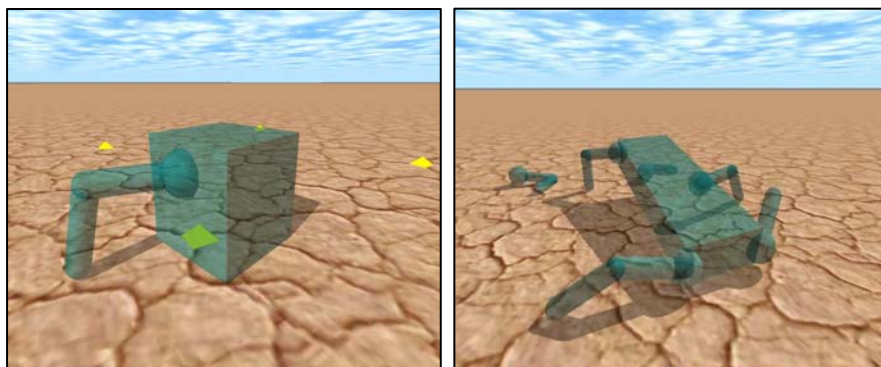


Figure 1. Robot à une patte et le modèle d'hexapode correspondant lorsqu'il perd 2 pattes.

Notre robot est donc constitué d'un corps cubique de masse 3 kg et de 20 cm de coté. Il est doté d'une patte pesant 1,5 kg de longueur 30 cm, constituée elle-même de 3 corps : une épaule sphérique à 2 degrés de liberté, un membre cylindrique lié à un autre membre cylindrique par une liaison à 1 degré de liberté constituant le genou (**Figure 2**). Une telle cinématique de patte à 3 degrés de liberté est intéressante car c'est un modèle qu'on retrouve chez certains animaux. De plus, cela permet de ne pas trop contraindre l'espace de recherche pour l'algorithme génétique, ce qui nuirait, pour la commande, à l'aboutissement de solution originale.

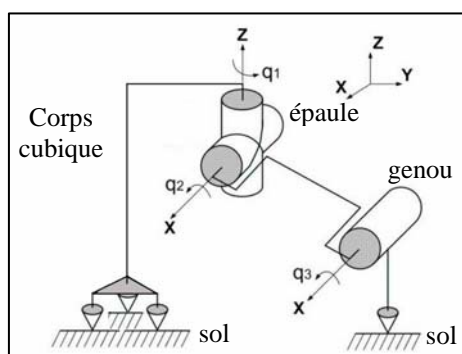


Figure 2. Morphologie et modèle cinématique du monopode posé sur le sol

L'outil de simulation utilisé est ODE³, un simulateur dynamique qui permet de simuler tous les systèmes multi-corps rigides. Il est rapide et suffisamment précis pour reproduire à chaque évaluation de contrôleur un comportement réaliste du robot sans perte de temps (10 secondes réelles sont simulées en quelques millisecondes sur un PC ordinaire). Les frottements secs et visqueux ainsi que les chocs entre corps solides y sont reproduits avec une grande fidélité. Le modèle de frottement sec est basé sur une loi de type Coulomb dont le cône de friction est approximé par une pyramide afin d'accélérer les calculs. Chaque corps est défini par sa forme (cubique ou cylindrique), sa taille et sa masse (considérée comme ponctuelle). Les liaisons entre les corps peuvent être cylindriques ou prismatiques.

Le contact pied/sol est considéré comme ponctuel car l'extrémité de la patte est sphérique. Les 3 degrés de liberté motorisés de la patte lui permettent d'effectuer des mouvements d'amplitude maximale dans l'espace cartésien. De plus, un contact avec le sol est toujours possible (dans les limites de l'espace atteignable de la patte), même si, compte tenu de la forme cubique du corps, l'effort de poussée produit par la patte provoque un pivotement important du cube sur une arête ou un sommet (l'effort de poussée peut provoquer un basculement complet du corps). ODE calcule en effet à chaque instant les points de contact entre la face inférieure du cube et le sol en supposant que ces points de contact sont situés sur 3 des sommets de la face. Ainsi, le cube peut pivoter sur une arête ou sur un sommet.

³ ODE, Open Dynamic Engine a été créé par Russel Smith (<http://www.ode.org/>)

Nous avons intégré dans ODE un modèle d’asservissement de moteur plus précis que celui existant par défaut et qui ne prend pas en compte une caractéristique couple/vitesse des moteurs suffisamment réaliste. Ce modèle d’asservissement, basé sur un contrôleur PID, est équivalent en boucle fermée à un système du second ordre réglé classiquement pour un coefficient d’amortissement de 0,45. Les moteurs considérés ont un couple de 2 N.m au démarrage et peuvent atteindre une vitesse maximum de 5,5 rd/s.

Le réseau de neurones doit calculer les 3 consignes de position articulaire à appliquer à la patte de façon à générer le cycle de locomotion qui déplace le cube le plus loin possible dans un temps fixé à 10 secondes et correspondant à la durée de l’évaluation. La fonction d’évaluation est donc la distance finale qui sépare le robot de sa position initiale à la fin de la simulation.

L’algorithme génétique est basé sur un modèle générationnel qui utilise la bibliothèque GA_Lib du MIT. Pour la sélection des individus, l’échantillonnage utilise la méthode dite « Stochastic Universal Sampling » qui est une technique optimale garantissant que chaque individu ne sera ni sur-sélectionné ni totalement ignoré (Baker, 1987). Nous standardisons ainsi notre plate-forme de simulation ODE+GA_Lib. Les paramètres de l’algorithme d’évolution listés dans le **Tableau 1** ont été déterminés empiriquement après plusieurs essais qui ne peuvent être décrits ici pour des raisons évidentes de clarté.

Paramètre	Valeur
Nombre de générations	1000
Nombre d’individus par génération	200
Probabilité de recombinaison (cross-over)	0,6
Probabilité de mutation	10^{-3}
Pression de sélection	1,1

Tableau 1. Paramètres de l’évolution artificielle

Nous proposons de comparer trois modèles de réseaux de neurones. L’architecture de ces 3 réseaux est identique (perceptron récurrent de Meeden), mais les modèles des neurones sont différents. Leurs paramètres (poids, biais, constantes de temps) sont calculés par évolution sauf pour le troisième modèle pour lequel la valeur du biais dépend des poids synaptiques.

- Les neurones du premier réseau sont classiques, sans comportement dynamique comme le résumant les équations suivantes :

$$y_i = \sum_{j=1}^{N_j} \omega_{ij} o_j \quad \text{et} \quad o_i = \text{sigmo}(y_i + \theta_i) \quad [1]$$

Seuls les poids ω_{ij} et le biais θ_i de chaque neurone sont calculés par l’évolution.

- Les neurones du second réseau sont des neurones dits « intégrateurs à fuite » ou « Leaky Integrators » qui possèdent une dynamique du premier d'ordre avec une non-linéarité liée à la fonction de décision (souvent une sigmoïde). Ce modèle reproduit assez fidèlement ce qui est observé en biologie pour la locomotion (Beer, 1995).

$$y_i = -\tau_i \frac{dy_i}{dt} + \sum_{j=1}^{N_i} \omega_{ij} o_j \quad o_i = \text{sigmo}(y_i + \theta_i) \quad [2]$$

L'évolution calcule les poids, le biais et la constante de temps de chaque neurone.

