



HAL
open science

Commande bio-inspirée et genèse de mouvements rythmiques en robotique

Patrick Henaff

► **To cite this version:**

Patrick Henaff. Commande bio-inspirée et genèse de mouvements rythmiques en robotique. Automatique / Robotique. Université de Cergy Pontoise, 2011. tel-00667651

HAL Id: tel-00667651

<https://theses.hal.science/tel-00667651>

Submitted on 8 Feb 2012

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.



HABILITATION À DIRIGER DES RECHERCHES

Présentée par

Monsieur PATRICK HÉNAFF

Discipline :

Sciences de l'Ingénieur

COMMANDE BIO-INSPIRÉE

ET

GENÈSE DE MOUVEMENTS RYTHMIQUES EN ROBOTIQUE

Soutenue le 5 Juillet 2011, devant le jury composé de :

Mr. Auke Jan Ijspeert	Professeur à l'Ecole Polytechnique Fédérale de Lausanne	Rapporteur
Mr. Philippe Poignet	Professeur à l'Université de Montpellier	Rapporteur
Mr. Olivier Sigaud	Professeur à l'Université Pierre et Marie Curie	Rapporteur
Mr. Raja Chatila	Directeur de Recherche au LAAS, CNRS, Toulouse	Examinateur
Mr. Daniel Cattaert	Directeur de Recherche à l'INRIA, CNRS, Bordeaux	Examinateur
Mr. Fethi Ben Oueddou	Professeur à l'Université de Versailles S ^t Quentin	Examinateur
Mr. Philippe Gaussier	Professeur à l'Université de Cergy-Pontoise	Examinateur

Laboratoire des Equipes Traitement de l'Information et Systèmes (ETIS)
UCP - CNRS -ENSEA- UMR 8051
Cergy-Pontoise
France

Remerciements

Je remercie tous les collègues du LISV, chercheurs, techniciens et administratifs, avec qui j'ai vécu, travaillé et échangé durant ces années, et aussi ceux d'ETIS qui m'ont laissé assez tranquille depuis 2 ans pour terminer ce travail. Un remerciement particulier à Philippe Gaussier pour sa richesse d'esprit, sa générosité, et pour sa « perception de la robotique » (sans jeu de mot) qu'il représente et qui nous ouvre de nouvelles voies.

C'est un honneur pour moi qu'Auke Jan Ijspeert ait accepté de rapporter sur mon travail malgré le peu de temps qu'il avait de disponible, je le remercie vivement.

J'exprime mes remerciements sincères à Olivier Sigaud pour son aide précieuse (sur le fond et la forme) et pour ses critiques constructives qui m'ont aidé dans ma réflexion lors de la dernière ligne droite.

Merci à Philippe Poinet qui m'a conseillé depuis quelques années sur les bonnes étapes à suivre pour aller vers l'HDR.

Merci aussi à Daniel Cattaert qui a toujours été disponible pour nous aider à progresser sur nos modèles neuronaux et qui a sans cesse émis des avis positifs et admirateur sur nos « machines marcheuses » qui sont pourtant encore très loin d'atteindre la perfection qu'il observe tous les jours dans la nature.

Merci à Raja Chatila qui m'a toujours impressionné par sa capacité d'analyse et sa vision très en avance de la robotique.

Je dédie ce document d'HDR aux doctorants et étudiants de Master (Roland, Gildas, Vincent, Thierry, Hayssam, John, Elmira, Slava, Artem et les autres...) que j'ai encadrés et sans qui ce travail n'aurait pas pu être mené avec autant de richesses et d'idées. Les échanges et débats quelquefois longs, contradictoires, voire même passionnés ou remplis d'émotions sont souvent nécessaires à la progression scientifique, à la création et à l'aboutissement d'une thèse.

Enfin, merci aussi et surtout à ma famille qui me supporte depuis quelques mois...

Messages Personnels

*Aucun Boeing sur mon transit,
Aucun bateau sur mon transat,
Je cherche en vain la porte exacte.
Je cherche en vain le mot exit.*

*Je chante pour les transistors,
Ce récit de l'étrange histoire,
De tes anamours transitoires,
De Belle au Bois Dormant qui dort.*

*Je t'aime et je crains
De m'égarer.
Et je sème des grains
De pavot sur les pavés
De l'anamour.*

*Tu sais ces photos de l'Asie,
Que j'ai prises à deux cents Asa,
Maintenant que tu n'es pas là,
Leurs couleurs vives ont pâli.*

*J'ai cru entendre les hélices
D'un quadrimoteur, mais hélas
C'est un ventilateur qui passe
Au ciel du poste de police.*

*Je t'aime et je crains
De m'égarer,
Et je sème des grains
De pavot sur les pavés
De l'anamour.*

*Je t'aime et je crains
De m'égarer,
Et je sème des grains
De pavot sur les pavés
De l'anamour...*

L'anamour, Serge Gainsbourg, 1969

Table des matières

I	Curriculum Vitæ	15
1	Curriculum Vitæ	17
1.1	Position personnelle et professionnelle	17
1.2	Titres Universitaires	17
1.3	Parcours académique	18
1.4	Rayonnement scientifique national et international	18
1.5	Expérience en entreprise	19
1.6	Synthèse de l'activité de recherche	20
1.6.1	Principales Responsabilités Scientifiques	20
1.6.2	Activités Internationales	21
1.6.3	Encadrement doctoral	21
1.6.4	Encadrement de Master, DEA et ingénieurs	23
1.6.5	Publications en tant que Maître de Conférences	24
1.6.6	Publications avant le statut de Maître de Conférences	27
2	Synthèse de l'activité d'Enseignement	29
2.1	Enseignement niveau Licence 1	29
2.2	Enseignement niveau Licence 2	30
2.3	Enseignement niveau Master 1	31
2.4	Enseignement niveau Master 2	32
2.5	Synthèse des services depuis 2000 :	32
2.6	Enseignements en tant qu'enseignant contractuel (1990-1997)	32
2.7	Responsabilités administratives	33
II	Activité scientifique	35
3	Introduction	37
3.1	Résumé du parcours scientifique	37
3.2	apprentissage en ligne d'un robot non-holonyme	38
3.2.1	Exemple de résultats expérimentaux	39
3.2.2	Conclusion	42
3.3	Pourquoi et comment s'inspirer de la nature?	42
3.4	La complexité des robots marcheurs	43
3.5	Objectif scientifique	45
4	Contrôle moteur : aspects biologiques et computationnels	49
4.1	Principes généraux de la neurophysiologie du geste moteur	49
4.1.1	Structure et complexité du Système Nerveux Central	49
4.1.2	Structure du Système Nerveux Moteur	50
4.1.3	Propriété de Plasticité	52
4.1.4	Mouvements réflexes et mouvements rythmiques.	53
4.1.5	Architecture sensori-motrices pour la locomotion	53
4.2	Modèles computationnels.	54
4.2.1	Modèles de neurone	55

4.2.2	Modèles d'architectures	57
4.2.3	Modèles de mécanisme adaptatif : plasticité	57
4.3	Modèles de neurocontrôleurs pour la locomotion	60
4.3.1	Modèle de CPG pour la nage de la lamproie	60
4.3.2	CPG pour le contrôle de la marche et de la nage	61
4.3.3	Modèles de CPG pour la marche bipède :	63
4.4	Conclusion	65
5	Apprentissage d'équilibre et de compliance en bipédie	67
5.1	BPTT et CTRNN : expérimentations	67
5.1.1	Apprentissage en ligne de l'équilibre pour un robot bipède en posture debout	68
5.1.2	Etude des performances temps-réel de la BPTT et améliorations	69
5.1.3	Influence de la dynamique neuronale sur la stabilité de l'apprentissage	71
5.1.4	Conclusion	72
5.2	La compliance articulaire améliore l'équilibre	73
5.3	La dynamique neuronale influence l'équilibre	76
5.4	Conclusion	78
6	Genèse de rythmes locomoteurs	79
6.1	Synthèse manuelle de contrôleurs rythmiques	79
6.1.1	Modélisation des blocs fonctionnels neuronaux :	80
6.1.2	Modèle de mécanismes adaptatifs	81
6.2	Contrôleurs neuronaux plastiques	83
6.2.1	Mise en évidence de l'importance de la dynamique neuronale sur l'efficacité du rythme locomoteur	83
6.2.2	Genèse de mouvements rythmiques stables pour la locomotion monopode	86
6.2.3	Locomotion adaptative pour les robots myriapodes.	90
6.3	Conclusion	93
7	Contrôleurs bio-inspiré pour les robots bipèdes	95
7.1	Modèle d'architecture sensori-motrice	96
7.1.1	Importance des retours sensoriels sur l'efficacité de la marche	96
7.1.2	Architecture neuronale complète pour la marche bipède	97
7.2	Conclusion	100
8	Conclusion et perspectives de recherche	101
8.1	Apprentissage et autonomie	103
8.2	Apprentissage moteur et compliance articulaire	103
8.3	Apprentissage et rythmicité	104
	Bibliographie	105
III	Annexe	113
9	Sélection d'articles	115
9.1	Control Engineering and Practice, 2010	115
9.2	Conférence Internationale IEEE IJCNN 2006	127
9.3	International Journal of Neural Systems, 2010	136
9.4	Conférence Internationale IEEE/RSJ IROS 2008	150
9.5	Conférence Internationale IEEE Humanoids 2008	159
9.6	Adaptive Behavior, 2011	166
9.7	Frontiers in Neurorobotics, 2010	189
9.8	Conférence Internationale IEEE/RSJ IROS 2009	199
9.9	Conférence Internationale SAB 2010	206

Table des figures

3.1	Schémas d'apprentissage en-ligne pour la commande du robot mobile. A gauche le réflexe d'évitement d'obstacle n'est pas à apprendre et l'entrée est constituée des erreurs cartésiennes et angulaires normalisées, la sortie des consignes en vitesses à appliquer aux moteurs. A droite le réflexe d'évitement d'obstacle est à apprendre et l'entrée est constituée en plus des retours sensoriels fournis par les capteurs à ultrasons.	39
3.2	Evolution du critère et des trajectoires du robot mobile (représenté à l'échelle en violet) pendant les deux premiers apprentissages (Henaff et Delaplace, 1996). . . .	39
3.3	Exemple de contrôle obtenu après apprentissage (le réseau de neurones commande le robot mais n'apprend plus), (Henaff et Delaplace, 1996).	40
3.4	Evolution du critère et des trajectoires du robot mobile pendant les deux premiers apprentissages sans présence d'obstacles. La trajectoire « hasardeuse » du robot provient de la phase d'exploration du réseau de neurones qui « découvre » peu à peu le comportement cinématique et dynamique du robot.(Bourdon et Henaff, 1997) . .	40
3.5	Apprentissage du réflexe d'évitement d'obstacles par interaction sensorielle. L'obstacle est situé sur le but. le robot s'arrête à la distance de sécurité $d_s = 50cm$. Quand l'obstacle se déplace, le contrôleur neuronal maintient la contrainte entre rejoindre le but et rester à une distance de 50cm de l'obstacle, (Bourdon et Henaff, 1997). . .	41
3.6	Apprentissage par interaction sensorielle avec le Robuter en sollicitant ses capteurs à ultrasons (Bourdon et Henaff, 1997).	41
3.7	Mise en évidence du réflexe d'évitement d'obstacle après l'apprentissage par interaction. Le robot s'arrête à la distance $d_s = 50cm$ de l'obstacle entre $t = 4s$ et $t = 11s$, et rejoint le but lorsque l'obstacle disparaît à $t = 11s$. (Bourdon et Henaff, 1997). . .	41
3.8	A gauche, exemple de déficiences subies par un robot marcheur hexapode : amputation de pattes. Comment pouvons-nous synthétiser un contrôleur capable de s'adapter à une telle situation? A droite : quels sont les modèles de contrôleurs bio-inspirés qui permettent un couplage dynamique stable entre la rythmicité des membres inférieurs et le mouvement du corps supérieur lorsque le robot est en interaction physique avec le sol ou avec l'être humain?	46
4.1	Grandes divisions du Système Nerveux (Extrait : Univ du Québec-etsmtl (cours)). Le SNP est anatomiquement mais pas fonctionnellement distinct du SNC. Sa partie autonome assure l'essentiel des fonctions végétatives, sa partie somatique assure la motricité.	50
4.2	Synthèse de l'architecture du Système Nerveux Moteur	51
4.3	Système nerveux moteur (à gauche) et fuseau neuromusculaire (à droite) (Extrait : cours du collèe de France, L. Damn).	51
4.4	Fonctionnement de la jonction neuro-musculaire. A gauche l'unité motrice, à droite un zoom de la jonction neuromusculaire sur la fibre musculaire.	52
4.5	Schéma général de la boucle sensori-locomotrice du membre arrière du chat (extrait de (Frigon et Rossignol, 2006)) et de l'écrevisse dans la phase de saut (extrait de (Cattaert et LeRay, 2001)).	54

4.6	Propriétés intrinsèques des neurones rythmiques biologiques en haut (figure extraite de (Marder et Bucher, 2001)), en en bas simulation correspondante (extrait de (Hoinville, 2007)).	56
4.7	Modèles de CPG décrits par Mc Crea et Ryback (figures extraites de (Mc Crea et Ryback, 2008)).	58
4.8	Plasticité synaptique (mécanisme de mise à l'échelle). Un neurone post-synaptique reçoit des entrées excitatrices de deux neurones pré-synaptique. Si l'activité post-synaptique est trop élevée, les mises à l'échelle synaptiques agissent proportionnellement à leur poids. Ce mécanisme de concurrence synaptique induit une potentialisation dans cette synapse et une dépression dans l'autre.	59
4.9	Régulation homéostatique de l'excitabilité neuronale. Pour un neurone, la fonction d'activation sigmoïdale lie l'activité totale reçue en entrée synaptique au taux moyen d'activité en sortie. Selon l'activité synaptique reçue, une partie seulement de la fonction d'activation est utilisée. Si cette partie correspond à une activité neuronale trop élevée ou trop basse, la fonction d'activation est déplacée pour revenir à un niveau d'activité « normale » : l'excitabilité est adaptée.	60
4.10	Neuromodulation extrinsèque (à gauche) et intrinsèque (à droite). Les neurones classiques sont en gris clairs et les neurones modulateurs en gris foncé (Extrait de (Marder et Bucher, 2001)).	60
4.11	Modèle de CPG proposé par Ekeberg pour la nage de la lamproie (figure modifiée de (Ekeberg, 1993)). Ce modèle inclut différents types d'interneurones inhibiteurs et excitateurs, deux motoneurones et des neurones sensoriels articulaires (Edge Cells). On voit bien la symétrie latérale dans la topologie du réseau et dans les interconnexions inhibitrices et excitatrices qui permettent de créer un mouvement rythmique par des actions antagonistes sur les motoneurones. Ce segment neuronal se répète à chaque articulation.	61
4.12	Modèle de contrôleur neuronal de la salamandre proposé par A. J. Ijspeert (figures extraites de (Ijspeert, 2001)). Le CPG (à gauche) est constitué de 3 types d'interneurones (A, B, C) et de motoneurones M. Les entrées BS (Brain Stem) provenant du niveau supérieur de la commande activent les neurones. L'architecture du contrôleur neuronal (à droite) est en 3 niveaux.	62
4.13	Architecture du CPG à base d'oscillateurs non linéaires (figure extraite de (Ijspeert et al., 2007)).	62
4.14	Architecture neuronale à base de CPG proposée par Taga et al. (figure modifiée de (Taga et al., 1991)).	63
4.15	Modèle de contrôleur proposé par F. Worgotter et robot bipède utilisé pour valider la méthode. <i>FS</i> et <i>ES</i> sont des neurones sensoriels détecteurs de flexion et d'extension. <i>FM</i> et <i>EM</i> sont les neurones moteurs de flexion et d'extension, <i>GR</i> et <i>GL</i> sont des neurones sensoriels (droit et gauche) détecteurs du contact pied-sol. (Extrait de (Worgotter et al., 2006)).	64
4.16	Architecture du neurocontrôleur à base d'oscillateurs de Hopf qui a permis de commander plusieurs allures de marche du robot Hoap-2 . La boucle de retour permet d'adapter la fréquence de chaque oscillateur au signal de synchronisation $P_{teach}(t)$ (figures extraites de (Righetti et Ijspeert, 2006)).	64
5.1	Plateformes expérimentales ayant servi à étudier les contrôleurs de type CTRNN avec l'algorithme de rétropropagation temporelle du gradient (BPTT) : à gauche le bipède ROBIAN du LISV, à droite le simulateur de route de la société BIA.	68
5.2	Principe de l'apprentissage du contrôle de l'équilibre. A gauche, position du capteur de force entre les membres inférieurs et le torse, à droite schéma d'apprentissage en ligne.	68
5.3	Application manuelle des forces extérieures sur le torse de ROBIAN.	69
5.4	Apprentissage de l'équilibre (extrait des enregistrements pour l'axe X uniquement). Au cours du temps le CTRNN apprend à commander le mouvement des masses pour compenser les forces mesurées (Hénaff et al., 2011).	70