- Enfin, les neurones du troisième contrôleur sont aussi des neurones « intégrateurs à fuite » mais chaque biais est déterminé par un mécanisme bio-inspiré (Mass *et al.*, 1998). Le mécanisme homéostatique proposé, appelé Center-Crossing (Boonyanit *et al.*, 2002), tend à réguler l'excitabilité du neurone autour d'un seuil d'activation idéal calculé sur les valeurs des synapses entrantes. Les équations du neurone sont les suivantes :

$$y_i = -\tau_i \frac{dy_i}{dt} + \sum_{j=1}^{N_i} \omega_{ij} o_j \quad o_i = \text{sigmo}(y_i + \theta_i) \quad \theta_j = -\frac{\sum \omega_{ij}}{2} \quad [3]$$

Seuls les poids et les constantes de temps de chaque neurone sont calculés par évolution. Le paramètre de biais θ est toujours constant mais n'est plus calculé par l'évolution puisqu'il dépend ici des poids synaptiques.

Pour ces trois modèles, la sortie interne y_i est l'état d'excitation interne du neurone et correspond en biologie au potentiel de membrane. La sortie externe o_i correspond à la fréquence moyenne à court terme des potentiels d'action (activité du neurone). La constante de temps τ présente dans le modèle dynamique est une propriété intrinsèque au neurone liée à sa bande passante. Enfin, le biais θ du neurone correspond au seuil d'activation de la membrane. ω_{ij} est l'efficacité synaptique de la connexion (synapse) du neurone j vers le neurone i , N_i est le nombre de synapses entrantes du neurone i .

2.2. Résultats comparatifs

Pour faire une étude plus exhaustive sur la robustesse de la méthode, nous avons effectué 10 évolutions pour chaque type de réseau (avec les paramètres du Tableau 1, le temps de calcul d'une évolution est d'environ 5 heures sur un Pentium 4 à 2,5 Ghz). Pour l'ensemble des expériences, l'évolution a permis de synthétiser des neuro-contrôleurs capables de calculer une commande qui déplace le robot, mais avec des performances différentes. Les résultats sont présentés en terme de commande, d'évaluation des performances et de diversité de solutions.

Les stratégies adoptées ne sont pas identiques selon le type de réseau synthétisé (**Figure 3** et **Figure 4**). Ainsi, celle des perceptrons récurrents standards (**Figure 3**

en haut) consiste à positionner le bras en appui sur le sol et à appliquer un mouvement de faible amplitude avec des consignes saturées à -1rad/s pour Q_{d1} et Q_{d2} et une oscillation rapide de Q_{d3} autour de $0,75\text{rad/s}$. Les vibrations obtenues permettent de déplacer le corps du robot, mais dans la plupart des cas, le robot avance sur une faible distance puis s'arrête.

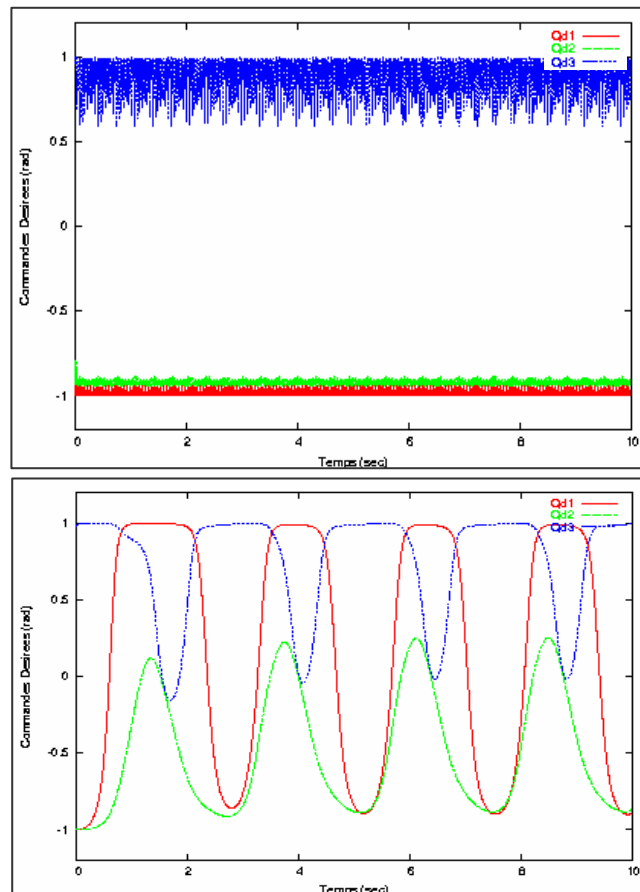


Figure 3. Comparaison des commandes articulaires calculées par le meilleur neuro-contrôleur d'une population. En haut, celles d'un perceptron récurrent standard, en bas, celles d'un perceptron récurrent avec neurones intégrateurs à fuite. Dans les deux cas, les biais sont calculés par l'évolution.

Lorsqu'on remplace les neurones classiques par des « neurones intégrateurs à fuite » la commande est différente (**Figure 3**, en bas) mais reste saturée. L'amplitude des mouvements est importante, le bras se soulève assez vite et régulièrement pour reprendre appui sur le sol. Le modèle de neurone comportant une

dynamique (constante de temps) optimisée par l'évolution influence donc directement la dynamique de la commande. Le troisième type de contrôleur, pour lequel le biais de chaque neurone est ajusté selon une heuristique bio inspirée, applique une commande oscillante de fréquence plus importante que le second contrôleur. Les oscillations ne sont pas saturées (**Figure 4**) et le régime transitoire qui précède le régime permanent tend à montrer la stabilité du contrôleur.

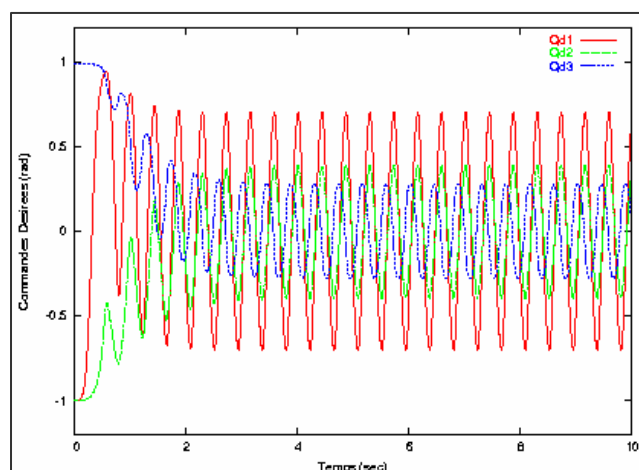


Figure 4. Commandes articulaires avec neurones intégrateurs à fuite et biais calculés par heuristique (meilleur contrôleur d'une population)

Les courbes de la **Figure 5** permettent de comparer la performance du meilleur des trois types de contrôleurs en terme de tâche, c'est à dire en mesurant l'éloignement par rapport au point de départ obtenu dans le temps fixé de 10 secondes. On aperçoit nettement l'efficacité des neurones intégrateurs sur les neurones standard et l'efficacité de l'heuristique dans le calcul des biais sur les neurones intégrateurs. On voit que le réseau classique ne génère pas de cycle locomoteur sur la commande, et le robot arrête de se déplacer vers la 9^{ème} seconde. En effet, les simulations montrent que la patte actionne le genou durant 8 secondes environ en effectuant une poussée sur le sol qui fait pivoter le corps sur l'arête opposée à l'épaule. Dans le cas du meilleur contrôleur cette poussée fait basculer le corps sur une autre face, la patte se retrouvant « en l'air », empêchant ainsi tout déplacement.

Le réseau de type intégrateur à fuite génère quant à lui un cycle locomoteur dont la fréquence est faible (4 enjambées pendant 10 s). Les simulations montrent que cycle se maintient à l'infini après les 10 secondes. Le troisième type de réseau génère un cycle de locomotion de fréquence plus élevée (20 enjambées environ) qui se maintient aussi à l'infini. Pour ce dernier type de réseau, on observe plus particulièrement, en début du cycle, un régime transitoire et une commande du robot

à une vitesse importante et stable. La distance parcourue est finalement presque 3 fois plus longue que pour les 2 premiers types. Les différentes simulations montrent qu'à l'issue de chaque évolution très peu d'individus parmi les 3 modèles de réseau déplacent le robot avec un mouvement de traction sur le sol (ce qui est évident car dans ce cas, la patte glisse), tandis que la majorité des contrôleurs font exercer à la patte un mouvement de poussée.

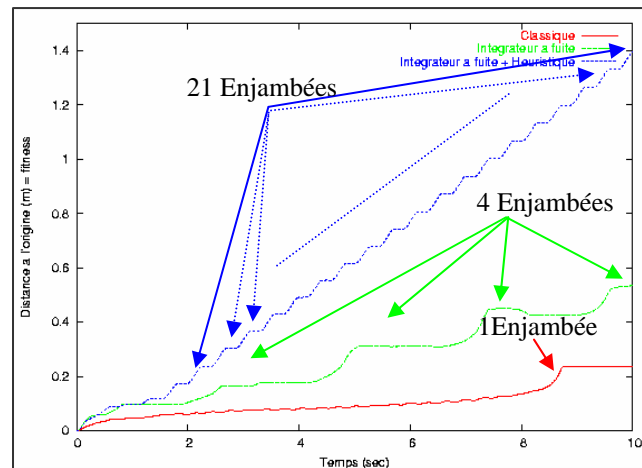


Figure 5. Fonction d'évaluation pour les 3 types de réseau (meilleur contrôleur d'une population). L'indice de performance est la distance parcourue par le robot en 10s. Une variation de distance observée pour le corps du robot sur chacune des 3 courbes correspond à une enjambée pour la patte. Chaque enjambée est visible par une variation rapide la distance parcourue.

La qualité des résultats obtenus peut être analysée à travers la diversité des solutions trouvées par les dix expériences d'évolution et la valeur de la fonction d'évaluation. Les graphes de la **Figure 6** montrent, au cours des 1000 générations, l'évolution de la valeur moyenne de la fonction d'évaluation (couramment nommée fitness) d'une population (200 neuro-contrôleurs) et celle du meilleur individu obtenu en fin d'évolution. L'écart type est calculé sur les 10 expériences. On constate nettement que les réseaux de neurones intégrateurs à fuite permettent d'obtenir une valeur de la fonction d'évaluation plus importante en fin d'évolution avec un faible écart type. De plus, le calcul des biais par une heuristique permet, non seulement d'augmenter fortement la valeur de la fonction d'évaluation en fin d'évolution, mais d'accélérer l'exploration avec une diversité qui décroît.

Ces résultats comparatifs montrent que l'heuristique apportée pour le calcul des biais accélère fortement l'évolution en la focalisant sur une zone de l'espace de recherche où les solutions sont déjà efficaces grâce à cette heuristique rendant ainsi robuste la population de solutions. On peut donc conclure qu'il est intéressant

d'incorporer dans le réseau des mécanismes biomimétiques de type homéostatique sous forme d'heuristiques plutôt que de laisser à l'évolution ce calcul.

De plus, nous avons montré comment un contrôleur neuronal peut commander ces robots avec de bonnes performances, tout en étant robuste et sans aucun modèle de ceux-ci, et comment il peut s'adapter à leur comportement dynamique.

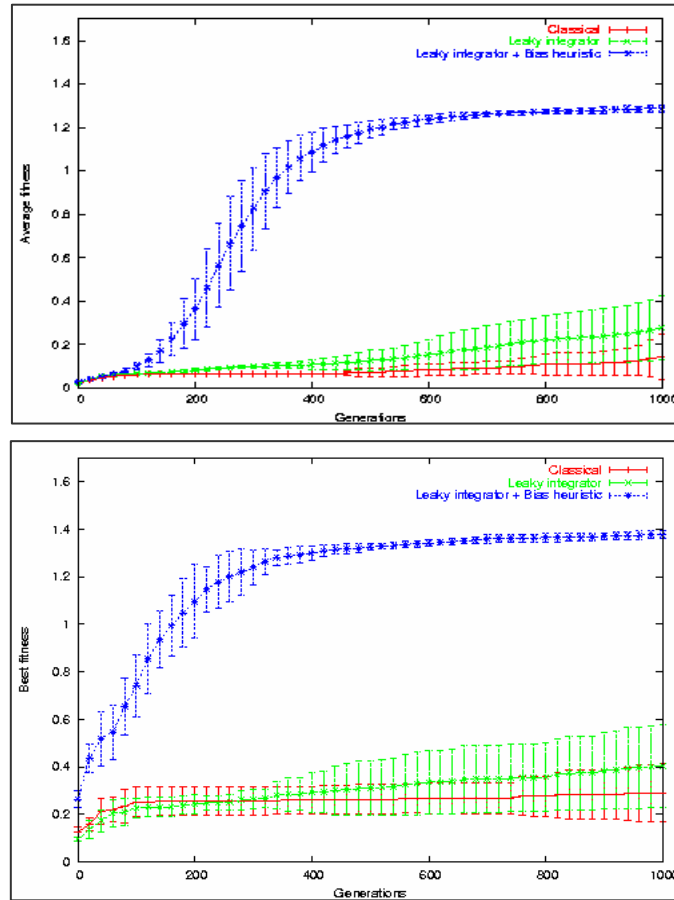


Figure 6. Comparaison de la fonction d'évaluation pour les trois types de réseaux. En haut : valeur moyenne sur la population, en bas : meilleur contrôleur de la population.

Ainsi nous pouvons dire que, puisque différents contrôleurs d'une même génération ont les mêmes performances, une adaptation en-ligne de leurs paramètres maintiendra cette performance face aux perturbations auxquelles sera soumis le robot. Le contrôleur deviendra alors adaptatif et non pas adapté comme ceux présentés jusqu'ici.

Le paragraphe suivant présente donc cette approche qui consiste à incorporer des mécanismes adaptatifs pour les neurones et les synapses.

3. Modèle bio-inspiré de contrôleur neuronal adaptatif

Dans notre démarche (pour plus détails voir (Hoinville *et al.*, 2004)), l'algorithme évolutionniste est chargé de faire évoluer, non pas la valeur des paramètres des neurones mais leur variation afin de doter le réseau de la propriété de « plasticité ». Ainsi, comme (Floreano *et al.*, 1999, 2000) on peut proposer différentes lois locales d'adaptation des poids inspirées de la biologie : règle de Hebb, règle post-synaptique, règle pré-synaptique, règle de covariance...