5.5	Architecture temps-réel et ordonnancement propagation/rétropropagation (extrait de Scesa (2006)).	70
5.6	Mise en œuvre expérimentale pour l'évaluation des performances de l'algorithme BPTT par l'identification de l'équilibre de ROBIAN. Les mouvements des masses du torse induisent des variations sur la position du ZMP au sol, ce qui permet de construire les fichiers d'apprentissage et de test.	70
5.7	Mise en évidence de l'instabilité de la BPTT due à une explosion du gradient le long de la fenêtre temporelle. A gauche phénomène d'explosion qui fait diverger l'algorithme, à droite influence du pas d'apprentissage et de la taille de la fenêtre sur cette divergence (extraits de (Scesa, 2006)).	71
5.8	Influence de la taille de la fenêtre temporelle sur la stabilité de l'algorithme BPTT. Les indicateurs SDI et TES correspondent à la qualité de l'apprentissage, respectivement la rapidité de convergence et le taux d'erreur sur un fichier de test. Les pourcentages expriment le nombre d'essais qui ont convergé lorsque tous n'ont pas convergé (extraits de (Scesa, 2006)).	71
5.9	En haut : mise en évidence du phénomène de limitation de la dynamique dans les CTRNN en fonction des variations de la constante de temps calculées par l'algorithme d'apprentissage. En bas, le gradient lié aux variations de la constante de temps joue comme un facteur d'amplification dans le gradient rétropropagé. On voit bien que si l'expression de la constante de temps est linéaire (en bleue), ce facteur est inversement proportionnel à la période d'échantillonnage du modèle numérique et tend vers l'infini. Si l'expression est sigmoïdale (rouge), ce facteur est équilibré sur toute l'échelle de la constante de temps (extraits de (Scesa, 2006)).	72
5.10	Apprentissage en ligne de l'identification du comportement dynamique du simulateur de route de la société BIA : début d'apprentissage (en haut) et fin (en bas) après 300 boucles de rétropropagation temporelle sur la base d'apprentissage. Les accélérations sont celles mesurées sur la partie mobile dont la masse est 680 kg (ici une seule accélération est montrée)(extraits de (Scesa, 2006)).	72
5.11	Décomposition du cycle de la marche humaine en 8 phases élémentaires (Serhan <i>et al.</i> (2008b,a,c)).	73
5.12	Approche du contrôle articulaire compliant par réseau de neurones. L'émulateur de muscle adapte les gains articulaires à la dynamique de la consigne. L'algorithme de marche dynamique est basé sur une machine à états (Serhan <i>et al.</i> (2008b,a,c)).	74
5.13	Schéma de la structure d'apprentissage choisie pour l'émulateur de muscle. A gauche avec un modèle mathématique d'actionneur, à droite en apprenant un modèle neuronal de l'actionneur. $y(t)$ est le couple moteur, $y_d(t)$ est le couple du modèle de muscle (consigne), $r(t)$ est le signal de commande, et $I(t) = [e(t), e(t-1), e(t-2), \rho(t)]$ un vecteur composé de trois échantillons du signal d'entrée $e(t)$ et d'un signal image du sens de variation de $r(t)$	74
5.14	A gauche : Structure du perceptron chargé de calculer les gains du PID de la boucle d'asservissement articulaire. A droite comparaison de la réponse du moteur DC avec le modèle de muscle, effet de l'apprentissage de ce perceptron en cas de variation de la dynamique du moteur.	75
5.15	Comparaison des trajectoires articulaires entre l'humain (marche moyenne d'un enfant de la taille de ROBIAN) et ROBIAN (jambe droite) avec et sans émulateur de muscle : angles de la hanche, du genou, et de la cheville pour deux cycles de marche. La vitesse du robot est 0.6 m/sec. Les phases indiquées sont celles de la figure (5.11) issues de la décomposition du cycle de la marche humaine.	75
5.16	Etude comparative du contrôle de l'équilibre avec et sans compliance contrôlée sur les genoux du robot bipède. Vitesses et position de ROBIAN dans le plan sagittal, après application d'une force de poussée perpendiculaire au sens de la marche (plan frontal). La courbe bleue correspondant à la configuration sans compliance montre la chute du robot : en haut la vitesse de la marche devient nulle, en bas la trajectoire est interrompue.	76

5.17	Mise en évidence de l'instabilité du contrôleur proposé par F. Worgotter en fonction des paramètres du neurone sensoriel extenseur <i>ES</i> de chaque hanche. A gauche, critère utilisé pour valider ou non la stabilité de la marche. A droite, en gris : la dynamique contrôleur/robot est instable et le robot tombe. En couleur, cette dynamique est stable et la vitesse de la marche (exprimée ici par les couleurs) dépend de la valeur des paramètres.	77
5.18	Mécanisme d'apprentissage de la vigilance avec un système d'alerte prévisionnelle (<i>EWS</i> : Early Warning System). A gauche, l'apprentissage est effectué sur 1500 essais de marche avec une pente du sol aléatoire. A droite, le mécanisme d'apprentissage évalue la distance entre le vecteur d'état du système et chaque carte. Un changement de la pente conduit à faire basculer le système de décision entre les neurones pour assurer la marche.	78
5.19	Mise en évidence de l'efficacité du mécanisme d'adaptation : les paramètres des neurones sensoriels sont adaptés à la pente.	78
6.1	Structure modulaire du contrôleur global de l'hexapode et chronogramme du cycle de la marche d'une patte de l'hexapode. Lors de la phase de transfert la patte n'est pas en contact avec le sol, elle revient en avant en se soulevant, puis en se baissant suivant 4 sous-séquences (B : perte de contact avec le sol, C : transfert montant, D : transfert descendant vers l'avant, E : descente et prise de contact avec le sol). . . .	80
6.2	Générateurs de rythmes. A gauche : oscillateur basique à 1 sortie. A droite : oscillateurs à 2 sorties en opposition de phase.	80
6.3	Bloc fonctionnel commandant le mouvement vers l'avant d'une articulation. L'entrée I_0 est l'impulsion qui déclenche le mouvement, la sortie O_0 correspond à l'action, c'est-à-dire à la vitesse angulaire durant le mouvement, la sortie O_2 annonce la fin du mouvement, I_1 est le signal image de la position angulaire.	80
6.4	Structure du contrôleur complet pour le cycle de marche de l'hexapode	81
6.5	Simulation dynamique de l'hexapode sous ODE (Open Dynamic Engine : http://www.ode.org/ode.html)	82
6.6	Amplitude des mouvements des pattes lors du cycle de marche de l'hexapode. En haut, début du cycle et zoom sur 2 pattes où l'on aperçoit que l'une d'elle reste inerte entre les itérations 600 et 800. En bas cycle de marche après adaptation du contrôleur neuronal. On constate que le cycle tripode alterné est parfaitement obtenu (zoom à droite).	82
6.7	Variation des paramètres de neuromodulation d'une patte durant la marche du robot. Le temps pendant laquelle la patte reste en butée diminue car la sensibilité du neurone chargé de la fréquence du cycle augmente tandis que celui chargé du contrôle de la vitesse diminue. L'efficacité du cycle de la marche augmente et la coordination des pattes apparaît.	82
6.8	Robot à une patte et le modèle d'hexapode correspondant lorsqu'il est amputé de 2 pattes	84
6.9	Comparaison des commandes articulaires calculées par le meilleur neurocontrôleur d'une population : a) perceptron récurrent standard, b) perceptron récurrent avec neurones dynamiques de Beer. Dans les deux cas, les biais sont calculés par l'évolution. c) même réseau avec neurones de Beer, mais le biais est calculé par un mécanisme homéostatique.	85
6.10	Fonction d'évaluation pour les 3 types de réseau. L'indice de performance est la distance parcourue par le robot en 10s. A gauche : meilleur contrôleur d'une population. Chaque enjambée est visible par une variation rapide de la distance parcourue. A droite : évolution de la valeur moyenne de la fonction d'évaluation d'une population (1000 générations, 200 neurocontrôleur). L'écart type est calculé sur 10 expériences.	85
6.11	Morphologie et débattements articulaires du robot monopode glissant sur son rail.	86
6.12	Etude statistique de l'influence des différents modèles de neurones sur les performances des neurocontrôleurs à chaque génération (à gauche) et pour les quatres scénarios d'évalutaion apres évolution pour une durée de 100s.	88

6.13	Suivi de consignes de vitesse V_d avec ou sans perturbation (consignes sinusoïdales et en échelon) et variations temporelles des angles articulaires. Chaque oscillation de la vitesse mesurée V_m correspond à une foulée du robot. On note une phase transitoire pendant laquelle le contrôleur neuronal converge vers son cycle stable. La perturbation provoquée par l'augmentation des frottements visqueux induit une légère diminution de la vitesse.	88
6.14	Neutralisation de la plasticité du contrôleur aux instants $t=0s$ (à gauche), $t=6s$ (milieu) et $t=18s$ (à droite). Les bandes grises montrent le domaine de variation angulaire de chaque articulation en fonctionnement normal. On constate nettement que la rythmicité du contrôleur est fortement perturbée lorsque la plasticité disparaît : la consigne est non satisfaite (gauche et droite), ou le contrôleur est incapable d'exhiber des comportements d'arrêt (milieu).	89
6.15	Mise en évidence de l'adaptation du cycle de marche de la patte du monopode. En gris clair, sans perturbation, en gris foncé, avec perturbation (doublement des frottements sur le rail). On voit que les mécanismes homéostatiques provoquent un recul du point d'appui de la patte et un allongement de l'enjambée. La répartition des point d'appui sur le sol (à droite) montre les variations dans le cycle de la patte (extrait de Hoinville (2007)).	90
6.16	Mise en évidence des sous-structures identifiées dans le neurocontrôleur obtenu (à gauche). Quand on observe attentivement la structure de ce contrôleur, on peut distinguer une sous-structure en boucle fermée régulatrice de la vitesse (au milieu) et une autre sous-structure coordinatrice du mouvement de la patte (à droite). . .	90
6.17	Modèle de robot myriapode utilisé dont le nombre de modules est variable. Exemple d'une structure de robot à 19 modules (longueur 3m80). En bas, deux types de déficiences simulées : amputations de pattes et du corps. (extrait de (Hoinville, 2007)).	91
6.18	Plateforme de d'évaluation de chaque neurocontrôleur. La vitesse du tapis est variable. Le module de tête du robot (en jaune) doit resté maintenu dans un « domaine de viabilité » représenté par le carré rouge (extrait de (Hoinville, 2007))	92
6.19	Trajectoires obtenues après évolution : à gauche pour l'hexapode, à droite pour un décapode. La consigne reçue par les capteurs sensoriels de direction est d'effectuer une trajectoire rectiligne à faible vitesse ($C=20\%$) pendant les 30 premières secondes, puis à vitesse maximale ($C=100\%$) pendant les 30 suivantes. Une trajectoire rectiligne est donc un bon résultat. Le régime transitoire correspondant à la mise coordination des pattes provoque un désalignement de la trajectoire du robot par rapport à l'axe y visible sur la figure.	92
7.1	Mécanisme de synchronisation de phase et boucle de résonance entre un système actif et un système dissipatif. Selon les valeurs du paramètre de couplage γ , l'oscillateur chaotique synchronise ou non la phase du système dissipatif (le robot). Figure extraite de (Pitti <i>et al.</i> , 2009).	96
7.2	Modèle de CPG, pour l'articulation du genou (le CPG au niveau de la hanche n'est pas représenté pour plus de clareté). Les entrées proprioceptives proviennent de la hanche (θ_{hip} angle d'extension) et les entrées extéroceptives du contact contre le sol (CF : intensité de la composante verticale de la force de contact \vec{F}). A droite, statistique comparative (247 configurations initiales différentes de la patte) pour huit valeurs du paramètre σ_s du neurone (paramètre pilotant la fréquence d'oscillation, voir équation 4.3). En rouge pourcentage d'essais aboutissant à un déplacement du robot. En bleu : valeur moyenne de l'augmentation de la distance parcourure avec le retour sensoriel du sol GCS par rapport à sans GCS	97
7.3	(a) Structure du CPG à base de neurones rythmiques de Rowat et Selverston. (b) propriété du CPG en fonction de ses paramètres de conductance σ_s et σ_f . (c) sauts de phase entre deux CPG contrôlant une patte.	98
7.4	Architecture du neurocontrôleur complet du bipède : (a) Modèle de bipède, (b) Couplages articulaires entres les neurones rythmiques, (c) Contrôleur du tronc . .	98
7.5	Détail des CPG pour chaque articulation	99

7.6	Représentation matricielle 2D des modes rythmiques : (a) au niveau du CPG où les neurones de la carte de Kohonen sont classés suivant l'indice énergétique ε du CPG, (b) au niveau de la patte où le mouvement de la patte dépend de l'indice énergétique du neurone de la carte.	99
7.7	Réflexe de changement d'allure suite à une perturbation. (a) sans mécanisme d'adaptation et sans perturbation la marche est stable, (b) sans mécanisme d'adaptation et avec perturbation le robot tombe, (c) avec mécanisme d'adaptation et avec perturbation le robot fait un grand pas vers l'avant pour rattraper son équilibre. . . .	100

Liste des tableaux

2.1	Tableau synthétique des heures enseignées depuis 2000	32
6.1	Trois scénarii d'évaluation des contrôleurs du monopode : différentes formes de consignes et différentes valeurs de frottements visqueux contre l'environnement. . .	87

Introduction générale

Lors de ma nomination en 1997, en tant que Maître de Conférences à l'IUT de Cergy Pontoise, le Directeur de l'IUT m'a confié la responsabilité du département GEII, option Réseaux Locaux Industriels, qui venait d'être créé sur le site délocalisé de l'Université de Cergy-Pontoise, à Sarcelles, distant de 35km du pôle Universitaire Cergy-Pontain. Pendant 3 ans j'ai assumé avec enthousiasme cette lourde charge de lancer un tout nouveau département d'IUT dans une agglomération peu favorisée de l'Ile de France. Le budget d'équipement était conséquent et j'ai pu assurer l'équipement complet du département (matériels scientifiques, techniques, informatiques, mobiliers...). J'ai organisé et défini la pédagogie autour d'une coloration « locale » liée à l'option RLI, c'est-à-dire orientée vers les réseaux de terrains industriels et la maintenance des réseaux d'entreprises. J'ai formé mes collègues recrutés aux principes de fonctionnement des IUT. Je les ai initiés à une pédagogie ouverte sur nos jeunes étudiants mais rigoureuse sur la qualité de la formation. Les charges pédagogiques et administratives dans les IUT sont en général importantes. Elles le sont bien plus sur un site distant et isolé de « l'université mère ». Durant 3 années, j'ai mené ce travail d'administration à temps plein, tout en gardant un lien suffisant avec les laboratoires de recherche en robotique, mais trop ténu pour mener simultanément une activité de recherche active et productive en termes de publications.

Après ces 3 années d'investissement total à l'IUT, j'ai mis près d'une année supplémentaire à relancer ma carrière de chercheur en faisant le choix, plutôt que de tenter une conversion thématique qui aurait rallongé ma « période creuse », de rester sur ma thématique initiale mais non présente à l'UCP, celle de la commande des robots marcheurs. En septembre 2001, j'ai rejoint, avec l'accord de l'UCP, l'équipe Robot Humanoïde du laboratoire LIRIS (devenu LISV¹) de l'Université de Versailles. Cette équipe dirigée, par F. Ben Ouezdou est spécialisée dans la conception, la simulation et la modélisation des robots marcheurs. J'ai pu relancer ainsi une activité de recherche originale issue des travaux de ma thèse sur la commande neuronale, mais axée sur la robotique humanoïde, domaine que la communauté roboticienne Française commençait à développer. Pendant les 8 années passées au LISV, le travail de recherche mené en collaboration avec les mécaniciens m'a permis de développer un « regard de roboticien » sur mes travaux en commande bio-inspirée des robots marcheurs. J'ai pu construire une activité de recherche active et combler ainsi en partie « mon déficit recherche », tout en maintenant une implication forte dans le fonctionnement du site de Sarcelles.

En 2008, j'ai rejoint l'équipe Neurocybernétique du laboratoire ETIS de l'UCP, dirigée par P. Gaussier. Nous collaborons étroitement avec le LISV sur des projets de robotique humanoïde. L'équipe Neurocybernétique est reconnue nationalement et internationalement pour ses travaux dans les domaines de l'apprentissage et de l'imitation, plus généralement sur la modélisation des processus cognitifs validés sur des systèmes robotisés, robots mobiles ou robots humanoïdes. Mon expérience apporte à cette équipe des connaissances plus axées sur l'apprentissage de la commande bas niveau, l'automatique et sur la genèse de mouvements rythmiques.

L'objectif scientifique que j'ai suivi consiste à développer et à intégrer aux contrôleurs de robots à pattes, des mécanismes adaptatifs inspirés de la biologie afin que ces robots s'adaptent aux dommages internes, soudains ou progressifs, qu'ils peuvent subir et qui mettent en péril leur

1. Le LISV a été créé en janvier 2006 suite à une fusion de trois laboratoires de l'Université de Versailles : le Laboratoire d'Instrumentation et de Relation Individus Systèmes (LIRIS) auquel j'appartenais, le Laboratoire de Robotique de Versailles (LRV), et le Laboratoire d'Études Mécaniques des Assemblages (LEMA). Le laboratoire LIRIS était lui-même issu du démantèlement du Laboratoire de Robotique de Paris en janvier 2002. Mon activité de recherche dans ces laboratoires faisait l'objet d'une convention d'accueil entre l'UCP et l'UVSQ.

mission (usures articulaires, amputation de membres moteurs, pertes sensorielles ...) et à des perturbations externes dues à l'environnement (glissement, pentes, forces externes). Pour ce faire, j'étudie les capacités de généralisation offertes par les algorithmes d'apprentissage des réseaux de neurones et les synergies de leurs mécanismes adaptatifs (homéostasie, plasticité neuronale et synaptique). Je cherche à confronter nos solutions en soumettant les robots à des perturbations ou à des dysfonctionnements externes ou internes, lents ou brusques. J'évalue et je compare entre eux ces mécanismes biomimétiques aux solutions plus classiques de l'automatique.

Je cherche donc à répondre aux questions suivantes : comment la marche adaptative des robots peut elle être produite ? Comment peut-elle être contrôlée ? Quels types d'architecture permettent à la fois la production de rythmes locomoteurs et le contrôle de la posture ? Quels sont les mécanismes adaptatifs sensori-moteurs qui régissent ces architectures pour les marches normales et déficientes des robots ? Quelles sont les méthodologies possibles pour modéliser et reproduire ces architectures ? Quelles sont les limites de ces approches ?

Le présent rapport d'Habilitation à Diriger les Recherches retrace mon parcours en termes d'enseignement, d'administration et de recherche depuis 1997. La première partie de ce rapport est consacrée à mon Curriculum Vitae et à la synthèse de mes activités d'enseignement.