Nous proposons dans notre démarche d'incorporer en plus d'autres lois biologiques de type « homéostatique » (Turrigiano, 1999) qui vont tendre par synergie à stabiliser la fréquence d'activation du neurone dans un domaine efficace. Ainsi, un neurone « saturé » ou « muet » verra son excitabilité (son biais) diminuer ou augmenter vers un niveau d'activité raisonnable.

3.1. Mécanismes homéostatiques et synapses adaptatives

Le modèle de neurone « d'intégrateur à fuite » représenté plus haut par l'équation [4] peut être réécrit de la manière suivante :

$$\tau_i \frac{dy_i}{dt} = -y_i + \sum_{j=1}^{N_i} \omega_{ij} o_j - \theta_i + I_i \quad \text{avec} \quad o_j = \frac{1}{1 + e^{-\alpha_j y_j}} \quad (\text{sigmoïde}) \quad [4]$$

I_i désigne l'excitation externe provenant d'un capteur dans le cas d'un neurone sensoriel. α_i est un gain déterminant la pente de la fonction sigmoïde.

Deux modèles de mécanismes homéostatiques ont été envisagés. Le premier, présenté au paragraphe 2 et nommé « center-crossing », régule l'excitabilité membranaire des neurones en déterminant le seuil d'activation idéal d'un neurone en fonction des poids des synapses entrantes suivant une loi du type :

$$\theta_i = \frac{1}{2} \sum_{j=1}^{N_i} \omega_{ij} \quad [5]$$

Le second que nous proposons d'incorporer stabilise l'efficacité synaptique globale des connexions entrantes de chaque neurone par un procédé de normalisation (Gerstner *et al.*, 2002) du type :

$$\|\vec{\omega}_i\| = \sqrt{\sum_{j=1}^{N_i} \omega_{ij}^2} \quad [6]$$

Pour les synapses, c'est leur variation que l'évolution va adapter par l'intermédiaire des quatre lois locales et du taux d'apprentissage. C'est alors la « plasticité » du

réseau qui est optimisé par l'algorithme d'évolution artificielle. Floerano dans (Floerano *et al.*, 1999, Floerano *et al.*, 2000) a en effet montré que cette méthode de synapses adaptatives conduit à un comportement robuste et adaptatif plutôt que adapté.

Ces lois locales d'adaptation des poids, inspirées de la biologie sont les suivantes (λ est le pas d'apprentissage) :

- loi de Hebb simple :

$$\frac{d|\omega_{ij}|}{dt} = \lambda(1 - |\omega_{ij}|)o_j o_i \quad [7]$$

- loi post-synaptique :

$$\frac{d|\omega_{ij}|}{dt} = \lambda \left[(1 - |\omega_{ij}|)o_j o_i + |\omega_{ij}|o_i(o_j - 1) \right] \quad [8]$$

- loi pré-synaptique :

$$\frac{d|\omega_{ij}|}{dt} = \lambda \left[(1 - |\omega_{ij}|)o_j o_i + |\omega_{ij}|o_j(o_i - 1) \right] \quad [9]$$

- loi de covariance :

$$\frac{d|\omega_{ij}|}{dt} = \begin{cases} \lambda(1 - |\omega_{ij}|)\delta(o_j, o_i) & \text{si } \delta(o_j, o_i) > 0 \\ \lambda|\omega_{ij}|\delta(o_j, o_i) & \text{sinon} \end{cases} \quad [10]$$

avec $\delta(o_i, o_j)$: mesure de différence entre la sortie o_i du neurone i et celle o_j du neurone j .

Nous proposons ainsi un méta modèle capable d'instancier 4 sous modèles : le modèle « center-crossing » à synapses normalisées (CCNS), le modèle « center-crossing » (CC), le modèle à synapses normalisées (NS) et le modèle classique (CM). La topologie du réseau est libre, c'est à dire que seul le nombre de neurone est imposé ainsi que leur rôle (neurone sensoriel pour l'entrée, neurone moteur pour la sortie, et inter neurone). La synthèse évolutionniste a donc pour rôle de chercher pour chaque neurone le meilleur modèle, et d'associer à chaque synapse un type de mécanisme adaptatif parmi les quatre proposés ci-dessus. Ainsi, nous voulons montrer que l'incorporation de ces lois ou d'autres lois de ce type, contribue à une évolution plus aisée de comportements adaptatifs et que de l'apprentissage, au sens commun du terme émerge, des interactions locales entre les neurones.

3.2. Méthodologie d'évaluation des performances

3.2.1. Aspect robotique

Nous considérons le même robot qu'au chapitre 2. Il est composé d'une patte à 3 degrés de liberté poussant un corps. Les caractéristiques mécaniques et cinématiques de la patte sont identiques. Cependant nous lions le corps cubique à un rail prismatique (axe X, **Figure 7**) qui est fixe par rapport au sol à une hauteur de 12,5 cm. Le corps du robot peut glisser sur ce rail contre une force de frottement visqueux réglable. Ainsi, si l'on considère l'hexapode de la **Figure 1**, cette force de frottement qui s'oppose au déplacement du monopode peut être vue comme une perturbation extérieure provoquée par les mouvements des autres pattes de l'hexapode. Pour faire avancer le robot, le réseau locomoteur peut donc provoquer sur les pattes un mouvement de poussée sur le sol, ou bien des accélérations telles que, sans contact avec le sol, la dynamique du mouvement suffit au glissement du robot sur le rail.

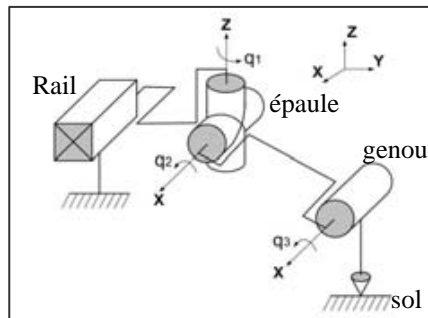


Figure 7. Morphologie et modèle cinématique du monopode glissant sur le rail.

La tâche du robot consiste à suivre une consigne de vitesse V_{des} quelle que soit la force de frottement. Sa performance p est calculée selon l'erreur entre la consigne en vitesse et la vitesse moyenne \tilde{V}_{eff} du corps du robot calculée sur la durée T de la simulation (10 secondes) :

$$p = \frac{1}{T} \int_0^T |V_{des} - \tilde{V}_{eff}| dt \quad [11]$$

La moyenne \tilde{V}_{eff} de V_{eff} est obtenue par un filtrage de type passe-bas du second ordre afin d'atténuer les brusques variations de vitesse apparaissant à chaque foulée.