La seconde partie présente de façon plus approfondie mon activité de recherche sur la commande bio-inspirée particulièrement liée à l'encadrement doctoral que j'ai commencé en 2002 au LISV et que je poursuis à ETIS depuis 2008. Après un chapitre introductif qui présente mon activité de recherche dans la suite de mes travaux de thèse, un second chapitre est consacré aux aspects biologiques du contrôle moteur dans lequel je résume les fondamentaux qui ont guidé mon travail. Le troisième chapitre présente mes travaux de simulation et expérimentaux sur le contrôle de l'équilibre des bipèdes par apprentissage. Le quatrième présente les études menées sur la genèse de rythmes locomoteurs pour les robots myriapodes. Le cinquième concerne les structures de neurocontrôleurs pour la marche bipède. Enfin, le dernier chapitre est dédié à la conclusion et propose des perspectives de travaux dans le cadre de l'apprentissage biomimétique appliqué à la robotique humanoïde.

Première partie
Curriculum Vitæ

Chapitre 1

Curriculum Vitae

1.1 Position personnelle et professionnelle

► **État civil :**

Nom : HÉNAFF

Prénom : Patrick Jean François

Date et lieu de naissance 11 mai 1965 à Bois Colombes

Situation familiale : Marié, 2 enfants

Nationalité : Française

► **Situation Professionnelle :**

Depuis 1997 : Maître de Conférences 61^{ème} section, Université de Cergy-Pontoise

Enseignement :

Département : Institut Universitaire de Technologie, département Génie Électrique et Informatique Industrielle (site de Sarcelles)

Thèmes d'enseignement :

- Automatique, Robotique
- Automatismes Industriels et Réseaux Locaux Industriels
- Robotique et contrôle bio-inspiré
- Informatique, Systèmes Temps Réels

Recherche :

Laboratoire ETIS UMR 8051, CNRS-UCP-ENSEA, Directrice I. Fijalkow.

Thèmes de recherche :

- Commande par réseaux de neurones
- Contrôleurs adaptatifs inspirés de la biologie
- Robotique humanoïde

1.2 Titres Universitaires

► **Juin 1994 :** Doctorat de l'Université Pierre et Marie Curie – Spécialité Robotique

Titre : Mises en Œuvre de Commandes Neuronales Par Rétro-Propagation Indirecte : Applications à la Robotique Mobile

Jury :

P. COIFFET, Directeur de recherche C.N.R.S, président

M. MILGRAM , Professeur à l'Université Paris 6, Directeur de thèse

W. KHALIL, Professeur à L'Ecole Centrale de Nantes, rapporteur
 T. MAURIN, Professeur à L'E.N.S. de Cachan, rapporteur
 J. RABIT, Professeur à l'Université Paris 11, examinateur
 Mme de SAINT BLANCAR, Ingénieur de Recherche à P.S.A.

- ▶ **Juin 1989** : DEA de Robotique de l'Université Pierre et Marie Curie –ENSAM-INSTN
- ▶ **Juin 1988** : Maîtrise EEA de l'Université de Rennes I
- ▶ **Juin 1985** : DUT d'Electronique et d'Informatique Industrielle, Université de Rennes I , Lan-nion

1.3 Parcours académique

▶ Recherche :

1997-2002 : Laboratoire de Robotique de Paris (LRP), CNRS-Université de Versailles St Quentin- Université Pierre et Marie Curie
2002-2008 : Laboratoire d'Ingénierie des Systèmes de Versailles (LISV), Directeur Y. Alayli, Université de Versailles Saint Quentin
2009- : Laboratoire ETIS UMR 8051, CNRS-UCP-ENSEA, Directrice I. Fijalkow, Université de Cergy Pontoise

▶ Promotions :

1999 : Passage au grade de 1^{ère} classe (contingent local)
2005 : bénéficiaire de la PEDR
2006 : CRCT de 2 semestres (contingent local)

▶ Responsabilités pédagogiques et administratives

1995-1996 : Directeur des études, Département GEII , IUT de Cergy-Pontoise
1997-2001 : **Chef du Département GEII de Sarcelles, IUT de Cergy-Pontoise**
 Site de Sarcelles : Création du département (Réseaux Locaux Industriels), mise en place complète du département.
2008- : **Responsable des stages, département GEII de Sarcelles, IUT de Cergy-Pontoise.**
 Suivi individuel des étudiants de 2^{ème} année dans leur démarche de recherche de stage.
2011-2014 : **Responsable d'un Master Franco/Ukrainien :**
 Master de l'UCP « systèmes intelligents et Communicants, parcours Robotique et Intelligence artificielle » adapté pour les étudiants francophones de l'Université Nationale Technique de Donetsk . Financement ambassade de France 2011/2014

▶ Situations contractuelles avant la nomination en tant que Maître de Conférences

1990-1993 : Allocataire de recherche Paris 6, moniteur Paris 7
1993-1996 : I.U.T de Cergy Pontoise, Département GEII : ATER
1996-1997 : I.U.T de Vélizy, Département GEII : PRAG Contractuel

1.4 Rayonnement scientifique national et international

- ▶ **Conseil de Laboratoire** : membre élu au CdL de ETIS depuis septembre 2010.
- ▶ **Conseil d'Administration** : membre élu au conseil de l'IUT de Cergy-Pontoise depuis 1998.
- ▶ **Commissions de Spécialistes et Comités de Sélections**
 - Membre titulaire du Comité de sélection 61^{ème} section de l'Université de Versailles St Quentin (2009)

- Membre titulaire du Comité de sélection 61^{ème}section de l'Université de Cergy-Pontoise (2008, 2010, 2011)
- Membre titulaire du Comité de sélection 63^{ème}section de l'Université de Cergy-Pontoise (2011)
- Membre titulaire de la CSE mixte 61/63^{èmes}sections de l'Université de Versailles S^t Quentin (2001-2008)
- Membre titulaire CSE 61^{ème}section de l'Université de Cergy-Pontoise (2004-2007)
- Membre suppléant de la CSE mixte 61/63^{èmes} section de l'Université de Cergy-Pontoise (2000/2004)

► **Jurys de thèse (examinateur) :**

- **En tant que co-encadrant scientifique :**

Thèse de Doctorat de l'Université de Versailles Saint Quentin : Vincent Scesa, le 20 octobre 2006, « Contrôle adaptatif par réseaux de neurones : évaluation d'une implémentation temps réel sur un contrôleur industriel multiprocesseur pour le contrôle du comportement dynamique d'une structure mécanique complexe ».

Thèse de Doctorat de l'Université de Versailles Saint Quentin : Thierry Hoinville, 25 octobre 2007, « Evolution de contrôleurs neuronaux plastiques. De la locomotion adaptée vers la locomotion adaptative »

Thèse de Doctorat de l'Université de Versailles Saint Quentin : Hayssam Serhan, 9 juillet 2009 : « Approches biomimétiques pour le contrôle de la marche des robots bipèdes : stratégie intuitive et intégration d'un modèle de muscle »

- **En tant que membre extérieur :**

Thèse de Doctorat de l'Université Pierre et Marie Curie : Stéphane Doncieux le 23 juin 2003, « Evolution de contrôleurs neuronaux pour animats volants : méthodologie et applications »

Thèse de Doctorat de l'Université de Cergy-Pontoise : Christophe Giovanangeli, le, 13 décembre 2007 « Navigation autonome bio-inspirée en environnement intérieur et extérieur : apprentissage sensori-moteurs et planification dans un cadre interactif ».

► **Comités de lecture :**

- **Revue/Journaux :** JESA (2001, 2002, 2003, 2004, 2005), AMSE Journal of Advances in Modelling (2006), IEEE Trans. on Automation Science and Engineering (2006), ROBOTICA (2007, 2008), IEEE Trans on Control and Systems Technology (2008), Neural Networks (2009), Frontiers in Neurosciences-Neurorobotics (2009), International Journal of Neural Systems (2009, 2010, 2011).
- **Conférences Internationales :** CIFA (2003), Handicap (2004), ICRA (2008), IEEE/WCCI (2008), IEEE/IJCNN (2007, 2008, 2009, 2011), ICIRA (2011),
- **Reviewer Editor :** revue Frontiers in NeuroRobotics published by Frontiers Research Foundation (<http://frontiersin.org/editorialboards#neurorobotics>)s

► **Expertises :**

- **Expert ANR 2008 :** programme « Jeunes chercheuses et Jeunes chercheurs »
- **Expert COFECUB 2011 :** Programme Franco-Brésilien de coopération scientifique .
- **Expert externe de l'Université TUT, Pretoria, Afrique du Sud (2007, 2008) :** évaluation de rapports de recherche d'étudiants en master.

1.5 Expérience en entreprise

► **Mai-juin 1985 :**

THOMSON-CSF (Brest), Service Technique. "Elaboration d'une mire de test d'un système de traitement d'images radar".

► **Juillet-Août 1987 :**

E.D.F (Villeurbanne), Service Études et Projets Thermiques et Nucléaires. "Etude bibliographique : testabilité des composants VLSI, machines à architecture systolique"

► **Août-Septembre 1989 :**

MATRA ESPACE (Vélizy) Département Téléopération et Robotique. "Modélisation et simulation (C.A.O EUCLID-IS) d'un robot d'intervention spatial".

► **Avril-Juillet 1989 :**

C.E.A (Fontenay aux Roses) : Unité de Génie Robotique Avancée. "Etude, modélisation de l'interaction Homme/Machine d'un bras de téléopération spatial".

► **Novembre 1989-Octobre 1990 :**

DIRCEN (Direction Des Centres D'Expérimentations Nucléaires) : Service National effectué dans l'Armée de l'Air (Villacoublay-Mururoa) : Responsable du projet d'informatisation du service logistique de gestion et de transport du fret aérien (en collaboration avec une SSII). Conception et validation des interfaces de saisies. Formation du personnel (officiers supérieurs, officiers et sous officiers). Mission de formation sur les sites militaires du Pacifique (Tahiti-Faaa) : 30 jours.

1.6 Synthèse de l'activité de recherche

Objectif scientifique :

Mon objectif scientifique consiste à développer et intégrer aux contrôleurs de robots, des mécanismes adaptatifs inspirés de la biologie afin que ces robots s'adaptent aux dommages internes, soudains ou progressifs, qu'ils peuvent subir et qui mettent en péril leur mission (usures articulaires, amputation de membres moteurs, pertes sensorielles ...) et à des perturbations externes dues à l'environnement (glissement, pentes, forces externes). Pour ce faire, j'étudie les capacités de généralisation offertes par les algorithmes d'apprentissage des réseaux de neurones et les synergies de leurs mécanismes adaptatifs (homéostasie, plasticité neuronale et synaptique). Je compare nos solutions en soumettant les robots à des perturbations ou à des dysfonctionnements externes ou internes, lents ou brusques. J'évalue et compare entre eux ces mécanismes biomimétiques par rapport aux solutions plus classiques de l'automatique. Je cherche donc à répondre aux questions suivantes : comment la marche adaptative des robots peut elle être produite ? Comment peut-elle être contrôlée ? Quels types d'architecture permettent à la fois la production de rythmes locomoteurs et le contrôle de la posture ? Quels sont les mécanismes adaptatifs sensori-moteurs qui régissent ces architectures pour les marches normales et déficientes des robots ? Quelles sont les méthodologies possibles pour modéliser et reproduire ces architectures ? Quelles sont les limites de ces approches ?

1.6.1 Principales Responsabilités Scientifiques

► **2002-2005 : Responsable scientifique du projet IRON**

Implémentation Robotique de Neurocontrôleurs adaptatifs. Programme ROBEA du CNRS : pré-projet 2002-2003, projet 2003-2005. Partenaires : LIP6 (AnimatLab, J. A. Meyer), Centre de Neurosciences Intégratives et Cognitives (Equipe Comportement et Réseaux Neuronaux, Université de Bordeaux 1, Daniel Cattaert,).

► **2005-2009 : Co-Responsable scientifique du projet PHEMA**

« Plateforme Humanoïde pour l'Etude de la MArche avec et sans déficiences motrices », avec F. B. Ouezdou, LISV. Programme blanc ANR 2005. Partenaires : Centre de Neurosciences Intégratives et Cognitives (Equipe Comportement et Réseaux Neuronaux, Université de Bordeaux 1, Daniel Cattaert,), IRCCYN (Université de Nantes, Y. Aoustin), service de rééducation fonctionnelle de l'Hôpital de Garches Raymond Poincaré (Pr. B. Bussel,).

► **2011-2014 : Responsable du projet EGIDE-PHC DNIPRO (UKRAINE)**

Modèles Bio-inspirés pour les Robots Humanoïdes en Interactions Rythmiques avec leur Environnement, Université Nationale Technique de Donetsk/Université de Cergy-Pontoise.

1.6.2 Activités Internationales

► Université Nationale Technique de Donetsk, Ukraine :

Depuis décembre 2008, collaboration avec le Pr. Vladimir Borissenko chaire « Commande Electrique et Automatisation des Installations Industrielles » :

- **2011 : Création d'un Master Franco/Ukrainien** : Master adossé au Master SIC de l'UCP « systèmes intelligents et Communicants, parcours Robotique et Intelligence artificielle » adapté pour les étudiants francophones de l'UNTD. Financement ambassade de France 2011/2014
- **Responsable du Projet Egide-PHC Dnipro 2011/2013** : UCP/UNTD : « Modèles Bio-inspirés pour les Robots Humanoïdes en Interactions Rythmiques avec leur Environnement »
- 2 Cotutelles de thèse :
 - Slava Khomenko, « Compensation des déficiences électriques ou mécaniques pour la marche des robots humanoïdes ».
 - Artem Melnyk, « Perfectionnement des algorithmes de contrôle-commande des robots manipulateurs électriques en interaction physique avec leur environnement par une approche bio-inspirée »
- 8 séjours d'une semaine depuis septembre 2009
- Conventions de coopération UCP/UNTD et UVSQ/UNTD
- Séminaires de 18 heures sur le thème « Contrôle Bio-Inspiré pour la robotique » (2010, 2011).

► Université Libanaise :

- Thèse de Hayssam Serhan soutenue le 9 juillet 2009 sur le thème de la modélisation neuronale d'actionneur bio-fidèles, codirection de thèse avec le Pr. C. Nasr de l'Université Libanaise, faculté du Génie.
- Séminaire de 12 heures en Master 2 de l'Université Libanaise sur le thème « Contrôle Bio-Inspiré pour la robotique »

► Université TUT, Pretoria, Centre Franco-Sud Africain F'SATIE , Afrique du sud :

- **Décembre 2007** : Expert externe pour l'évaluation d'un rapport de Master de technologie : « Identify verification system using hand gesture Information ». superviseur Pr. B.J Van Wyk
- **Avril 2008** : Expert externe pour l'évaluation d'un rapport de Master de technologie : « A comparison of algorithms for image-based classification » . Superviseur Pr. B.J Van Wyk

► Université Otto Von Guericke de MagdeBurg :

- **2003/2005, Projet PAI-Egide (Procope)** :
Collaboration avec le Pr. Franck Palis de l'Université Otto Von Guericke de MagdeBurg (Allemagne) et le Pr. P. Y. Glorennec de l'IRISA pour un projet sur « L'Étude de la commande symbolique et neuronale d'une main artificielle à partir de signaux myo-électriques ».
- **Octobre 2006** : Séjour du Pr. Palis au LISV en tant que professeur invité. Convention de partenariat Socrates a été signée en janvier 2007 entre l'UVSQ et l'Université Otto Von Guericke.

► Co-encadrement doctoral avec le Pr G. Cheng

(Actuellement à *Technical University of Munich, CoTeSys , Central Robotics Laboratory*) : Dans le cadre de son séjour d'un mois, en tant que Professeur Invité, au LISV en mai 2008, G. Cheng a collaboré avec moi, F. B. Ouezdou et J. Nassour (doctorant que j'encadre à 60%). Ce travail a abouti à deux publications communes (conférences IEEE/IROS 2009, SAB 2010). John Nassour a obtenu un financement Franco/Bavarois pour effectuer 4 séjours d'un mois auprès de G. Cheng au Munich Technical University.

1.6.3 Encadrement doctoral

► BAYARD Gildas (2000-2004)

→ Thèse abandonnée en 2004

Encadrement : 30%, Jean Arcady Meyer-LIP6 70% (directeur de thèse).

Financement : BDI CNRS 2000-2003 au LIP6-UPMC

Titre : « Apprentissage en ligne des contrôleurs neuronaux pour des systèmes à locomotion articulée. »

► **THIEFFRY Rolland (2000-2004)**

→ Thèse soutenue le 8 décembre 2004, *mention Très Honorable*.

Financement : MNERT 00-03, ATER 2003-2004

Titre : « Étude et validation de systèmes adaptatifs d'aide à l'autonomie des personnes handicapées ».

Encadrement : 30%, Eric Monacelli 60% (co-encadrant), S. Delaplace 10% (directeur de thèse) .

Situation actuelle du docteur : Ingénieur de recherche Agence Française contre la Myopathie.

Publications : 1 article de revue interationale, 1 article de conférence internationale.

► **HOINVILLE Thierry (2002-2007)**

→ Thèse soutenue le 25 octobre 2007, *mention Très Honorable*.

Encadrement : 80%, S. Delaplace 20% (directeur de thèse) .

Titre : « Évolution de contrôleurs neuronaux plastiques. De la locomotion adaptée vers la locomotion adaptative».

Financement : MNERT 02-05, ATER 2005-2007,

Situation actuelle du docteur : Post-Doc à l'Italian Institute of Technology, équipe Te-lerobotics and Applications

Publications : 1 article de revue internationale, 2 articles de conférence internationale, 1 article de revue européenne.

► **SCESA Vincent (2003-2006)**

→ Thèse soutenue le 20 octobre 2006, *mention Très Honorable*.

Encadrement : 80%, F. B. Ouezdou 20% (directeur de thèse) .

Titre : « Contrôle adaptatif par réseaux de neurones : évaluation d'une implémentation temps réel sur un contrôleur industriel multiprocesseur pour le contrôle du comportement dynamique d'une structure mécanique complexe ».

Financement : CIFRE , entreprise BIA

Situation actuelle du docteur : Ingénieur consultant en systèmes robotiques

Publications : 1 article de revue interationale, 3 articles de conférence internationale.

► **SERHAN Hayssam (2005-2009)**

→ Thèse soutenue le 9 juillet 2009, *mention Très Honorable*.

Encadrement : 60%, Pr. C. Nasr 20% (Co-directeur au Liban), F. B Ouezdou 20% (Directeur de thèse).

Titre : « Approches Biomimétiques pour le contrôle de la marche des robots bipèdes : stratégie intuitive et intégration d'un modèle de muscle ».

Financement : salarié au Liban, coopération Franco-Libanaise entre l'Université de Versailles St Quentin et l'Université Libanaise.

Situation actuelle du docteur : salarié au Liban, Enseignant associé à l'Université Libanaise

Publications : 1 article de revue interationale, 4 articles de conférence internationale.

► **NASSOUR John (2007- juillet 2011)**

Encadrement : 60%, G. Cheng 20%, F. B. Ouezdou 20% (directeur de thèse) .

Titre : « Implémentation de contrôleurs neuronaux adaptatifs pour la marche déficiente des robots bipèdes».

Financement : Bourse du gouvernement Syrien

Publications : 2 articles de conférence internationale, 1 workshop international

► **KHOMENKO Viacheslav (janvier 2009- décembre 2011)**

Encadrement : Co-tutelle de thèse (UVSQ/UNTD) avec Pr. V. Borissenko, Université Technique Nationale de Donetsk, Ukraine. Encadrement 60%, directeurs de thèse 40% (V. Borissenko, 30% , F. B. Ouezdou 10%) .

Titre : « Modélisation et compensation des déficiences linéaires et non linéaires dans les transmissions électromécaniques des robots humanoïdes ».

Financement des séjours en France :

- de janvier 2009 à juillet 2010 : bourse AUF obtenue .
- de novembre 2010 à avril 2011 : financement LISV et projet PHC-Dnipro

Publications : 1 article de revue (Ukraine), 2 articles de conférences internationales (Ukraine)

► **MELNYK Artem (janvier 2010-décembre 2012)**

Encadrement : co-tutelle de thèse (UCP/UNTD) avec Pr. V. Borissenko, Université Technique Nationale de Donetsk, Ukraine. Encadrement 60%, directeurs de thèse 40% (V. Borissenko, 30% , P. Gaussier 10%) .

Titre : « Perfectionnement des algorithmes de contrôle-commande des robots manipulateurs électriques en interaction physique avec leur environnement par une approche bio-inspirée ».

Financement des séjours en France :

- de avril à juin 2011 : projet PHC-Dnipro, ambassade de France en Ukraine et Ecole Doctorale UCP.

Publications : 1 article de conférence internationale.

1.6.4 Encadrement de Master, DEA et ingénieurs

Mars à juillet 1993 : Un stagiaire ingénieur de l'Université de Karlsruhe, sur « le contrôle par réseaux de neurones de l'équilibre dynamique d'un bipède plan ». Encadrement 100%.

Mars à juillet 1995 : Un stagiaire du D.E.A de Robotique de l'Université Paris 6, sur « la commande articulaire neuro-floue d'un robot manipulateur rapide ». Encadrement 100%.

Mars à juillet 1996 : Un stagiaire du D.E.A de Robotique de l'Université Paris 6, sur « la commande neuronale d'un bipède 3d à 35 ddl simulé sous CAO SDS ». Co-encadrement 70%, Pr. F. B. Ouezdou 30%

Janvier à juin 2000 : Un stagiaire Ingénieur (ESIA), sur « l'Apprentissage de la marche d'un quadrupède par réseaux de neurones : application aux chiens Sony ». Encadrement 100%.

Février à juillet 2001 : Un stagiaire Ingénieur ENSAM, sur « Évaluation des performances du logiciel DYNAMO pour la simulation dynamique des robots à pattes ». Co-encadrement 70%, Pr. F. B. Ouezdou 30%

Mars à juillet 2001 : Un stagiaire du D.E.A de Robotique de l'Université Paris 6, sur « Parallélisation temps réel de contrôleurs neuronaux adaptatifs. Application à la robotique mobile ». Encadrement 100%.

Mars à juillet 2002 : Un stagiaire du D.E.A d'Intelligence Artificielle de l'Université Paris 6, sur « Synthèse évolutionnaire de contrôleur neuronaux pour le contrôle d'un robot hybride roues/pattes ». Encadrement 100%.

Février à juin 2003 : Un stagiaire Ingénieur (ECAM Lyon), sur « Contrôle neuronal adaptatif en ligne pour compenser l'équilibre dynamique d'un robot bipède lors de son déplacement suivant plusieurs types d'allures humaines contraintes et soumises à des perturbations ». Encadrement 100%.

Mars à juillet 2003 : Un stagiaire du D.E.A de Robotique de l'Université Paris 6, sur « Contrôle neuronal adaptatif pour la commande d'un engin volant soumis à des dysfonctionnements ». Encadrement 100%

- Mars à juillet 2003** : Un stagiaire du D.E.A de Robotique de l'Université Paris 6, sur « Contrôle neuronal adaptatif en ligne pour compenser la perte d'éléments locomoteurs : application à un robot mobile à 4 roues motrices ». Encadrement 100%.
- Mars à juillet 2004** : Un stagiaire du D.E.A Orsay et Ingénieur ISEEE, sur "Etude de l'implantation temps réel d'un réseau neuronal pour la commande d'un robot quadrupède.". Encadrement 100%.
- Février à juin 2005** : Un stagiaire Ingénieur (ECAM Lyon), sur, sur « Étude préliminaire de la commande d'équilibre du robot bipède ROBIAN par réseau de neurones multicouches ». Encadrement 100%.
- Mars à aout 2006** : Une stagiaire du Master ASP Robotique et Traitement du Signal (Nantes), sur « Etude de la commande neuronale d'un robot anguille ». Co-encadrement 75%, Pr. W. Khalil (IRCCyN) 25%.
- Janvier à juin 2007** : Un stagiaire Ingénieur ENSEA, sur « Exhibition d'allures de marches statiques et quasi-dynamiques pour un robot bipède avec et sans prothèse humaine ». Encadrement 100%.
- Février à juin 2008** : Une stagiaire Master CSER UVSQ, sur « Etude du modèle neuronal de Rowat Selverston pour la genèse de rythmes locomoteurs ». Encadrement 100%
- Avril à septembre 2011** : Un stagiaire Master ENS Cachan, sur « Apprentissage et synchronie d'un mouvement rythmique par le robot humanoïde Nao ». Encadrement 70%.

1.6.5 Publications en tant que Maître de Conférences

Articles de revues internationales avec comité de lecture

- [RI1] T. Hoinville, C. Tapia, P. Hénaff, Flexible and Stable Pattern Generation by Evolving Constrained Plastic Neurocontrollers, *Adaptive Behavior journal* ($IF_{2010} = 1.911$), Accepted le 22 fevrier 2011, to be published in 2011.
- [RI2] Amrollah E. and Hénaff P. On the role of sensory feedbacks in Rowat-Selverston CPG to improve robot legged locomotion, *frontiers in Neurobotics*. Vol 4 :113, december 2010. doi : 10.3389/fnbot.2010.00113, 2010
- [RI3] Hénaff, P. ; Scesa, V. ; Ouezdou, F. B. & Bruneau, O. Real time implementation of CTRNN and BPTT algorithm to learn on-line biped robot balance : Experiments on the standing posture *Control Engineering Practice* ($IF_{2010} = 1.943$), Vol 19, pp 89 - 99, 2010
- [RI4] H. Serhan, C. Nasr, P. Hénaff, Muscle Emulation With DC Motor and Neural Networks For Biped Robots, *International Journal of Neural Systems* ($IF_{2010} = 2.98$), Vol. 20, No. 4 , pp 341-353, 2010
- [RI5] T. Hoinville - P. Hénaff -E. Monacelli, Evolving Simplistic Neural Controllers toward Adaptation to Internal Perturbations, *AMSE Journal of Advances in Modelling*, advances C, Vol. 60, N°6, pp 57-71, 2005
- [RI6] R. Thieffry, E. Monacelli, P. Hénaff, S. Delaplace, Technical aid adaptation with robotic, *AMSE Journal of Bioengineering Applied to Handicap Rehabilitation*, pp. 57-69, 2003

Articles de revue nationale avec comité de lecture

- [RN1] Viacheslav Khomenko, Patrick Hénaff, Vladimir Borysenko,, generalized model of articulated electromechanical systems with elasticities in transmissions. *Journal of Technical Electrodynamics*, Institut d'Électrodynamique de l'Académie Nationale des Sciences d'Ukraine, Vol. 1, pages 141-146, septembre 2010
- [RN2] T. Hoinville, P. Hénaff ,S. Delaplace, Etude sur l'intérêt des modèles biologiques de réseaux de neurones pour la synthèse de rythmes locomoteurs adaptatifs, *Journal Européen des Systèmes Automatisés*, RS-JESA, Vol. 41-N°3/4, pp 413-435, 2007

Articles de Conférences Internationales avec comités de lecture et actes

- [CI1] Artem Melnyk, Patrick Hénaff, Sebastien Razakarivony, Volodymyr Borysenko, Philippe Gaussier, Adaptive Behavior Of An Electromechanical Arm Robot In Case Of Physical Interaction With The Human Being, to be published in proceedings of 2011 IEEE/ASME AIM International Conference on Advanced Intelligent Mechatronics, Juillet 2011
- [CI2] J. Nassour, P. Hénaff, F. B. Ouezdou, and G. Cheng. Adaptive Locomotive Behaviors of a Biped Robot : Patterns Generation and Classification, proceeding of 11th International Conference on Simulation of Adaptive behavior, SAB 2010, LNAI 6226, pp 313-324., 2010
- [CI3] John Nassour, Patrick Hénaff, Fethi Ben Ouezdou and Gordon Cheng, Experience-based Learning Mechanism for Neural Controller Adaptation : Application to Walking Biped Robots, Proceedings of 2009 IEEE/RSJ International Conference on Intelligent Robots and Systems IROS 2009 (sél : 900/1650), pp. 2616-2621, 2009
- [CI4] S. Alfayad, F.B. Ouezdou, F. Namoun, O. Bruneau, and P. Hénaff Three DOF Hybrid Mechanism for Humanoid Robotic Application :Modeling, Design and Realization, Proceedings of 2009 IEEE/RSJ International Conference on Intelligent Robots and Systems IROS 2009 (sél : 900/1650), pp 4955-496, 2009
- [CI5] H. Serhan, P. Hénaff, C. Nasr, F. Ouezdou , “State Machine-Based Controller for Walk-Halt-Walk Transitions on a Biped Robot”, Proceedings of 2008 IEEE-RAS International Conference on Humanoid Robots, Humanoids 2008, December 2008, Korea, pp. 535-540, 2008.
- [CI6] Hayssam Serhan, Chaiban Nasr, Patrick Hénaff, F. B Ouezdou " Dynamic Control Strategy of a Biped Inspired from Human Walking” Proceedings of 2008 IEEE/RAS/EMBS International Conference on Biomedical Robotics and Biomechatronics –BioRob 2008 (sél : 115/203), pp. 342-347, Scottsdale, USA, October 2008
- [CI7] Hayssam Serhan, Chaiban Nasr and Patrick Hénaff, F. B Ouezdou " A New Control Strategy for ROBIAN Biped Robot Inspired from Human Walking” Proceedings of 2008 IEEE/RSJ International Conference on Intelligent Robots and Systems IROS (sél : 660/1350), pp 2479-2485, Nice, France, september, 2008
- [CI8] Hayssam Serhan, Chaiban Nasr and Patrick Hénaff, “Designing An Artificial Muscle Based on PID Controller and Tuned by Neural Network with NN Identification of the Plant," Intelligent Engineering Systems Through Artificial Neural Networks, ASME books, ISBN 0791802566 ,Volume 16, pp. 747-752, 2006
- [CI9] Vincent Scesa, Patrick Hénaff, Fathi Ben Ouezdou, Faycal Namoun, Time Window Width Influence On Dynamic BPTT(h) Learning Algorithm Performances : Experimental Study, Proceedings of International Conference on Artificial Neural Networks, ICANN, Septembre 2006 , Lecture Notes in Computer Science, Volume 4131 LNCS-I, pp. 93-102, 2006, Springer-Verlag Berlin Heidelberg, Septembre 2006
- [CI10] Vincent Scesa, Patrick Hénaff, Fathi Ben Ouezdou and Faycal Namoun, On The Analysis Of Sigmoid Time Parameters For Dynamic Truncated BPTT Algorithm, Proceedings of IEEE International Joint Conference on Neural Networks, pp. 4498-4505, Vancouver, Canada, juillet 2006.
- [CI11] Hayssam Serhan, Chaiban Nasr and Patrick Hénaff, Designing a Muscle like System based on PID Controller and Tuned by Neural Network, Proceedings of IEEE International Joint Conference on Neural Networks, pp. 10090-10097, Vancouver, Canada, juillet 2006
- [CI12] V. Scesa, B. Mohamed, P. Hénaff, F.B. Ouezdou, Dynamic recurrent neural network for biped robot equilibrium control : preliminary results, Proceedings of IEEE International Conference on Robotics and Automation, Barcelone, Espagne, pp 4125-4130, Avril 2005
- [CI13] Thierry Hoinville and Patrick Hénaff, Evolving Plastic Neural Controllers stabilized by Homeostatic Mechanisms for Adaptation to a Perturbation, Proceedings of Ninth International Conference on the Simulation and Synthesis of Living Systems (ALIFE9), pp 81-88 Boston, USA, september 2004

- [CI14] Thierry Hoinville and Patrick Hénaff, 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, pp 2624-2629, Sendai, Japan, October 2004
- [CI15] Thieffry R., Monacelli E., Hénaff P., Delaplace S, Configuring sensors by user learning for a locomotion aid interface . Proceedings of IEEE International Conference on Robotics and Automation, pp 1509-1514, Taipei, Mai 2003.
- [CI16] P. Hénaff, O. Chocron, Adaptive Learning Control in Evolutionary Design of Mobile Robots, Proc. of IEEE International Conference on Systems, Man and Cybernetics, pp 345-349. Hammamet Tunisie, octobre 2002
- [CI17] R. Thieffry and E. Monacelli and P. Hénaff and S. Delaplace, Approaches and learning phases for a robotic adaptive deambulation task, Proc. of IEEE International Conference on Systems, Man and Cybernetics, pp 210-215, Hammamet Tunisie, octobre 2002
- [CI18] P. Hénaff, O. Chocron, Evolving mobile robots with learning abilities by neural controller, International Symposium on Measurement and Control In robotics, Bourges, juin 2002

Conférences internationales avec comité de lecture non répertoriées dans les bases de données internationales

- [CIN1] J. Nassour, P. Hénaff, F. B. Ouezdou, and G. Cheng, Bipedal Locomotion Control with Rhythmic Neural Circuits Patrick Hénaff, International Workshop on Bio-Inspired Robots, Nantes, April , 2011.
- [CIN2] Viacheslav Khomenko, Vladimir Borissenko, Patrick Hénaff, generalized model of articulated electromechanical systems with elasticities in transmissions. XI International scientific conference « problems of present-day electrotechnic 2010, Kiev, juin 2010.
- [CIN3] Olivier Bruneau, Chiheb Zaoui, Fathi Ben Ouezdou, Aref Maalej, Samer Alfayed, Patrick Hénaff , Vincent Scesa Dynamic balance of a bipedal robot with torso subjected to 3D external disturbances, Workshop on Humanoid and Legged Robots (HLR), Karlsruhe, Allemagne, 25-26 Septembre, 2006.
- [CIN4] F.B. Ouezdou, R. Sellaouti, A. Konno, B. Mohamed, V. Scesa, P. Hénaff, C. Zaoui, O. Bruneau, ROBIAN Biped Robot : Design and Experiments, Proceedings of HLR Workshop on Humanoid and Legged Robots, Metz, France, October 14-15, 2004
- [CIN5] P. Hénaff, T. Hoinville, V. Scesa, B. Mohamed, F.B. Ouezdou , Biologically inspired control based on Adaptive Neural Networks for Legged Robots, Proceedings of HLR Workshop on Humanoid and Legged Robots , Metz, France , October 14-15, 2004
- [CIN6] V. Scesa – P. Hénaff– B. Mohamed- R. Sellaouti-F. B. Ouezdou . Identification neuronale d'un torse de robot bipède : résultats expérimentaux, Actes de la Conférence Internationale sur la Modélisation Numérique et Applications, Beyrouth, Liban, Novembre 2003,

Conférences et colloques nationaux avec actes

- [CN1] Patrick Hénaff, IRON : Implémentation Robotique de Neuro-contrôleurs adaptatif, Actes des journées Robea/CNRS (Robotiques et entités artificielles), Montpellier, avril 2005
- [CN2] Patrick Hénaff, IRON : Implémentation Robotique de Neuro-contrôleurs adaptatif, Actes des journées Robea/CNRS (Robotiques et entités artificielles), Toulouse, Janvier 2004
- [CN3] R. Thieffry and E. Monacelli and P. Hénaff and S. Delaplace, différentes approches d'apprentissage neuronal pour la conduite adaptative d'un déambulateur, Actes de la conférence Handicap 2002, pages 157-162 ,juin, 2002, Paris

Communications conférences, et séminaires sans actes,

- [S1] J. Nassour, P. Hénaff, F.B. Ouezdou, G. Cheng, Adaptive Locomotive Behaviors of a Biped Robot : Patterns Generation and Classification, Journées Nationales de la recherche er Robotique Humanoïde, Poitiers, 2010

- [S2] Patrick Hénaff, Genèse des mouvements rythmiques et réflexes inspirée de la biologie : application à la locomotion et au contrôle d'équilibre des robots humanoïdes, Journées Nationales de la recherche en Robotique Humanoïde, Montpellier, mars 2007
- [S3] Patrick Hénaff, Contrôle Inspiré de la biologie pour les robots à pattes, Séminaire invité au laboratoire de Biorobotique de l'Université de Marseille, février 2006.
- [S4] Patrick Hénaff, Contrôle Inspiré de la biologie pour les bipèdes, Journée sur la Robotique Humanoïde et les applications médicales, Université de Versailles, novembre 2005.

Information scientifique et vulgarisation : Journaux scientifiques grand public, émission scientifique télévisée

- [M1] Patrick Hénaff, « la marche forcée des robots », les dossiers du magazine La Recherche, l'évolution, comment les espèces s'adaptent, N°27, pp 52, juin 2007.
- [M2] Patrick Hénaff, l'apprentissage des robots, France 2 television, Francois de Closet : Les robots à l'égal de l'homme, mars 2000 (rediffusions : France 5, TV5 : 2001, 2002).