3.2.2. Aspect évolutionniste

La synthèse du contrôleur est effectuée par l'algorithme évolutionniste qui choisit d'attacher à chaque synapse une loi plastique particulière. Durant l'évolution chaque contrôleur est évalué suivant 3 scénarii successifs qui durent chacun 10

secondes. Ces scenarii décrits par le **Tableau 2** sont définis par le coefficient de frottement k_{fr} du rail et par le profil temporel de la vitesse désirée V_{des} du robot. Pour les scenarii A et B, k_{fr} (i.e. la force de perturbation extérieure) est constant tandis que la vitesse désirée varie lentement suivant un profil en rampe (scénario A) ou varie brusquement suivant un profil en échelon (scénario B). Le scénario A correspond ainsi à un simple suivi de consigne et le scénario B récompense la capacité du contrôleur à stopper la marche. Le scénario C a pour but de favoriser l'adaptation du contrôleur aux perturbations. Pour ce faire, c'est la vitesse désirée qui est constante tandis que le coefficient de frottement (i.e. la force de perturbation extérieure appliquée au robot) varie brusquement suivant un profil en échelon qui double la valeur de k_{fr} au bout de 2,5 secondes.


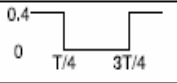

	Scenario A	Scenario B	Scenario C
$V_{des} [m.s^{-1}]$			0.3
$k_{fr} [N.s.m^{-1}]$	10	10	

Tableau 2. Scenarii définissant les épreuves imposées aux contrôleurs pendant l'évolution.

La fonction globale d'évaluation FE d'un individu est la somme quadratique des performances obtenues lors de chaque scénario, elle s'écrit :

$$FE = \sqrt{p_A^2 + p_B^2 + p_C^2} \tag{12}$$

Dans ce critère, p_A, p_B, p_C sont les performances obtenues suivant l'équation [11] respectivement pour les scenarii A, B, C. En combinant ces trois performances, ce critère atténue l'effet produit par la moyenne et en tenant compte des comportements différents qui seront obtenus par les trois scenarii. De plus, le processus de sélection des individus qui agit selon le rang obtenu par l'individu dans la population et non pas selon la valeur de sa performance permet de compenser la non quadraticité du critère.

Les contrôleurs sont génétiquement codés en respectant les principes habituels du domaine. Ainsi, des valeurs numériques listent les paramètres intrinsèques de chaque neurone et des codes symboliques listent les propriétés des synapses notamment l'une des 4 lois d'adaptation locales présentées plus haut. Tous les paramètres sont initialisés aléatoirement. Ces principes qui ne sont pas détaillés ici pour une meilleure compréhension, sont résumés par le schéma de la **Figure 8** qui montre la structure du génotype du neurone. Sur cette figure, on voit par exemple les paramètres initiaux d'un neurone : constante de temps, gain, biais, et les synapses

qui lui sont liées. La première de ces synapses est active et inhibitrice, sa plasticité suit la loi de covariance, et la constante de temps de cette loi vaut 7,55.

Dans cette étude, l'algorithme d'évolution artificielle a pour rôle d'optimiser les lois d'adaptation des synapses et non le nombre de neurones qui est fixé arbitrairement dès la première génération et reste donc constant durant l'évolution. Le nombre de neurones en entrée et en sortie dépendant de la tâche, deux neurones sensoriels sont fixés en entrée (l'un reçoit l'erreur en vitesse, l'autre le contact binaire sol/pied), tandis qu'en sortie, trois neurones calculent la consigne angulaire de chaque articulation.

Après quelques essais empiriques, la couche cachée est fixée à 3 inter-neurones ce qui donne une taille totale de 8 neurones pour le contrôleur de pattes.

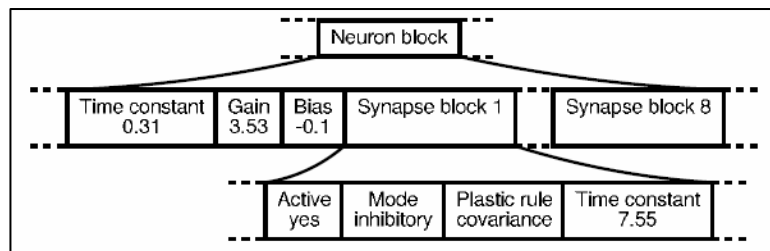


Figure 8. Structure du génotype du neurone

Le nombre de synapses est initialisé sur la base d'un réseau totalement connecté, c'est-à-dire que chaque neurone est connecté avec tous les autres y compris lui-même. Cependant, l'état initial de chaque synapse et sa loi d'adaptation sont choisis aléatoirement. Ainsi un état actif signifie que la synapse existe, tandis qu'un état inactif signifie qu'elle n'existe pas (voir exemple de la **Figure 8**). Au cours de l'évolution la topologie synaptique du réseau peut ainsi être modifiée.

3.3. Résultats

3.3.1. Aspects évolutionnistes et neuronaux

Pour chaque modèle de neurones, nous avons effectué sur 2000 générations de contrôleurs, 10 évolutions différentes avec une population initiale aléatoire de 200 contrôleurs. La **Figure 9** montre les résultats comparatifs pour les quatre modèles CM, NS, CC et CCNS.

On constate tout d'abord que, comme au **paragraphe 2.2**, le modèle CCNS qui incorpore les 2 mécanismes adaptatifs est le plus performant. Cependant, pour les modèles CM, NS et CC, l'écart type est faible au début de l'évolution et il augmente ensuite. Ceci provient des faibles performances, qui sont donc toutes du même ordre de grandeur, obtenues par les premières générations de ces contrôleurs dont très peu d'individus sont capables d'exhiber un rythme locomoteur comme cela a déjà été constaté au **paragraphe 2.2**. Pour ces modèles, l'écart type augmente au cours des

générations car la diversité des populations s'accroît du fait de l'apparition de contrôleurs plus performant dont le nombre se stabilise vers les dernières générations pour lesquelles toutes les performances existent. Pour le modèle CCNS, l'écart type reste quasiment constant au cours de l'évolution. Ceci est du à la capacité de ce modèle à exhiber un rythme locomoteur dès la première génération quelque soit ses paramètres initiaux. On retrouve là aussi les résultats du **paragraphe 2.2**.

Afin de vérifier la pertinence de cette analyse nous avons évalué, pour chaque modèle et chaque scénario, la performance moyenne des 10 meilleurs contrôleurs obtenus par les 10 évolutions. Pour ce faire, chacun d'entre eux est soumis après l'évolution aux quatre scenarii mais durant 100 secondes chacun.

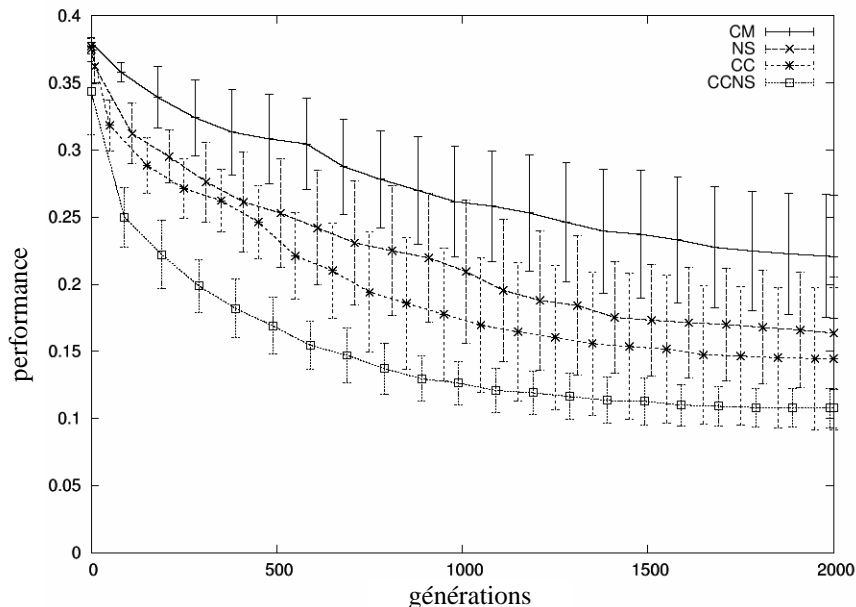


Figure 9. Meilleure performance obtenue à chaque génération pour les 4 modèles de neurone proposés : moyenne et écart type sur 10 évolutions.