1.6.6 Publications avant le statut de Maître de Conférences

Conférences Internationales avec actes et comités de lecture

1. G. Bourdon, P. Hénaff , Fuzzy and Neural Control for Mobile Robotics Experimentations, Proc. of IEEE Int. Conf. on Knowledge-Based Intelligent Systems, session invitée, pp 659-666, Adelaide, mai 1997,
2. P. Hénaff, S. Delaplace, Using Backpropagation Algorithm for Neural Adaptive Control : Experimental Validation on an Industrial Mobile Robot, Proc. 1996 Int. Symp. CISM-IFTOMM Robots Manipulators Systems, pp 347-354, Udine, juillet 1996,
3. P. Hénaff, S. Delaplace, Backpropagation Algorithm Without Output Desired : Implementation of Neural Control in Mobile Robotics, Proc. of Int. Conf. on Neural Networks and their Applications, Marseille, mars 1996.
4. P. Hénaff, M. Milgram, J. Rabit , Adaptive Neural Control in Mobile Robotics : Experimentation for a Wheeled Cart, Proc. of IEEE Int. Conf. on Systems, Man and Cybernetics, session invitée , pp 1139-1144, San Antonio-USA. Octobre 1994
5. P. Hénaff, M. Milgram, Adaptive Neural Control with Backpropagation Algorithm , Proc. of IEEE Int. Symp. Signal Processing, Robotics and Neural Networks, Villeneuve d'Ascq, Avril 1994.
6. P. Hénaff, M. Milgram, Backpropagation Learning Without Output Desired, Proc. of IEEE Int. Symp. on NonLinear Theory and its Applications (non présenté), Hawaii, 1993.
7. P. Hénaff, H. Schwenk, A Neural Network Approach of the Control of Dynamic Biped Equilibrium, M. Milgram, Proc. of Int. Symp. on Measurement and Control in Robotics, Turin, Novembre 1993

Conférences et colloques nationaux avec actes

1. Patrick Hénaff, Apprentissage de comportements réflexes par réseaux de neurones : Applications à la locomotion Articulée. Actes des Journées des Jeunes Chercheurs en Robotique, I3S UNSA, Sophia Antipolis, mars 1994.

Communications conférences, et séminaires sans actes,

1. Patrick Hénaff, Commande Neuronale en Robotique Mobile, Rencontres S.P.I de l'Université de Cergy Pontoise, Cergy Pontoise, décembre 1995
2. Patrick Hénaff, Using Neural Networks in Robotics, Seminar on Sensor-Technology, université de Cergy Pontoise/Fachschule Hannover, Allemagne, Hannover, avril 1995
3. Patrick Hénaff, Commande Neuronale en Robotique Mobile, Rencontres S.P.I de l'Université de Cergy Pontoise, Cergy Pontoise, décembre 1995

4. Patrick Hénaff, Contrôle adaptatif neuronal de l'équilibre dynamique d'un bipède, Journée CNRS Greco-pôle Sarta, LRP, Vélizy, Février 1994.
5. Patrick Hénaff, Perceptrons Multi-couches pour la commande de robot mobile autonome, Journée CNRS Greco-pôle Sarta, Paris, Avril 1992.

Information scientifique et vulgarisation : Émission Scientifique Télévisée

1. Patrick Hénaff, Réseaux de Neurones et Robotique, Arte , 10ème anniversaire de la Cité des Sciences et de L'Industrie Sous le Règne d'Héphaïstos : Les Robots, ([http ://www.yannminh.org/french/TxtRobotFilm030.html](http://www.yannminh.org/french/TxtRobotFilm030.html)) diffusions en mars 1996, décembre 1996, 1997

Chapitre 2

Synthèse de l'activité d'Enseignement

Depuis ma nomination en 1997 à l'IUT de Cergy Pontoise, je suis responsable des enseignements d'Automatique et d'Automatismes Industriels du département GEII sur le site de Sarcelles. Ce département qui a été créé en 1997 avec l'option Réseaux Locaux Industriels est un département de petite taille puisque la promotion de deuxième année a varié de 20 à 40 étudiants pour se stabiliser à environ 40 étudiants depuis 3 années. L'équipe enseignante comprend maintenant 6 Maîtres de Conférences et 3 professeurs agrégés. Nous intervenons aussi sur le département SRC (Service Réseaux et Communications) voisin pour toutes les matières techniques et les langues. Étant le seul enseignant du département spécialisé dans les domaines de l'automatique et de l'automatisme, j'ai dû créer la totalité des Cours Magistraux, Travaux Dirigés et Travaux Pratiques, et équiper entièrement la salle de TP qui est commune aux deux disciplines. Je dresse ici une synthèse de ce travail en donnant les volumes créés (en heure brute étudiant), ainsi qu'un tableau récapitulatif en heures équivalent TD de mon service par année depuis 2005.

2.1 Enseignement niveau Licence 1

► IUT de Cergy Pontoise, Département GEII, site de Sarcelles

→ depuis 1998 : Contrôle-commande des Systèmes Industriels (Module ARS2 du PPN GEII)

CM (12 h), TD (9 séances de 2 h), TP (7 séances de 4 h), 40 étudiants.

- Analyse de processus et spécification de cahier des charges, cycle de vie d'un automatisme,
- Structure des Automatismes industriels, Analyse
- spécification et modélisation d'une partie commande, représentation normalisée ; Grafset,
- Architecture d'un automate programmable industriel,
- Systèmes d'Entrée/Sorties industriels pour Automate Programmable Industriel (API) : cartes spécialisées, câblage capteurs, pré-actionneurs
- Sensibilisations aux différentes technologies (électrique, pneumatique, hydraulique, ...),
- Interface homme-machine,
- Analyse et critères de choix des technologies de commande industrielle,
- Langages normalisés de programmation IEC 1131-3 et techniques de mise en œuvre,
- Analyse et commande de processus industriel à partir de cahier des charges
- Modes de marche et d'arrêt d'un système automatisé, outil méthode GEMMA,
- Notions élémentaires de progiciels d'entreprise en relation avec l'automatisation

→ depuis 2010 : Même module en Apprentissage (8 étudiants)

► IUT de Cergy Pontoise, Département SRC, site de Sarcelles

→ de 2008 à 2010 : Informatique, Algorithmique :

CM (20 h), TD (10 h), département Service Réseaux et Communications, 65 étudiants

J'ai effectué durant 2 années cet enseignement dans le département SRC (Service Réseaux et Communication) en première année. La particularité de ce module est lié au profil des étudiants de SRC dont une majorité vient des filières non scientifiques du baccalauréat (L, ES, ..). Pour ces étudiants, l'informatique c'est utiliser un ordinateur et non pas le programmer. L'apprentissage de la programmation est très difficile pour eux et tout l'objet du module est de leur faire comprendre les bases : raisonner par l'approche descendante, savoir construire un algorithme simple, savoir l'écrire en C, comprendre la notion de compilateur, de codage, de debugger, de types, de boucles, de tableaux, de fonction...

2.2 Enseignement niveau Licence 2

►IUT de Cergy Pontoise, Département GEII, site de Sarcelles

→Depuis 1998 : Automatique-Asservissement (Module AU3 du PPN GEII) :

CM (18 h) , TD (30 h), TP(18 h), 40 étudiants.

J'ai créé le Cours Magistral (10 séances d'une heure et demie), les séances de TD (12 séances de 2h) et les séances de TP (6 séances de 3 h) ainsi que les documents associés. Chaque année, je prend en charge le Cours Magistral, au moins un groupe de TD et un groupe de TP. Le contenu est celui du module AU3 du PPN GEII. Il concerne la théorie des systèmes linéaires continus dont les points principaux sont les suivants :

- Caractéristiques temporelles et fréquentielles des systèmes, systèmes intégrateurs,
- systèmes bouclés (Chaîne directe et chaîne de retour), Calcul de la fonction de transfert en boucle fermée, stabilité,
- Étude de la précision statique,
- Méthodes d'identification expérimentale de systèmes,
- Rôle et structure générale des correcteurs (P, PI, PD, PID), synthèse des correcteurs (marge de phase,...),
- Numérisation des correcteurs analogiques,
- Régulation industrielle : normes et schémas normalisés, configuration d'un régulateur. autoréglage, régulation cascade,
- Validation de correcteur par CAO et/ou par automate et/ou par programmation de microcontrôleur.

Les séances de Travaux Pratiques s'appuient sur des simulations sous ACSYDE, des expériences correspondantes sur des bancs de moteurs à courant continu (asservissement de position et de vitesse) ainsi que de la programmation de PID sur Automate Programmables Industriels.

→Depuis 1998 : Réseaux de terrain (Module ARS21 du PPN GEII)

CM (6 h), TD (4 séances de 2 h), TP (4 séances de 4 h), 40 étudiants.

- Les réseaux de terrain : besoins spécifiques, évolution et normalisation (IEC 61158),
- La couche Physique dans les réseaux de terrain : câblage, normes
- Modèles de communication dans les réseaux de terrain :Client-Serveur, Producteur-Distributeur-Utilisateur, Maître-Esclave, base de données globale,
- Étude de réseaux de terrain utilisés dans les principaux secteurs : automatisation (ASI, MODBUS, PROFIBUS, Ethernet Industriel ...), domotique et immotique (Bus KNX...),
- Distribution du contrôle :
- commande de processus par réseaux de terrain, applications.

→depuis 1998 : Supervision et contrôle de procédés industriels (Module ARS4 du PPN GEII)

CM (6 h), TD (3 séances de 2 h), TP (3 séances de 4 h), 40 étudiants.

- Place et rôle de la supervision dans l'entreprise, les fonctions de la supervision : conduite, maintenance, qualité, gestion de production,

- Procédés et interface homme machine, notion d'ergonomie graphique,
- Normes, SCADA, MES, ERP
- Étude, configuration et mise en œuvre d'un progiciel de supervision industrielle,
- Techniques d'interface logicielle pour l'acquisition et le partage d'informations,
- Couplage bases de données- supervision,
- Notion de supervision à distance par Internet, sécurité

Pour ces 4 modules, j'ai créé les Cours Magistraux, TD et TP. J'ai équipé totalement une salle de travaux pratiques :

- 9 stations de simulation sous ACSYDE
- 9 bancs d'asservissement position/vitesse de moteur à courant continu
- 9 Automates Programmables Industriels SIEMENS (S7-300) en réseau (MPI, PROFIBUS-DP, ASI), de modules TOR et analogiques, coupleurs ETHERNET, ASI, pupitre IHM,
- Réseaux Locaux Industriels (réseaux de terrain) : ASI, PROFIBUS, MPI, Industrial Ethernet,
- logiciels de programmation : Step7 , PID-Sim, PLC-Sim,
- Systèmes de supervision : WinCC, Genesis 32 , Serveur OPC
- 5 parties opératives industrielles électro-pneumatiques FESTO : capteurs TOR (optiques, capacitifs, magnétiques, inductifs, illustration du cours de Physique) et analogiques (potentiométriques),
- 1 poste de régulation de niveau d'eau : 1er et 2nd ordre, retard pur, régulation TOR, numérique ou analogique.
- Réseaux domotique et bâtiments Intelligents : réseau KNX, sous réseau DALI, supervision IHM ETS, station de commande IP, station météo, volets roulants, horodateur

→de 2002 à 2008 : Systèmes temps réels (Module II du PPN GEII) :

CM (15 h), TP (4 séances de 3 hh), 40 étudiants.

Le contenu de ce module concernait les systèmes d'exploitation temps réel. Je faisais le lien entre les systèmes autonomes (robots) et les architectures d'automatismes industrielles (Réseau de terrain). 4 séances de TP/TD sous QNX étaient consacrées à ce cours.

- Vue fonctionnelle d'une application structurée en tâches parallèles coopérantes,
- Les mécanismes de coopération (synchronisation, communication, exclusion mutuelle),
- Évocation de la nécessité d'un ordonnancement, les attributs des tâches (priorité statique),
- Les services d'un noyau exécutif temps réel, -
- Exemples d'applications simples.

2.3 Enseignement niveau Master 1

►ENSEA Cergy-Pontoise

→Depuis 2009 : j'interviens dans le module optionnel « initiation aux sciences cognitives », niveau ingénieurs 4^{ème} année (10 étudiants ingénieurs en 2010) où j'effectue un séminaire de 9 h sur la commande des robots par réseaux de neurones.

►École Polytechnique, Palaiseau

→De 2002 à 2008 : j'ai encadré de 4 à 8 étudiants de 2^{ème} année de l'X qui travaillaient ensemble sur un projet scientifique concret dans le cadre des Projets Scientifiques Collectifs. Ce tutorat consistait en des réunions de 2 heures soit à l'école, soit au laboratoire LISV. J'ai d'autre part, participé au jury pour les soutenances et à l'évaluation du rapport de projet (Pr référent à l'X, M. Patrick Huère) :

- 2002-2006 : Réalisation d'un robot mobile (mécanique, électronique, informatique), coupe E=M6.
- 2008 : projet Zéro-G « SpaceWalk : dispositif d'aide au contrôle de la posture pour les spationautes». Prix du meilleur PSC 2008 (1er ex-æquo) de l'École Polytechnique. Co-encadrement avec E. Monacelli du LISV. Sélection par le CNES pour un essai en Airbus Zero-G (http://www.cnes-multimedia.fr/dossiers/Journal_de_lespace2008/)

2.4 Enseignement niveau Master 2

Depuis 2001, j'ai enseigné régulièrement en Master 2 des modules concernant mon activité de recherche. Cette année je participe au démarrage du Master en bâtiment intelligent de l'UCP où j'enseigne le module automatismes et immotique.

► Université de Cergy-Pontoise

→2010 : Master recherche SIC-RIA (Robotique et Intelligence Artificielle), 8 étudiants

Je suis responsable de l'UE complémentaire « Robotique et Commande Bio-Inspirée » (20 de CM, 8 étudiants en 2010). Le but de ce cours est d'acquérir d'une part les éléments essentiels à la compréhension des modèles mécaniques utilisés classiquement en robotique, et d'autre part de découvrir les méthodes de commande de robots inspirées de la biologie. Le cours se situe dans le cadre de la robotique humanoïde en se focalisant sur les tâches de manipulation et de locomotion. Une première partie s'intéresse aux notions de robotique en général, une seconde à l'automatique linéaire et à la commande des robots, une troisième plus conséquente concerne la commande bio-inspirée.

→2010 : Master Professionnel Batiments Intelligents, 15 étudiants.

20 heures de CM, Module « Automatismes et immotique » en automatismes industriel et bâtiments intelligents. Ce master est adossé aux départements GEII et Génie Civil de l'UCP. J'ai participé à l'élaboration du programme et à la définition des contenus liés aux Réseaux de terrains dans les bâtiments.

► Université de Versailles St Quentin : Master Recherche CSER, 12 à 15 étudiants.

→ de 2004 à 2008 : 12 heures de CM, module « Contrôle Bio-Inspiré pour la robotique »

► Université Paris 6, DEA RESIN (robotique), 15 étudiants.

→ de 2001 à 2003 : 12 heures de CM, module « Mise en œuvre de contrôleurs neuronaux pour la robotique ».

2.5 Synthèse des services depuis 2000 :

Je résume dans le tableau ci-dessous les volumes horaires effectués chaque année depuis 2005 en heures brutes devant étudiants, et le total en équivalent Travux Dirigés.

Années	00-01	01-02	02-03	03-04	04-05	05-06	06-07	07-08	08-09	09-10	10-11
CM	28,5	39	36	39	61,5	59,5	CRCT	41	53	64,5	72
TD	157	128	110	41,5	70,5	53,5	CRCT	67	91,5	85,5	137,5
TP	90	152	118	150	172	116	CRCT	96	67	100	83
Total	260	287	243	200	277	220	CRCT	192	215	282	328

TABLE 2.1 – Tableau synthétique des heures enseignées depuis 2000

2.6 Enseignements en tant qu'enseignant contractuel (1990-1997)

► 1990-1993 : Université Paris 7, moniteur

→ UFR de Physique

– Électronique Analogique : TD, TP

► **1993-1996 : IUT de Cergy Pontoise (PRAG contracruel et ATER)**

→ *Département GEII*

- Automatique-Asservissement : création et enseignement des CM, TD, TP, (2^{ème} année)
- Automatismes Industrielle : création et enseignement des TP , (2^{ème} année)
- Robotique mobile : Projet long, (2^{ème} année)

► **1996-1997 : IUT de Vélizy (PRAG contracruel)**

→ *Département GEII (384HTD)*

- Électronique Analogique et numérique : TD, TP (1^{ère} année)
- Électrotechnique, Electronique de puissance : TP (2^{ème} année)

2.7 Synthèse des responsabilités administratives liées à l'enseignement

► **1995 -1996 : Directeur des Études en IUT**

J'ai eu cette fonction administrative durant l'année 95/96 au département GEII de l'IUT de Cergy-Pontoise. J'organisais des réunions mensuelles avec les délégués des groupes d'étudiants afin d'être à leur écoute et de les informer sur les évènements importants : modifications dans l'emploi du temps, arrivée d'un nouvel enseignant, stages....Je devais assurer une bonne coordination entre les enseignants, notamment sur l'évaluation des travaux pratiques et des travaux de réalisation. D'autre part, j'avais pour rôle de collecter les notes des étudiants et d'établir régulièrement un classement général. Enfin, pendant le second semestre, une part importante de mon travail a consisté, en concertation avec les autres enseignants à m'occuper des dossiers, de poursuite d'études des étudiants.