Sur la **Figure 10**, qui montre les statistiques obtenues, on constate que les mécanismes homéostatiques des modèles CC, NS et CCNS accroissent la performance quelque soit le scénario. Les performances des modèles CC et NS sont presque semblables même si celle du modèle CC est inférieure. Le modèle CCNS, qui met en oeuvre une synergie entre les 2 mécanismes, est le plus performant quelque soit le scénario. Les pourcentages montrent de plus que ce modèle est moins sensible à la difficulté de l'épreuve contrairement aux autres modèles, grâce à la synergie apportée par les deux mécanismes homéostatique.

La **Figure 11** représente un exemple de meilleur contrôleur obtenu pour le type CCNS et détaille les caractéristiques de ses synapses déterminées par l'évolution.

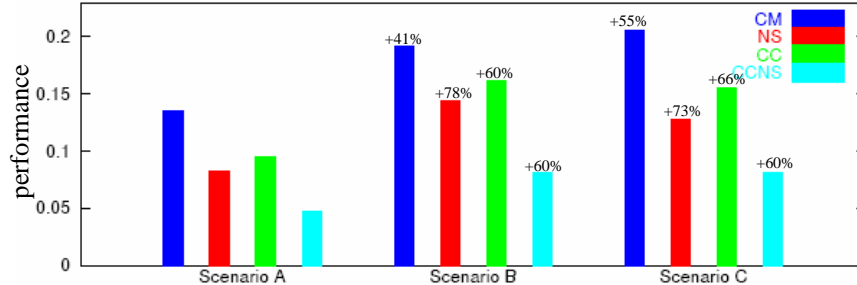


Figure 10. Moyenne des performances (sur les 10 meilleurs contrôleurs obtenus par les 10 évolutions et pour chaque type de contrôleur) obtenues lors de chaque scénario dont la durée est étendue à 100 secondes. Les pourcentages correspondent aux variations des performances obtenues au cours du scénario par rapport à celles obtenues lors du scénario A.

On constate dans cet exemple que la connexion finale n'est pas totale, que les 4 lois d'adaptation existent et qu'il y a autant de synapses excitatrices (9) que de synapses inhibitrices (8).

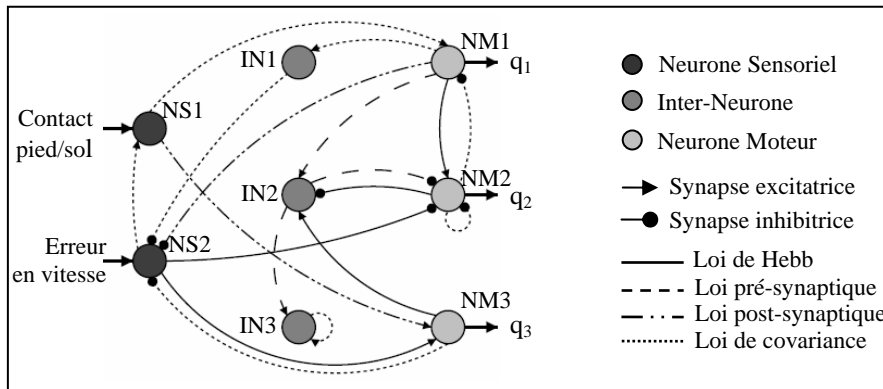


Figure 11. Exemple de structure de neuro-contrôleur obtenu par évolution.

Une analyse quantitative fine sur le fonctionnement des contrôleurs nécessiterait de se doter d'outils d'investigation puissant que notre plate-forme ne possède par encore. Une analyse qualitative effectuée sur cet exemple de contrôleur CCNS ainsi que sur d'autres obtenus nous permet cependant d'avancer quelques hypothèses. En effet, on s'aperçoit que les neurones sensoriels NS1 et NS2 excitent les neurones moteurs NM1 et NM3, tandis que NM1 excite NM2 et NS2 excite NS1. On peut penser que cette sous-structure est à la base de la production de mouvement rythmique à partir des informations sensorielles. En effet, les synapses excitatrices de NS2 (erreur en vitesse) et NS1 (contact pied/sol) vont vers NM3 et NM2 qui

commandent respectivement l'articulation q_3 et q_2 responsables de la longueur d'enjambée, donc de la vitesse.

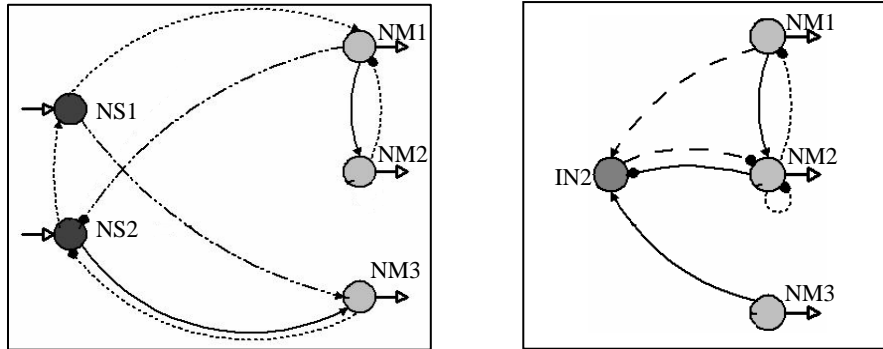


Figure 12. Mise en évidence des sous structures identifiées dans le neuro-contrôleur de la **Figure 11**. A gauche une structure en boucle fermée régulatrice de la vitesse, à droite une structure coordinatrice du mouvement de la patte.

La présence de connexions inhibitrices allant de NM3 et NM1 vers NS2 confirme cette hypothèse puisque apparaît alors une structure en contre-réaction qui semble réguler le rythme en fonction de l'erreur en vitesse comme le montre la **Figure 12** à gauche. L'inter-neurone IN2 n'est pas directement connecté aux neurones sensoriels mais est excité par les neurones moteurs NM1 et NM3, tandis qu'il est lié à NM2 par une boucle inhibitrice. Cette sous-structure (**Figure 12**, à droite) semble être responsable de la coordination des articulations pour produire une marche efficace car NM2 commande l'articulation q_2 du genou qui influence directement le contact patte/sol. Le neurone IN3 quant à lui est seul car il n'excite ni inhibe aucun autre. Il semble être inutile pour la génération de mouvement mais a du servir à l'algorithme évolutionniste comme degré de liberté supplémentaire pour l'exploration de l'espace des solutions.