► **1997-2000 : Chef de département d'IUT**

→ *Création du Département GEII (option RLI) de l'IUT de Cergy-Pontoise sur le site délocalisé de Sarcelles.*

Lors de ma nomination, en septembre 1997, en tant que Maître de Conférences à l'I.U.T de Cergy-Pontoise, et compte tenu de mon expérience d'enseignement dans cet I.U.T entre 1994 et 1995, le Directeur m'a proposé le poste de chef de département du département Génie Electrique et Informatique Industrielle créé à la même date sur le Site de Sarcelles. Ce département est le dernier de cette spécialité créé en France. A mon arrivée, l'ensemble des locaux était vide. Un budget d'équipement conséquent m'a été attribué (380 k€). Le département a ouvert avec 1 enseignant PRAG, 2 Maîtres de Conférences dont moi, une secrétaire et un technicien. Durant trois années, j'ai assumé pleinement la charge de faire de ce département situé en pleine zone urbaine déforarisée un département d'une qualité exemplaire tant sur l'encadrement et l'innovation pédagogique, sur l'ouverture aux étudiants, sur la qualité technologique et scientifique sans cesse mise à jour, que sur le niveau de la formation. Durant ces trois années, les responsabilités principales que j'ai du assumer sont les suivantes :

- Gestion de budget d'équipement lié à la création et du budget initial de fonctionnement
- Equipement pédagogique complet du Département
- Organisation pédagogique liée au PPN.
- Formation des personnels IATOS au fonctionnement d'un IUT
- Formation des jeunes enseignants aux diverses responsabilités (direction des études, des stages, des emplois du temps)
- Recrutement et gestion de vacataires
- Relations extérieures
- Organisation et pilotage du Conseil de Département et des jurys
- Conseils d'IUT

► Depuis 2008 : Responsable des stages en IUT

En tant que responsable des stages au département GEII de Sarcelles depuis 3 années. Mon rôle consiste à suivre les étudiants de 2^{ème} année (38 étudiants en 2011) dans leur démarche de recherche de stage, dès le mois de septembre de chaque année : rédaction des CV, conseil à la recherche de stage (quel domaine ? quelle type d'entreprise ? par quel canal ?), formation à l'entretien (avec un DRH d'une entreprise), vérification de la cohérence des sujets de stage proposés par les entreprises, établissement des conventions de stage avec le secrétariat pédagogique, suivi des candidature pour un stage à l'étranger (en collaboration avec ma collègue chargée des relations internationales), préparation des soutenances et jury de stage. . .

► 2011-2014 : Création d'un Master Franco/Ukrainien

J'ai créé un parcours spécifique au Master de l'UCP « systèmes intelligents et Communicants, parcours Robotique et Intelligence artificielle », adapté pour les étudiants francophones de l'Université Nationale Technique de Donetsk en Ukraine. L'objectif principal de ce master en partenariat Franco-Ukrainien est de transmettre aux étudiants du magistère de la chaire Commande Electrique et Automatisation des Installations Industrielles de l'UNTD les connaissances les plus récentes dans le domaine de l'intelligence artificielle appliquée aux systèmes robotisés (apprentissage, systèmes multi agents, réseaux de neurones, vision artificielle. . .). Ce cursus vient compléter leur formation actuelle plutôt tournée vers les méthodes de l'ingénieur en leur apportant des outils et des méthodes nouvelles issues de la recherche en Intelligence Artificielle. Ce projet est soutenu financièrement par l'ambassade de France pour la période 2011/2014.

Deuxième partie

Activité scientifique

Chapitre 3

Introduction

Je résume dans ce chapitre introductif les idées principales qui sont à la base de mon parcours et qui ont construit ma réflexion jusqu'à aujourd'hui. On comprendra ainsi mieux le cheminement que j'ai suivi de la commande par apprentissage « basique » à la commande bio-inspirée.

3.1 Résumé du parcours scientifique

De 2002 à 2008, j'ai été rattaché au laboratoire LISV de l'Université de Versailles Saint Quentin.¹ J'intervenais dans l'axe Robotique Humanoïde dont les objectifs scientifiques concernent la conception, l'évaluation et la maîtrise de systèmes d'assistance (ou de suppléance) pour l'homme. Cet axe s'intéresse à l'analyse de deux tâches essentielles dans la vie quotidienne de l'homme, la locomotion et la manipulation. Ces travaux sont orientés vers les modélisations et simulations biomimétiques, la commande dynamique et *la commande bio-inspirée que j'animais*.

Mon activité de recherche dans le domaine de la commande bio-inspirée a été initiée en 2002 en collaboration avec le LIP6 (équipe AnimatLab de Jean Arcady Meyer). Durant la période 2002-2008, ces recherches se sont déroulées au sein de 2 projets scientifiques.

Le premier projet, IRON (Implémentation Robotique de Neurocontrôleurs adaptatifs), que j'ai dirigé scientifiquement a été financé par le programme ROBEA du CNRS de 2002 à 2005. Ce projet mené en collaboration avec J. A. Meyer (LIP6, AnimatLab), puis avec Daniel Cattaert (Centre de Neurosciences Intégratives et Cognitives, Equipe Comportement et Réseaux Neuronaux, Université de Bordeaux 1), a permis la soutenance de 2 thèses menées sous ma direction scientifique. La première en octobre 2006 (Vincent Scesa) et la seconde en octobre 2007 (Thierry Hoinville).

Le second projet, PHEMA (Plateforme Humanoïde pour l'Etude de la MArche humaine avec et sans déficience), dont j'ai assuré la codirection scientifique avec F. B. Ouezdou jusqu'en 2008, a été financé par le programme ANR blanc de 2005 à 2009. Ce projet avait pour but de construire au LISV une nouvelle plate-forme bipède dédiée à l'analyse de l'appareil locomoteur de l'être humain. PHEMA a été mené conjointement avec l'IRCCyN (Nantes), le CNIC (Bordeaux) et l'Hôpital Raymond Poincaré de Garches (Pr. B. Bussel). Dans ce projet, une thèse a été soutenue sous ma direction scientifique en juillet 2009 (Hayssam Serhan), une autre a débuté en 2007 (John Nassour).

En 2008, j'ai intégré au laboratoire ETIS de l'Université de Cergy-Pontoise l'équipe neurocybernétique dirigée par Philippe Gaussier. Cette équipe est reconnue nationalement et internationalement pour ses travaux dans les domaines de l'apprentissage et de l'imitation, plus généralement sur la modélisation des processus cognitifs validés sur des systèmes robotisés, robots mobiles ou robots humanoïdes. Nous collaborons étroitement avec le LISV sur des projets de robotique humanoïde dans lesquels je suis impliqué pour la partie motricité. Mon expérience apporte en effet à ETIS des connaissances axées sur l'apprentissage de la commande bas niveau, l'automatique et sur la genèse de mouvements rythmiques. Je m'intéresse plus particulièrement à l'apprentissage des

1. Le LISV a été créé en janvier 2006 suite à une fusion de trois laboratoires de l'Université de Versailles : le Laboratoire d'Instrumentation et de Relation Individus Systèmes (LIRIS) auquel j'appartenais, le Laboratoire de Robotique de Versailles (LRV), et le Laboratoire d'Études Mécaniques des Assemblages (LEMA). Le laboratoire LIRIS était lui-même issu du démantèlement du Laboratoire de Robotique de Paris en janvier 2002. Mon activité de recherche dans ces laboratoires faisait l'objet d'une convention d'accueil entre l'UCP et L'UVSQ.

contrôleurs bio-inspirés lors de l'interaction physique entre un robot et son environnement lorsque cette interaction est rythmique.

Depuis 2008, mon activité se concrétise à ETIS par l'encadrement de 2 nouveaux doctorants dans le cadre d'une collaboration avec l'Université Nationale Technique de Donetsk en Ukraine. Ces doctorants sont inscrits en cotutelle de thèse. L'un, rattaché au LISV (thèse commencée en 2008), travaille sur le contrôle de la compliance et de la rythmicité des membres inférieurs des robots humanoïdes en interaction avec le sol. L'autre doctorant (thèse commencée en 2009) est rattaché à ETIS et travaille sur le contrôle de la rythmicité des membres supérieurs en interaction avec l'être humain. Comme on le verra dans la conclusion générale, le contrôle et la modélisation de la rythmicité globale des robots humanoïdes en interaction avec leur environnement est l'un des enjeux de la robotique actuelle.

3.2 En commençant par l'apprentissage en ligne de commandes motrices pour un robot mobile non-holonome

Le travail que j'ai mené durant ma thèse concernait l'apprentissage en ligne de contrôleurs neuronaux par l'algorithme de rétropropagation du gradient, pour le pilotage de robots mobile et le contrôle d'équilibre de robot bipède. Le titre exact était « Mises en Œuvre de Commandes Neuronales Par Rétro-Propagation Indirecte : Applications à la Robotique Mobile ». L'originalité de mon travail, soulignée alors par le jury, était d'une part l'utilisation de tels algorithmes pour **l'apprentissage moteur en ligne** sur des robots mobiles réels, et d'autre part la démonstration (en simulation) que l'apprentissage par réseaux de neurones permettait **l'acquisition d'un contrôle réflexe** d'équilibre pour les robots à pattes. Durant les deux années suivantes, j'ai travaillé sur le couplage de ces deux comportements : apprentissage d'une loi motrice et apprentissage d'un réflexe d'évitement d'obstacles d'une **boucle sensori-motrice**. Cette approche a été validée expérimentalement sur un robot mobile.

L'apprentissage en ligne sur des robots réel avait pour objectif de démontrer qu'on pouvait apprendre une loi de commande motrice par des réseaux de neurones sans base d'apprentissage initiale, c'est-à-dire sans connaître la loi de commande désirée. Les expérimentations ont été menées pour la thèse en 1994 sur un prototype de robot du Laboratoire de Robotique de Paris, puis en 1997 sur un robot industriel de près de 100kg pour des vitesses atteignant presque 1,5 m/s (robot ROBUTER). Les modèles neuronaux utilisés alors étaient statiques, c'est-à-dire des perceptrons multicouches pour la topologie et le modèle de Mac-Culloch et Pitts (McCulloch et Pitts, 1943) pour les neurones.

Les résultats ont montré qu'un critère mathématique exprimant une configuration du robot mobile $[X, Y, \theta]$ à atteindre dans le plan, permettait au réseau de neurones d'apprendre, par rétropropagation du gradient de ce critère, les commandes à appliquer à chaque instant aux deux moteurs pour conduire le robot à une configuration désirée (voir figures 3.1 et 3.2). Le réseau, aléatoire au départ, apprenait tout en pilotant le robot : chaque itération d'apprentissage provoquait une nouvelle commande qui permettait au contrôleur neuronal d'explorer l'espace des états du système dynamique qu'il commandait. Petit à petit, la commande convergeait vers une commande de plus en plus performante et finalement, après 3 ou 4 « leçons d'apprentissage » le réseau avait appris à commander le robot quelle que soit sa position initiale dans le plan, y compris celles jamais observées pendant l'apprentissage (figure 3.3).

Ensuite, j'ai pu intégrer à ce travail d'apprentissage en ligne de la boucle sensori-motrice du robot mobile des retours sensoriels supplémentaires (capteurs à ultrasons) afin que le robot apprenne en même temps à éviter un obstacle. Cet apprentissage se faisait par interaction sensorielle avec le robot : en sollicitant les capteurs à ultrasons durant l'apprentissage de la commande, je pouvais faire reculer le robot alors qu'il n'était qu'à quelques centimètres de la position désirée (figure 3.6).

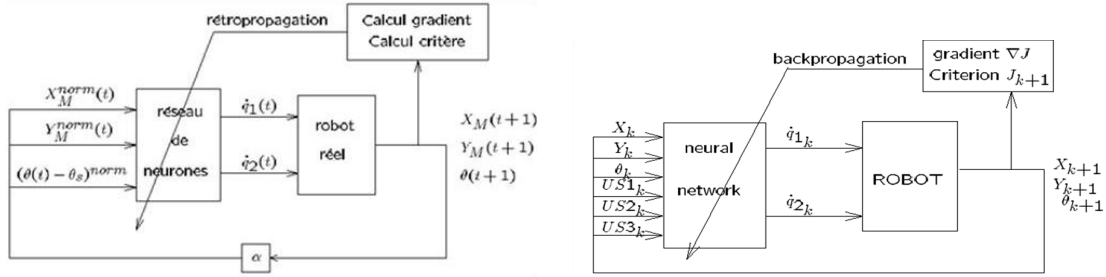


FIGURE 3.1 – Schémas d'apprentissage en-ligne pour la commande du robot mobile. A gauche le réflexe d'évitement d'obstacle n'est pas à apprendre et l'entrée est constituée des erreurs cartésiennes et angulaires normalisées, la sortie des consignes en vitesses à appliquer aux moteurs. A droite le réflexe d'évitement d'obstacle est à apprendre et l'entrée est constituée en plus des retours sensoriels fournis par les capteurs à ultrasons.

3.2.1 Exemple de résultats expérimentaux

Apprentissage de loi de commande sans réflexe d'évitement d'obstacle :

Dans cet exemple (Henaff et Delaplace, 1996), la fonction de coût (équation 3.1) qui exprime l'objectif à atteindre en respectant une contrainte Θ_s sur la stratégie de manœuvre, était la suivante :

$$J = \alpha_1 X^2(t+1) + \alpha_2 Y^2(t+1) + \alpha_3 [\Theta(t+1) - \Theta_s]^2 \quad (3.1)$$

La figure 3.2 compare deux leçons d'apprentissage successives. On peut voir que le deuxième apprentissage engendre une trajectoire cartésienne du robot vers la cible bien plus directe et plus rapide (le temps de convergence de l'apprentissage et de la commande est divisé par 4)

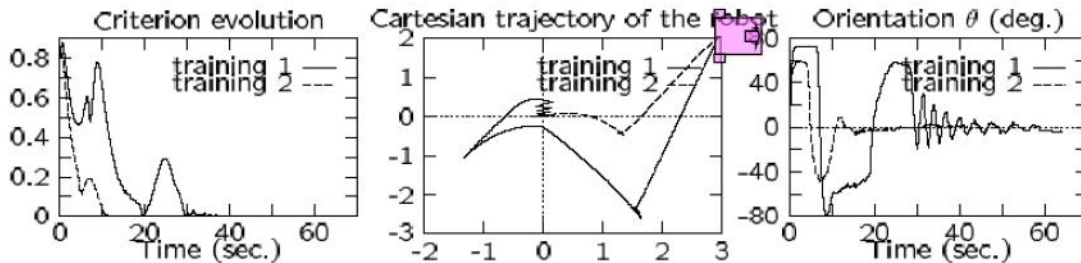


FIGURE 3.2 – Evolution du critère et des trajectoires du robot mobile (représenté à l'échelle en violet) pendant les deux premiers apprentissages (Henaff et Delaplace, 1996).

Après un total de 3 leçons d'apprentissage, on constate que le contrôleur neuronal a acquis une très bonne connaissance sur le comportement dynamique du robot, car il peut le commander lorsque la boucle d'apprentissage est ouverte. Le réseau de neurones pilote alors le robot vers le centre de la pièce à partir d'une position jamais observée précédemment, il y a généralisation de la commande (figure 3.3).

Apprentissage de loi de commande avec réflexe d'évitement d'obstacle :

L'expérience précédente a été étendue en intégrant dans la boucle de commande les retours sensoriels fournis par 3 capteurs à ultrason situés, pour l'un, en façade du robot et, pour les deux autres, sur ses cotés. L'objectif de l'expérience consistait à démontrer que le contrôleur neuronal pouvait résoudre un dilemme en apprenant à la fois la commande précédente et un réflexe d'éloignement lorsqu'un obstacle est présent devant l'un des capteurs.

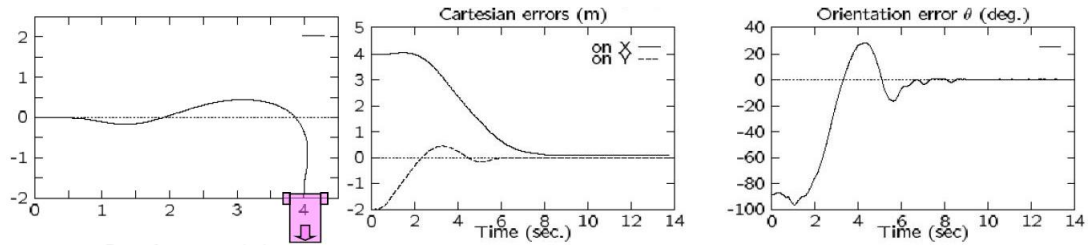


FIGURE 3.3 – Exemple de contrôle obtenu après apprentissage (le réseau de neurones commande le robot mais n’apprend plus), (Henaff et Delaplace, 1996).

La fonction de coût est modifiée (équation 3.2) afin de prendre en compte cette contrainte de réflexe par rapport à la présence d’un obstacle : rejoindre la cible mais rester à une distance suffisante de l’obstacle, appelée « distance de sécurité » .

$$J = \alpha_1 X^2(t+1) + \alpha_2 Y^2(t+1) + \alpha_3 [\Theta(t+1) - \Theta_s]^2 + \alpha_4 \sum_{j=0}^3 \beta_j e^{-(d_j - d_s)} \quad (3.2)$$

Durant les deux premières phases d’apprentissage, il n’y a pas d’obstacle sur la trajectoire du robot. Le problème à résoudre par le contrôleur neuronal est donc le même que précédemment, sauf que la couche d’entrée contient 3 neurones supplémentaires correspondant aux 3 capteurs à ultrason et que le contrôleur neuronal doit tenir compte d’une contrainte dans la loi de commande. Pendant l’apprentissage, l’exploration du comportement du robot par le réseau est plus lente car le nombre de paramètres du contrôleur a augmenté, ce qui se traduit par des trajectoires « plus hasardeuses » pour les deux premiers apprentissages (figure 3.4). Pour la troisième leçon, un obstacle est présent sur la position finale. Le contrôleur neuronal doit donc apprendre à résoudre le dilemme entre rejoindre cette position désirée et rester à une distance de sécurité d_s de l’obstacle située à cette position (figures 3.5 et 3.6).

Après l’apprentissage, les expériences ont montré que le contrôleur neuronal a acquis une loi de commande permettant de commander le robot et que, lorsque un obstacle se présente, il peut modifier cette loi de commande pour s’éloigner de cet obstacle (figure 3.5).

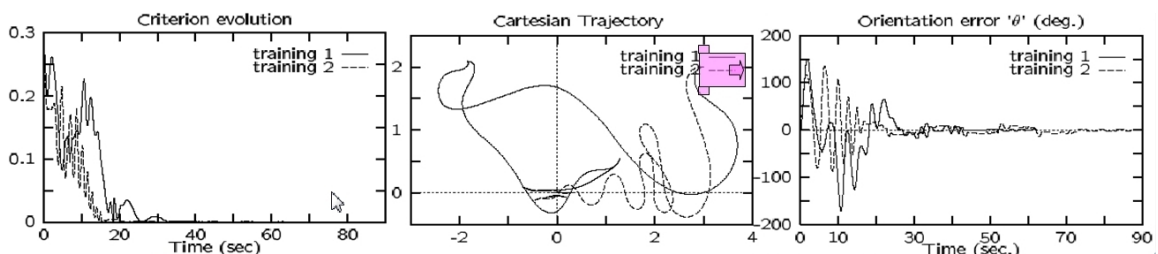


FIGURE 3.4 – Evolution du critère et des trajectoires du robot mobile pendant les deux premiers apprentissages sans présence d’obstacles. La trajectoire « hasardeuse » du robot provient de la phase d’exploration du réseau de neurones qui « découvre » peu à peu le comportement cinématique et dynamique du robot.(Bourdon et Henaff, 1997)

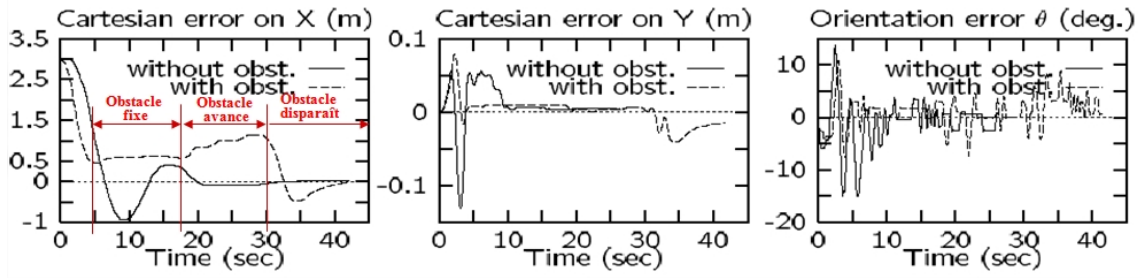


FIGURE 3.5 – Apprentissage du réflexe d'évitement d'obstacles par interaction sensorielle. L'obstacle est situé sur le but. le robot s'arrête à la distance de sécurité $d_s = 50\text{cm}$. Quand l'obstacle se déplace, le contrôleur neuronal maintient la contrainte entre rejoindre le but et rester à une distance de 50cm de l'obstacle, (Bourdon et Henaff, 1997).



FIGURE 3.6 – Apprentissage par interaction sensorielle avec le Robuter en sollicitant ses capteurs à ultrasons (Bourdon et Henaff, 1997).

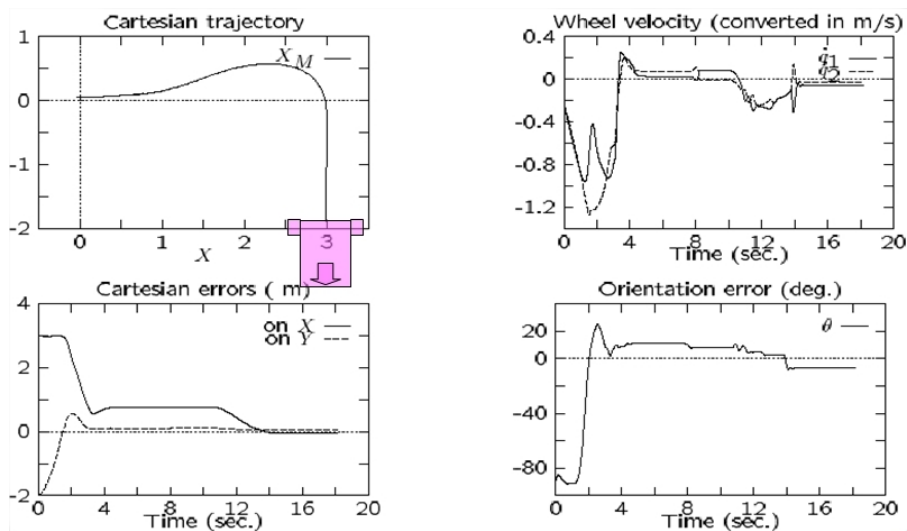


FIGURE 3.7 – Mise en évidence du réflexe d'évitement d'obstacle après l'apprentissage par interaction. Le robot s'arrête à la distance $d_s = 50\text{cm}$ de l'obstacle entre $t = 4\text{s}$ et $t = 11\text{s}$, et rejoint le but lorsque l'obstacle disparaît à $t = 11\text{s}$. (Bourdon et Henaff, 1997).

3.2.2 Conclusion

Les résultats obtenus lors de ces expériences d'apprentissage en ligne ont démontré qu'un contrôleur biologiquement inspiré (même si le perceptron multicouches et l'algorithme de rétropropagation sont de nos jours considérés comme peu bio-inspirés) pouvait apprendre, sans aucune connaissance a priori sur le comportement du système, à fusionner une loi de commande globale et un comportement réflexe. Un tel contrôleur peut donc changer de phase et « basculer » dans un mode de commande ou dans un mode réflexe ou même maintenir un équilibre entre ces deux modes. Il était donc possible d'apprendre le couplage dynamique entre le robot en interaction avec son environnement et le contrôleur neuronal percevant lui-même cet environnement et l'état du robot.

Appliquer cette approche pour le contrôle de la locomotion des systèmes marcheurs et le contrôle de leur posture devenait alors intéressant car il ouvrait de nouvelles possibilités d'autonomie par l'apprentissage pour les robots humanoïdes. En effet, l'existence de ce couplage dynamique étant démontré dans le système nerveux moteur de certains animaux, le passage à l'étude des contrôleurs bio-inspirés pour le contrôle des robots marcheurs devint alors mon objectif scientifique.

3.3 Pourquoi et comment s'inspirer de la nature ?

L'évolution de la technologie nous permet maintenant de construire des robots dont la morphologie est inspirée par l'animal ou par l'homme. Ainsi les robots animaloïdes ou humanoïdes les plus récents sont des systèmes technologiques très complexes, d'un niveau d'intégration mécanique et électronique extrêmement élevé. Ils sont dotés d'organes perceptifs complets (visuels, sonores, tactiles) qui leur permettent a priori d'interagir avec l'homme et de se mouvoir dans un environnement que l'être humain a dû bâtir pour sa propre existence. Nous construisons par ailleurs, pour des défis scientifiques ou technologiques, des machines robotisées à pattes ou à roues dont l'autonomie est un atout car elles interviennent dans un environnement qui nous est hostile ou distant (exploration spatiale, sous marine, sauvetage. . .). Le robot androïde ressemble donc à l'homme ou à l'animal par son allure et son comportement et par l'environnement dans lequel il doit évoluer. Le robot non androïde doit réaliser des tâches que l'homme ne peut réaliser. Ces deux grandes classes de machines ont pour point commun d'être nécessairement dotées d'une « autonomie globale », c'est-à-dire énergétique et décisionnelle. Les solutions apportées jusqu'ici par les ingénieurs et les chercheurs font en sorte que cette autonomie globale se rapproche de plus en plus de celle de l'homme ou de l'animal. Cependant, encore peu de solutions existent pour que l'autonomie des robots soit à un stade où ils peuvent s'adapter à des dysfonctionnements externes et internes, lents ou brusques : glissements au sol, apparition d'usures ou de jeux, assouplissement des liaisons rigides, inhibition ou perte d'organes locomoteurs, changement de morphologie ou de la dynamique du robot, baisse prématurée des ressources énergétiques.

Dans la nature, un point essentiel dans l'autonomie des espèces animales (y compris l'homme) concerne leur capacité à s'adapter à des événements imprévus dont les échelles temps peuvent être très différentes : modification importante, lente ou brutale de l'environnement, dommage soudain ou progressif de certains organes sensoriels ou moteurs, (organes musculaires ou squelettiques) dus à des agressions ou des dégradations. . . Souvent l'animal réussit à s'adapter à ces événements avec plus ou moins de temps, avec plus ou moins de réussite. Ainsi certains animaux peuvent survivre avec une ou plusieurs pattes inertes ou amputées, mais en boitant et en se déplaçant moins vite. Un être humain peut marcher même si son genou est bloqué soudainement ou non, ou s'il est doté d'une prothèse de jambe. De nombreux exemples peuvent être encore cités tant sur les dysfonctionnements des organes moteurs, que de la perception auditive ou visuelle, ou encore d'autres retours sensoriels. La survie dans un environnement imprévisible et hostile est donc un but pour toutes les espèces vivantes de la nature, et un des moyens pour y arriver est de retrouver si possible la performance initiale, mais souvent la solution consiste à survivre en maintenant des performances moyennes acceptables.

Rendre les robots adaptatifs au sens où on peut l'observer dans la nature est donc un objectif scientifique majeur pour l'avenir (Ijspeert, 2008). Pour y arriver, une démarche intéressante consiste à développer des procédures d'adaptation inspirées du vivant pour les intégrer dans les architectures de contrôle ou pour en tenir compte dans la conception. Face à des situations dangereuses comme celles évoquées plus haut, les mécanismes adaptatifs que nous devons inventer pour les robots du

futur pourront intervenir à tous les niveaux de l'architecture de contrôle du robot, mais aussi sur sa morphologie par la possibilité d'une adaptation morphologique comme dans le cas des robots reconfigurables.

L'approche Animat, contraction d' « animal artificiel », et initiée en France par J-A Meyer et A. Guillot s'inspire de cette règle. « *Cette démarche scientifique en robotique a pour objectif de concevoir des systèmes artificiels simulés ou réels inspirés des animaux, s'adaptant de façon autonome dans un environnement qui n'est pas toujours prévisible par l'homme. L'animat aura ainsi à acquérir empiriquement ses capacités adaptatives en appréhendant le monde par ses capteurs et ses actionneurs avec une intervention humaine minimale. Son architecture de contrôle devra être capable, si besoin est, de modifier sa propre organisation, par des processus inspirés de la biologie comme le développement, l'apprentissage ou l'évolution* » (Guillot et Meyer, 2007; Meyer *et al.*, 2002). Dans l'approche animat, le robot ne doit donc pas être seulement adapté, mais il doit être adaptatif. L'évolution d'un robot autonome dans un milieu ouvert impose la gestion de schémas de contrôle robustes vis-à-vis des aléas dus, non seulement à l'environnement, mais aussi aux ressources matérielles propres au robot. L'existence de pannes ou de situations conflictuelles peut conduire à un blocage de celui-ci et, par conséquent, une mise en cause de la réussite de sa mission. Il faut donc proposer des modes alternatifs de fonctionnement permettant au robot de sortir de cette situation de blocage afin qu'il continue sa mission même avec des objectifs réduits, ou qu'il puisse rejoindre en mode dégradé sa base de maintenance, en un mot qu'il soit adaptatif.

3.4 La complexité des robots marcheurs

Le contrôle de la posture et la production d'activités motrices rythmiques sont deux activités intimement liées pour obtenir une locomotion stable et fluide chez l'homme et l'animal. Les robots marcheurs sont des machines intéressantes car elles ont a priori la possibilité de se déplacer sur n'importe quel type de terrain encombré ou non comme la plupart des animaux à pattes. Ce sont des systèmes poly-articulés dont la complexité mécanique induit de nombreuses difficultés dans la mise au point d'algorithmes de commande efficaces, c'est-à-dire exploitant pleinement le potentiel locomoteur de la machine. En effet, quelle que soit l'approche utilisée (commandes fondées sur un modèle dynamique, commandes fondées sur un modèle biologique), on se heurte à la redondance cinématique de ces robots qui, lorsqu'ils marchent, engendre des chaînes cinématiques alternativement ouvertes et fermées (Doublié *et al.*, 2011). Pour les approches fondées sur les modèles dynamiques, cela nécessite de connaître à chaque instant la configuration géométrique du robot et de prendre en compte son interaction avec le sol par la détection ou par la mesure du contact patte sol. L'utilisation d'un modèle dynamique pour la commande de la marche est donc une méthode souvent délicate à mettre en œuvre. Pour les approches fondées sur les modèles biologiques, cela nécessite de construire des architectures neuronales sensorimotrices produisant des rythmes stables face à cette redondance. La difficulté est alors de savoir quelle méthodologie suivre et quels modèles utiliser.

La question de l'équilibre durant la marche des robots humanoïdes est un problème fondamental dans leur commande car leur posture debout les rend naturellement instable. En effet, non seulement l'alternance des phases de simple appui et de double appui peut être irrégulière (causée par exemple par des phases d'arrêt), mais elle doit aussi pouvoir être modifiée subitement suite à une perturbation de type poussée latérale ou frontale afin d'éviter la chute (Goswami et Kallem, 2004). Enfin, les modifications morphologiques que pourrait subir un robot marcheur suite à des dommages (amputation, jeux mécaniques, élasticité organe actif devenant passif...) entraînent la prise en compte de situations de sous actionnement ou de sur actionnement suivant le type de dommage causé.

Toutefois, des solutions fondées sur la modélisation dynamique ont été trouvées pour contrôler la marche et l'équilibre des robots à pattes face à certaines perturbations comme des poussées latérales ou frontales, ou une irrégularité du sol. Ces méthodes basées essentiellement sur une approche qui modélise tout ou partie du comportement dynamique du robot, consiste à rechercher des lois de commande qui coordonnent les mouvements des pattes tout en maintenant l'équilibre global du robot. Ainsi, la majorité des contrôleurs de robots bipèdes actuels intègre un système de

contrôle de l'équilibre basé sur la théorie du *Zero Moment Point* (ZMP) (Vukobratovic et Borovac, 2004). L'avantage principal de cette méthode est qu'elle permet de suivre les séquences de contacts des pieds avec le sol en donnant une trajectoire désirée à la projection verticale du ZMP sur le sol. L'inconvénient majeur, est qu'elle n'exploite pas la dynamique des jambes et qu'il est difficile de gérer des perturbations inconnues comme une poussée extérieure.

L'une des difficultés pour le contrôle de la marche d'un robot est de conserver à chaque instant son équilibre dynamique compatible avec le type de contact choisi entre le pied de support et le sol. Pour éviter le risque de chute, les commandes développées sur la plupart des robots humanoïdes, et qui sont établies sur une suite de consignes temporelles, conservent souvent des marges de stabilité importantes. Des phases de déséquilibre sont alors prosrites alors que l'on sait qu'elles peuvent être prise en compte pour obtenir une marche plus fluide. Pour cela, les commandes utilisent des trajectoires de référence qui sont fonction de la configuration du bipède et non des mouvements de référence. L'évolution temporelle de la coordination du rythme locomoteur n'est pas alors fixée par la consigne haut niveau mais par le comportement dynamique du bipède. Une alternative intéressante à la méthode du ZMP, qui a l'avantage de requérir une connaissance partielle du robot (centre de masse et moment cinétique total) et qui nécessite moins de calcul, consiste à modéliser la dynamique du robot comme celle d'un pendule inverse, dont la masse est celle du robot mais considérée comme ponctuelle et confondue avec son centre de gravité. Le centre de rotation du pendule est attaché au centre de pression qui se déplace sous les semelles des pieds du robot. L'équivalence de la dynamique du robot à celle du pendule est intéressante car elle prend en compte la dynamique globale du robot tout en simplifiant les calculs à opérer en ligne, ce qui permet de stabiliser la marche en modifiant dynamiquement le placement des pieds. En couplant cette méthode avec celle du ZMP, une loi de commande prédictive permet de calculer en ligne la trajectoire du centre de gravité du robot tout en assurant son équilibre (Kajita *et al.*, 2003; Dimitrov *et al.*, 2009). Le positionnement du centre de gravité du robot étant obtenu par la résolution du modèle géométrique inverse des membres inférieurs ou de l'ensemble du robot.

La coordination des mouvements en 3 dimensions est aussi un élément essentiel pour la génération d'allures et le contrôle de l'équilibre dynamique des robots marcheur (David et Bruneau, 2011). Pour les bipèdes, cette coordination est liée à la succession des phases de double appui, de simple appui et des phases de vol dans le cas de la marche dynamique (Tlalolini *et al.*, 2009). La coordination peut être modélisée dans l'espace articulaire (spatialement ou temporellement) ou dans l'espace opérationnel. Les modèles géométriques directes et inverses du système locomoteur peuvent alors être couplés à des critères énergétiques permettant une propulsion dynamique stable (Bruneau et David, 2007). L'utilisation du pendule inverse pour modéliser la dynamique du robot permet par ailleurs de proposer des algorithmes de contrôle à base d'oscillateurs linéaires couplés judicieusement afin de coordonner le cycle de la marche dans l'espace articulaire sans aucun modèle géométrique. Cette approche bio-inspirée proposée par Morimoto *et al.* (Morimoto *et al.*, 2006) est résumée dans la section 4.3.3 de ce manuscrit.

Nous avons vu que la redondance cinématique est une caractéristique intrinsèque aux robots marcheurs. Celle-ci est un atout qui leur permet a priori une très grande mobilité sur tous les types de terrains comme le montre le robot Bigdog de la société Boston Dynamics. Dans la nature, la redondance cinématique existe aussi et elle permet en plus à certains animaux de sacrifier certains de leurs membres actifs pour fuir un prédateur, l'architecture neuronale s'adaptant alors à la nouvelle morphologie de l'animal. Dans le cas d'un robot qui se trouverait dans une même situation d'amputation, ou avec une liaison active devenue soudainement passive, on sera alors face à un système qui est devenu soudainement sous-actionné, rendant très délicat le problème de la commande. Cette adaptation à la morphologie changeante du robot est, à ma connaissance, un domaine encore peu étudié par les méthodes fondées sur la dynamique. Elle l'est cependant, par les méthodes bio-inspirées, particulièrement par l'utilisation d'algorithmes génétiques. Les travaux de l'équipe de J. C. Bongard (Kouchmeshky *et al.*, 2006) sur la synthèse de neurocontrôleurs robustes face aux variations de la morphologie du robot à pattes suite à des dommages en sont un exemple intéressant.

Les premières expériences de marche que nous avons menées au LISV sur le robot ROBIAN ont rapidement montré l'importance de contrôler l'équilibre et de coordonner les mouvements du système locomoteur. Elles nous ont permis aussi de mettre en évidence l'impact réel des effets dynamiques des pattes sur la stabilité de la marche particulièrement lors des phases transitoires du

début du cycle et lorsque celui-ci devient rapide. Les travaux sur la marche du robot par réseaux de neurones (apprentissage de l'équilibre et genèse de cycle de marche) que j'ai commencés alors à mener était une alternative intéressante à explorer et complémentaire aux autres travaux de l'équipe sur les modèles et lois de commandes dynamiques (travaux d'O. Bruneau et F. B. Oueddou).