3.3.2. Aspect robotique : robustesse du neuro-contrôleur CCNS

Après leur synthèse par l'évolution, les performances des meilleurs contrôleurs CCNS ont été étudiées. A cette fin, différents profils de vitesse désirée sont imposés au robot. La **Figure 13** (en haut) présente, durant 60 secondes, des résultats pour une vitesse de consigne sous forme d'échelons successifs appliqués sur une longue durée de 1000 secondes afin d'éprouver la stabilité. Le graphe du bas de la **Figure 13** présente, de 10 à 70 secondes, les mêmes résultats pour une vitesse de consigne sinusoïdale appliquée sur la même durée. La vitesse du corps du robot, les valeurs des consignes articulaires calculées par le réseau de neurones et l'allure du coefficient de frottement sont tracées sur ces deux figures. L'évolution temporelle de la vitesse du corps du robot nous indique clairement la stratégie de contrôle adoptée par l'évolution pour permettre au réseau locomoteur de suivre la consigne en vitesse. En effet, les courbes du haut de la **Figure 13** montrent qu'à une vitesse désirée positive correspond une phase active de propulsion composée d'une succession de « poussées » de l'extrémité de la patte sur le sol.

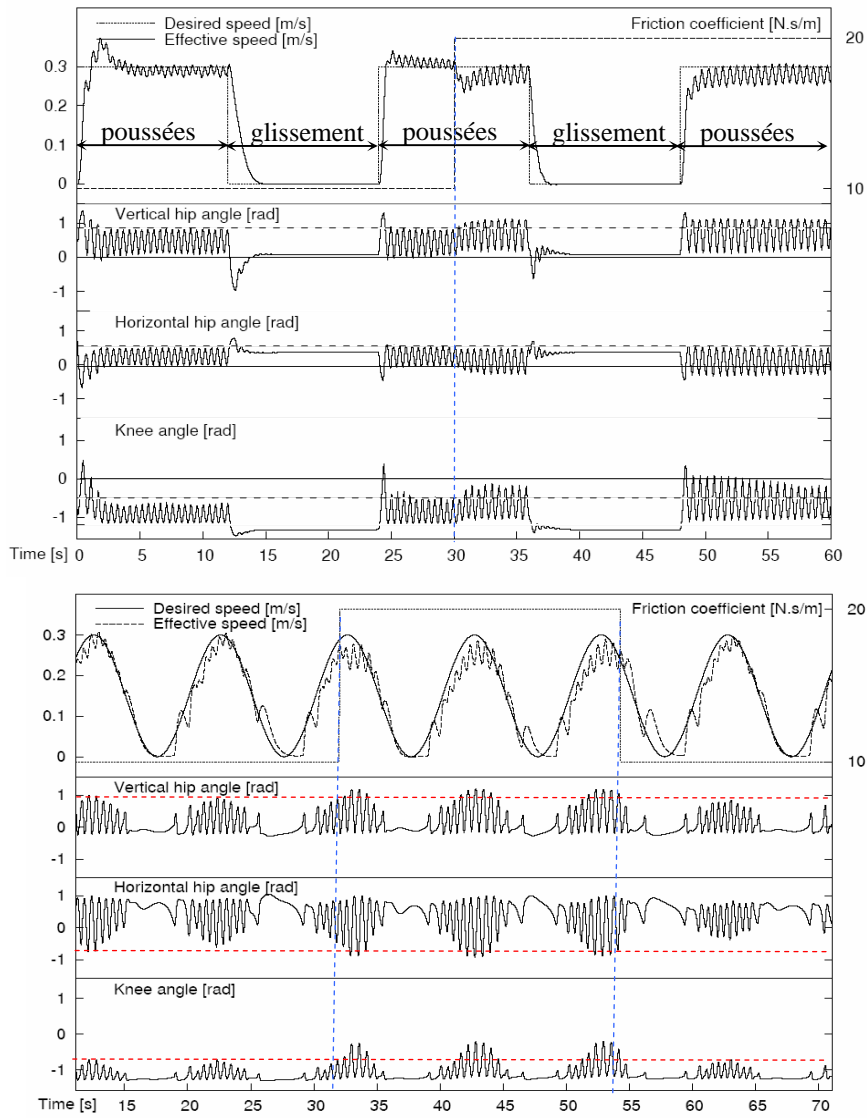


Figure 13. Graphe des consignes et des commandes (durant 60sec en haut et de 10s à 70sec en bas) obtenues par un contrôleur CCNS. En haut, le robot est perturbé à $t=30\text{sec}$ par un doublement du coefficient de frottement sur le rail. En bas, il est ralenti à $t=32\text{sec}$ par un doublement du coefficient, puis à $t=54\text{sec}$ par un retour de ce coefficient à sa valeur initiale. L'échelle du coefficient de frottement est affichée en haut à droite, celle de la vitesse en haut à gauche, et celle des variables articulaires à gauche. Chaque oscillation observée correspond à une poussée sur le sol

Chaque poussée est identifiable sur la figure par la présence d'une petite oscillation de la vitesse du robot autour de la vitesse de consigne et par les oscillations visibles des consignes articulaires. Lorsque la consigne en vitesse devient nulle, les oscillations s'arrêtent. La patte reste alors immobile au dessus du sol, ce qui signifie que le robot glisse sur sa lancée, sa vitesse décroissant vers zéro à cause des frottements visqueux contre le rail. Lorsque la consigne en vitesse est sinusoïdale, le même effet d'utilisation de l'énergie cinétique par le contrôleur est visible. Ce phénomène apparaît sur le bas de la figure par un nombre d'oscillations plus faible lorsque la consigne décroît que lorsqu'elle croît.

En effet, après application de la perturbation qui ralentit le robot (à $t=30s$ pour la consigne en échelon, et $t=32s$ pour la consigne sinusoïdale) ou qui l'accélère comme à $t=54s$ en bas de la **Figure 13**, l'amplitude des commandes est modifiée particulièrement sur le genou (les traits pointillés horizontaux marquent sur les deux figures les valeurs maximales des consignes articulaires avant perturbation) et la vitesse effective rejoint la vitesse de consigne. Il y a donc compensation de la perturbation par l'activité synaptique du contrôleur. L'effet des mécanismes adaptatifs homéostatiques est visible sur les deux figures. Cette adaptation des commandes calculées par le réseau de neurones est aussi visible pour le cycle de locomotion (**Figure 14**).

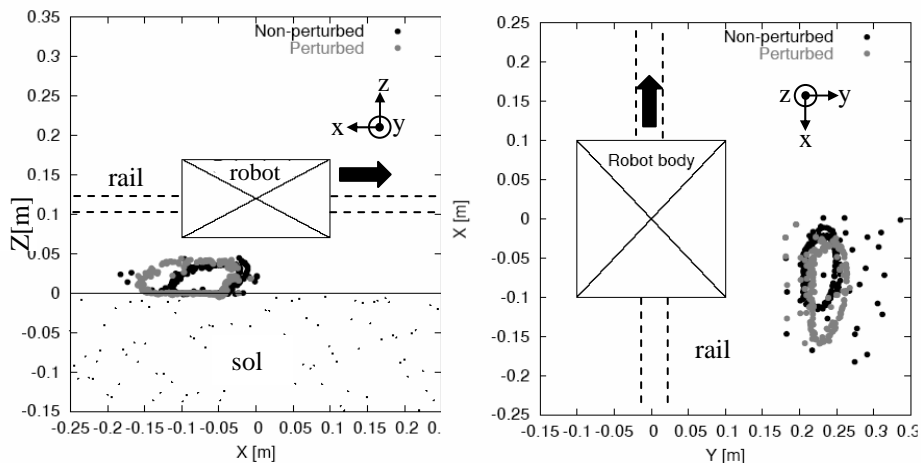


Figure 14. Trajectoires de l'extrémité de la patte avec perturbation (gris) et sans perturbation (noir) : vue latérale à gauche, vue de dessus à droite. La flèche noire indique la direction d'avance du robot et les repères sont ceux des **Figure 2** et **Figure 7**.

On constate que pour compenser la perte de vitesse due au frottement, le contrôleur génère une commande telle que l'amplitude de l'enjambée est augmentée et que le point d'appui de la patte sur le sol est reculé par rapport au corps pour augmenter l'effort de poussée afin de maintenir la vitesse. D'autre part, la vue du dessus (à droite dans la **Figure 14**) fait apparaître que ces points d'appuis sur le sol

sont distribués sur une surface assez large. Le mouvement de la patte n'est donc pas identique à chaque cycle. Ceci signifie que le contrôleur reste robuste par rapport à la configuration de la patte. Enfin, la surface définie par l'ensemble de ces points de contact diminue lorsque la perturbation est appliquée, ce qui montre que l'efficacité du contrôleur s'accroît lorsqu'il y a perturbation.

4. Conclusion

Nous avons montré dans cet article l'importance du modèle de neurone pour générer avec de bonnes performances des cycles de locomotion pour des robots à pattes sans aucun modèle de ceux-ci. Nous avons exposé d'autre part l'intérêt qu'il peut y avoir à s'inspirer de la nature pour rendre plus robustes ces contrôleurs en incorporant des mécanismes biologiques adaptatifs dans les modèles de neurones.

Les études statistiques que nous avons menées sur les contrôleurs synthétisés par évolution ont permis de valider nos hypothèses sur une grande population de solutions. Quel que soit sa performance, il apparaît que le contrôleur devient plus efficace et plus stable lorsqu'on lui incorpore de tels mécanismes homéostatiques car ceux-ci permettent une régulation interne de l'activité des neurones évitant ainsi des comportements instables du contrôleur. Nous avons montré ainsi que la plasticité synaptique des contrôleurs neuronaux est une propriété importante qui permet d'accroître l'autonomie des robots réels face aux perturbations qu'ils peuvent subir.

Nous avons utilisé la synthèse évolutionniste car c'est un outil intéressant qui permet un choix optimal de ces mécanismes et des lois synaptiques puisqu'elle évalue systématiquement par la simulation chaque contrôleur créé sur un groupe d'épreuves imposées au robot.

Cette démarche doit être poursuivie sur un robot simulé multi-pattes afin de tester l'efficacité des solutions pour des dysfonctionnements importants telle que la perte de patte et les changements morphologiques du robot.

4. Bibliographie

- Baker J. E., « Reducing bias and inefficiency in the selection algorithm », *Proceedings of the second International Conference Genetic Algorithms Applications*, Lawrence Erlbaum Associates, Hillsdale, 1987, p. 14-21
- Beer R.D., « On the dynamics of small continuous-time recurrent neural networks », *Adaptive Behavior*, 3(4), 1995, p. 469-509.
- Boonyanit M., Beer R.D., « Center-crossing recurrent neural networks for the evolution of rhythmic behavior », *Neural computation*, Vol 14, 2002, p. 2043-2051
- Cruse H., Kindermann T., Schumm M., Dean J., Schmitz J., « Walknet, a biologically inspired network to control six-legged walking », *Neural Networks*, Vol 11, R. Brooks, S. Grossberg Ed., 1998, p.1435-1447

- Doncieux S., Evolution de contrôleurs neuronaux pour animats volants : méthodologie et applications, Thèse de doctorat, Université Paris 6, 2003
- Floreano D., Urzelai J., « Evolution of neural controllers with adaptive synapses and compact genetic encoding », *Proceedings of the 5th European Conference on Artificial Life*, Lecture notes In Computer Science, Vol. 1974, 1999, p.183-194
- Floreano D., Urzelai J., « Evolutionary robots: the next generation », *Proceedings of the Seventh International Symposium on Evolutionary Robotics: From intelligent Robots to Artificial Life*, AAI Book, T. Gomi Ed. , 2000
- Gallagher J.C., Beer R.D, Espenschied K.S., Quinn R.D., « Application of evolved locomotion controllers to a hexapod robot », *Robotics and Autonomous Systems*, Vol 19, 1996, p. 95–103.
- Gerstner W., Kistler W.M., *Spiking Neuron Models. Single Neurons, Populations, Plasticity*, Cambridge University Press, (<http://diwww.epfl.ch/~gerstner/BUCH.html>), chapter 11, 2002.
- Hoinville T. Hénaff P., « Comparative Study of Two Homeostatic Mechanisms in Evolved Neural Controllers for Legged Locomotion », *Proceedings of 2004 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2004, p. 2624-2629.
- Ijspeert, A. J., Hallam J., Willshaw D., « From lampreys to salamanders: Evolving neural controllers for swimming and walking ». In *Fifth Intl Conf on Simulation of Adaptive Behavior*, 1998, p.390-399.
- Ijspeert A.J., « A connectionist central pattern generator for the aquatic and terrestrial gaits of a simulated salamander ». *Biological Cybernetics*, 84(5), 2001, p.331-348.
- Maas W., Bishop C. M., *Pulsed neural Networks*, MIT Press,1998
- Meeden L. A., McGraw, G., Blank, D., « Emergence of control and planning in an autonomous vehicle », *Proceedings of the Fifteenth Annual Meeting of the Cognitive Science Society*, Edition Lawrence Erlbaum, Hillsdale, NJ, 1993, p. 735-740
- Meyer J.-A., Doncieux S., Filliat D., Guillot A., « Biologically inspired robot behavior engineering », In *R.J. D. J. Santos, and M. Grana, editors, Evolutionary Approaches to Neural Control of Rolling, Walking, Swimming and Flying Animats or Robots*. Springer-Verlag, 2002, p. 1-43.
- Nolfi S., « Evolving non-trivial behaviors on real robots: a garbage collecting robot », *Robotics and Autonomous System*, Vol 22, 1997, p.187–198.
- Rucci M., Edelman G. M., Wray J, « Adaptation of orienting behavior: from the barn owl to a robotic system », *IEEE trans.s on Robotics and Automation*, Vol 15(1) , 1999, p. 96-110
- Shinkichi I., Hideo Y., Takanori S., Tamio A. « Wave CPG model for autonomous decentralized multi-legged robot: Gait generation and walking speed control», *Robotic and Autonomous Systems* 54, 2006, p.118-126
- Turriano G.G. « Homeostatic plasticity in neural networks: the more things change, the more stay the same », *Trend in Neurosciences*, Vol 22, 1999, p. 221-227.

Von Twinkel A., Paseman F. «Reflex-oscillations in evolved single leg neurocontrollers for walking machines», à paraître dans *Natural Computing*, 2006.

Ziemke T, « On "Parts" and "Wholes" of Adaptative Behavior: Functional Modularity and Diachronic Structure in Recurrent Neural Robot Controller », *Proceedings of the sixth international conference on simulation of adaptive behavior. From animals to animats*, Massachusetts Institute of Technology, Cambridge, 2000, pp. 115-124