Suivre une telle démarche amène à se poser la question de son positionnement par rapport aux autres travaux dans ce domaine. Si l'on veut s'inspirer de la nature pour commander des systèmes artificiels, on peut chercher à comprendre les mécanismes cognitifs haut niveau (comme l'approche développementale suivie par l'équipe neurocybernétique à ETIS) ou les mécanismes moteurs bas niveau, approche que j'ai choisie de suivre au LISV du fait de ma trajectoire initiale. Ce manuscrit d'Habilitation à Diriger les Recherches montre que mes travaux se sont au cours du temps, naturellement rapprochés des mécanismes cognitifs mis en jeu dans la locomotion, tandis que l'équipe ETIS s'est rapprochée de la problématique du contrôle moteur des robots humanoïdes. Cette convergence des travaux est naturelle car elle correspond par ailleurs aux découvertes sur les liens étroits qui existent entre les mécanismes cognitifs haut niveau chargés de l'apprentissage du geste et de l'interaction, et les mécanismes moteurs bas niveau chargés de mettre en mouvement les organes moteurs. Mon travail de recherche sur la commande bio-inspirée des robots marcheurs se situe donc à la frontière, entre les approches fondées sur les modèles dynamiques et celles fondées sur les aspects cognitifs du contrôle moteur.

3.5 Objectif scientifique

Mon objectif scientifique vise à mieux comprendre le fonctionnement des mécanismes d'apprentissage qui sont liés au contrôle moteur bas niveau chez l'humain, pour les modéliser et les intégrer dans les contrôleurs des robots humanoïdes, afin de rendre ces derniers plus robustes face aux perturbations externes dues à l'environnement, ou à leur interaction avec l'humain (forces externes, glissement, pentes ou irrégularités du sol), ou aux dommages internes, soudains ou progressifs, qu'ils peuvent subir et qui mettent en péril leur mission (usures articulaires, amputation de membres moteurs, pertes sensorielles...). Cet objectif est certes ambitieux, mais nous savons que les structures nerveuses chargées des commandes motrices intègrent des mécanismes dynamiques complexes similaires à différents niveaux et à différentes échelles. En effet, ces situations que peut rencontrer un robot à pattes (voire figure 3.8) amènent par ailleurs à se poser la question du contrôle de la posture, c'est-à-dire du couplage dynamique qui existe entre les membres inférieurs chargés de déplacer le robot et le corps supérieur chargé du contrôle de l'équilibre, particulièrement pour les robots bipèdes. Ce couplage dynamique existe aussi dans les structures nerveuses chargées à la fois de générer les rythmes locomoteurs et de contrôler la posture de l'animal (Cattaert et LeRay, 2001; Mc Crea et Ryback, 2008). Ces structures neuronales exhibent parfois des comportements dynamiques complexes, semblables à des oscillateurs chaotiques. On pense donc que, pour permettre aux robots marcheurs de produire des déplacements fluides, il est important de coordonner la rythmicité de leurs pattes par des contrôleurs chaotiques qui maintiennent un couplage dynamique entre les phases des pattes et la dynamique du robot qui est elle-même perturbée par l'interaction physique avec le sol (Pitti *et al.*, 2009), et/ou avec l'homme pour les robots humanoïdes. Pour avancer dans l'autonomie et la capacité d'adaptation des robots du futur, il est donc nécessaire de mieux comprendre et de mieux modéliser les mécanismes biologiques qui sont au cœur des systèmes nerveux moteurs des animaux (Ijspeert, 2008).

Si l'on considère un robot humanoïde en interaction physique avec l'humain, trois boucles principales d'interaction peuvent être identifiées :

- Celle qui lie le robot à son contrôleur. C'est une boucle dynamique sensori-motrice qui doit permettre au contrôleur de s'adapter à la dynamique perçue du robot.
- Celle qui concerne l'interaction du robot avec le sol par les mouvements de marche. C'est une interaction rythmique, même durant les phases d'arrêt pour lesquelles on sait que le centre de pression et le ZMP oscillent sans cesse à de faibles amplitudes (Faugloire *et al.*, 2006).
- Celle qui représente l'interaction physique (voire l'interaction sociale) entre l'humain et le robot, notamment au niveau des mouvements des membres supérieurs. C'est un bouclage dynamique qui peut être rythmique ou non.

Même si ces trois boucles sont de nature différente, il est admis que, chez l'humain, les mécanismes

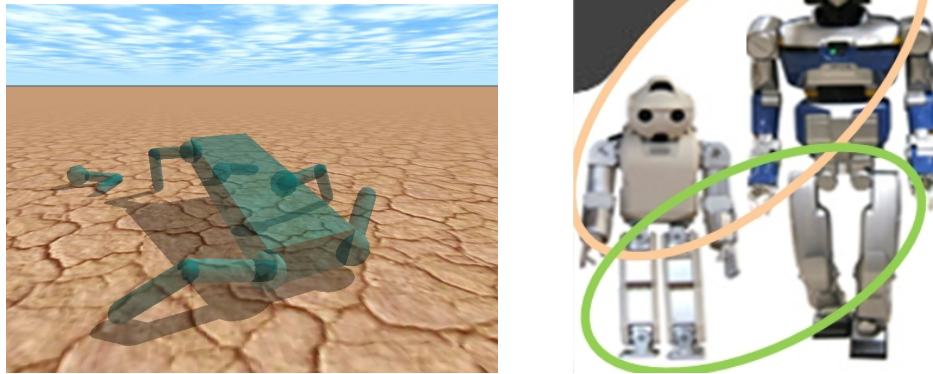


FIGURE 3.8 – À gauche, exemple de déficiences subies par un robot marcheur hexapode : amputation de pattes. Comment pouvons-nous synthétiser un contrôleur capable de s’adapter à une telle situation ? À droite : quels sont les modèles de contrôleurs bio-inspirés qui permettent un couplage dynamique stable entre la rythmicité des membres inférieurs et le mouvement du corps supérieur lorsque le robot est en interaction physique avec le sol ou avec l’être humain ?

d’apprentissage sous-jacents aux boucles sensori-motrices qui permettent d’une part, une marche fluide quelle que soient les perturbations, et d’autre part, une interaction physique « collaborative » quels que soient les mouvements, sont de même nature et basés sur les mêmes mécanismes sensorimoteurs. Une hypothèse plausible, assez proche de la commande optimale (Shadmehr, 2009), est que les centres nerveux moteurs responsables de la marche et du mouvement intègrent en leur cœur des oscillateurs chaotiques capables d’apprendre à être en phase, donc à minimiser l’énergie dissipée par le système commandé.

Nous verrons dans le chapitre consacré aux perspectives, que les travaux que je dirige maintenant vont vers la compréhension des mécanismes moteurs sous jacents à l’interaction physique et sociale, c’est-à-dire la troisième boucle. Par mon travail de recherche, résumé dans ce manuscrit, je me suis intéressé aux deux premières boucles en cherchant à répondre aux questions suivantes :

- Comment la marche adaptative des robots peut-elle être produite ?
- Comment peut-elle être contrôlée ?
- Quels types d’architecture permettent à la fois la production de rythmes locomoteurs et le contrôle de la posture ?
- Quels sont les mécanismes adaptatifs sensori-moteurs qui régissent ces architectures pour les marches normales et déficientes des robots ?
- Quelle est la robustesse de ces modèles faces aux perturbations et aux dysfonctionnements ?
- Quelles sont les méthodologies possibles pour modéliser et reproduire ces architectures ?
- Quelles sont les limites de ces approches ?
- Quelle est leur tolérance aux contraintes imposées par un système informatique temps réel embarqué d’un robot ?

Pour mener à bien ces objectifs, mes travaux se focalisent essentiellement sur deux axes principaux de la commande bio-inspirée des robots.

- Le premier axe concerne l’apprentissage du contrôle de la posture. Cette approche, qui est fondée sur des mécanismes d’apprentissage en ligne bas niveau a été initiée comme on l’a vu au chapitre 3 durant ma thèse pour la commande cartésienne de robots mobiles, puis appliquée ensuite à l’apprentissage de réflexes sur ces mêmes robots. J’ai donc appliqué cette approche par apprentissage aux robots humanoïdes pour l’apprentissage de réflexes d’équilibre (Scesa *et al.*, 2005; Hénaff *et al.*, 2011), et pour l’apprentissage de modèles de muscles (Serhan *et al.*, 2006, 2010).
- Le second axe concerne la genèse de rythmes locomoteurs adaptatifs pour les robots à pattes y compris les robots bipèdes. Cette approche est fondée sur l’étude des modèles neuronaux générateurs de rythmes (ou CPG pour central pattern generator, voir définition section 4.1.5)

qui sont à la base de la locomotion animale, et sur leurs mécanismes de régulation intrinsèque (Hoinville et Henaff, 2004a,b; Hoinville, 2007; Hoinville *et al.*, 2007; Nassour *et al.*, 2009, 2010a,b; Amrollah et Hénaff, 2010; Hoinville *et al.*, 2011).

Dans les travaux que j'ai dirigés, j'ai toujours eu la volonté d'étudier la robustesse de nos modèles face à des perturbations extérieures (frottements, forces extérieures, pentes du sol..) ou à des dommages importants. Nous verrons à travers la lecture de ce documents que l'objectif de compenser les amputations ou les sous actionnements n'est pas encore atteint. Cependant les résultats obtenus sur l'amputation des robots myriapodes sont intéressants et je pense que les modèles et méthodologies sur nous proposons permettront à moyen terme d'offrir des solutions nouvelles à ce type de problème.

Le chapitre suivant introduit les bases de la commande neuronale sans toutefois avoir la prétention de faire une revue exhaustive des différentes approches. Je présenterai ensuite les travaux que j'ai menés via l'encadrement de doctorants en présentant dans un premier chapitre les résultats obtenus sur l'apprentissage du contrôle de la posture, et dans les chapitres suivants les travaux sur la genèse de rythmes locomoteurs. La présentation de ces travaux en est une synthèse et ne correspond donc pas forcément à un ordre chronologique. Les références bibliographiques citées au cours de ces deux chapitres ne constituent pas une liste exhaustive. Pour une compréhension approfondie et une étude bibliographique complète, le lecteur se référera à mes articles principaux qui sont donnés en annexe de ce document..

Chapitre 4

Approche bio-inspirée du contrôle moteur : aspects biologiques et computationnels

De mon point de vue, il est important de comprendre les différences essentielles qui existent entre, d'une part les principaux modèles de réseaux de neurones que l'on peut utiliser pour la commande de robots, et d'autre part les algorithmes qui leur sont associés pour adapter, voire optimiser les paramètres de ces contrôleurs neuronaux. C'est pourquoi ce chapitre résume les mécanismes biologiques fondamentaux qui sont à la base des centres nerveux générateurs de rythmes en tentant de mettre en valeur ce qu'il est important de connaître pour un non-biologiste. Ces mécanismes engendrent un couplage dynamique entre le système nerveux moteur et le système musculo-squelettique par des boucles sensori-motrices qui permettent la fluidité et la précision du mouvement. Cependant, je n'aborderai pas, dans ce chapitre, les aspects biomécaniques et musculaires du mouvement même si ces aspects sont fondamentaux pour comprendre la marche animale et humaine. La biomécanique est un domaine de recherche important qui permet, entre autre, de produire des modèles de comportement dynamique pour la simulation de l'être humain (Multon, 2006). Ce champs de recherche permet aussi de proposer des modèles cinématiques de mannequins virtuels aboutissant à la simulation dynamique des robots humanoïdes et à leur conception comme cela a été fait au laboratoire LISV par les travaux de O. Bruneau sur le modèle de Hanavan (Gravez *et al.*, 2005). La compréhension des phénomènes musculaires est aussi un enjeu important pour la robotique humanoïde notamment pour la conception d'actionneurs originaux compliants (voir section 5.2). Les travaux des chercheurs sur la modélisation des muscles a produit différents modèles macroscopiques de muscle, dont les plus répandus sont le modèle de Hill (Hill, 1938) et celui de Hatz (Hatze, 1976). Nous verrons au chapitre suivant que nous nous sommes intéressé à cet aspect du contrôle moteur en proposant un modèle de muscle à base de réseaux de neurones de type NNARX (pour plus de détail voir la section 5.2 et (Serhan *et al.*, 2006, 2010) et la thèse de H. Serhan (Serhan, 2009)). Les autres travaux sur le système musculo-squelettique vont plutôt vers la modélisation et la compréhension des phénomènes d'activation d'un groupe musculaire pour un mouvement complexe donné (voir les travaux du LISV sur la préhension et la manipulation d'objet (Chalfoun, 2005)).

4.1 Principes généraux de la neurophysiologie du geste moteur

4.1.1 Structure et complexité du Système Nerveux Central

Chez l'être humain, les gestes moteurs sont « pilotés » par le système nerveux qui est une structure biologique très complexe constituée principalement du Système Nerveux Central (SNC) et du Système Nerveux Périphérique (SNP). Le SNC est lui-même constitué de l'encéphale (cerveau

et cervelet) et de la moelle épinière (ME). Cette architecture est telle que le cerveau peut être assimilé au niveau supérieur du contrôle et la moelle épinière au niveau inférieur. En effet, certains mouvements ne sont générés que dans la moelle épinière mais ils peuvent être modulés par le niveau supérieur. C'est pourquoi il a été observé chez certains animaux décérébrés des mouvements rythmiques synchrones permettant leur locomotion.

Le SNP est constitué par l'ensemble des nerfs rattachés au système nerveux central : nerfs afférents provenant des récepteurs périphériques chargés de percevoir les sensations (tactiles, thermiques, douleur..), nerfs efférents (ou nerfs moteurs) chargés de commander les muscles, ainsi que par les récepteurs périphériques (voir Figure 4.1).

Le SNP peut être décomposé en deux parties :

- le Système Nerveux Végétatif ou Systèmes Nerveux Autonome qui régule les différentes fonctions automatiques de l'organisme (digestion, respiration, cœur...).
- le Système Nerveux Somatique (SNS) qui correspond au Système Nerveux Moteur et sensitif (figure 4.2). C'est le SNS qui est responsable du maintien des postures, des mouvements volontaires et involontaires, ainsi que des sensations. C'est donc ce système qui intéresse les roboticiens pour trouver des solutions nouvelles dans la commande des robots.

La compréhension de la neurophysiologie du geste moteur dans le but de transférer les connaissances acquises vers la robotique est donc un défi important car cela nécessite d'établir des modèles du système musculo-squelettique et du système nerveux moteur qui sont liés par les terminaisons nerveuses neuronales. Le neurone et les mécanismes d'apprentissage ou de régulation qui lui sont associés sont au centre des boucles sensori-motrices. Il est donc l'élément clé de la communication dans le système nerveux moteur. Dans ce qui suit, je résume les grands principes régissant les structures nerveuses motrices que nous avons retenue pour l'élaboration de nos contrôleurs neuronaux chargés de la genèse de rythme locomoteurs.

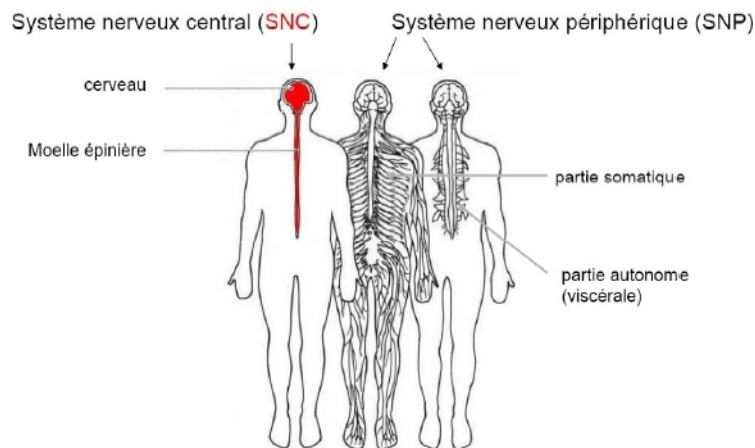


FIGURE 4.1 – Grandes divisions du Système Nerveux (Extrait : Univ du Québec-etsmtl (cours)). Le SNP est anatomiquement mais pas fonctionnellement distinct du SNC. Sa partie autonome assure l'essentiel des fonctions végétatives, sa partie somatique assure la motricité.

4.1.2 Structure du Système Nerveux Moteur

Le Système Nerveux Moteur (SNM) comprend plusieurs milliards de neurones qui communiquent entre eux par l'intermédiaire des synapses qui permettent le passage des influx nerveux nécessaires aux activations musculaires et aux sensations. Pour le roboticien, cette structure très complexe peut être vue comme un système en boucle fermée (figure 4.3). En effet, les cellules sensorielles nombreuses et variées qui jouent le rôle de capteurs extéroceptifs et proprioceptifs sont présentes sur la peau, dans les articulations et dans les muscles. De plus, les muscles, qui jouent le rôle d'actionneurs, sont constitués de nombreuses fibres musculaires dont chacune peut être vue comme un actionneur particulier couplé aux autres. Enfin, les lois de contrôle et de correction sont établies pour certaines au niveau local (SNM), et pour d'autres au niveau global (SNC).

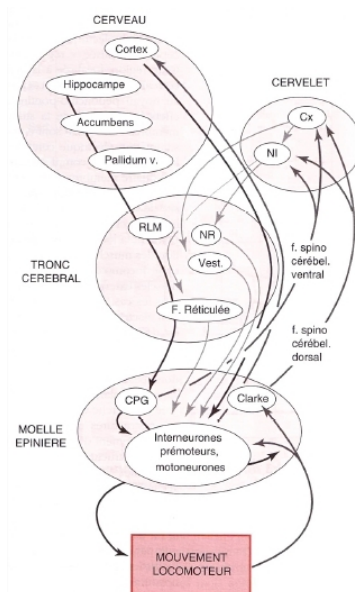


FIGURE 4.2 – Synthèse de l'architecture du Système Nerveux Moteur

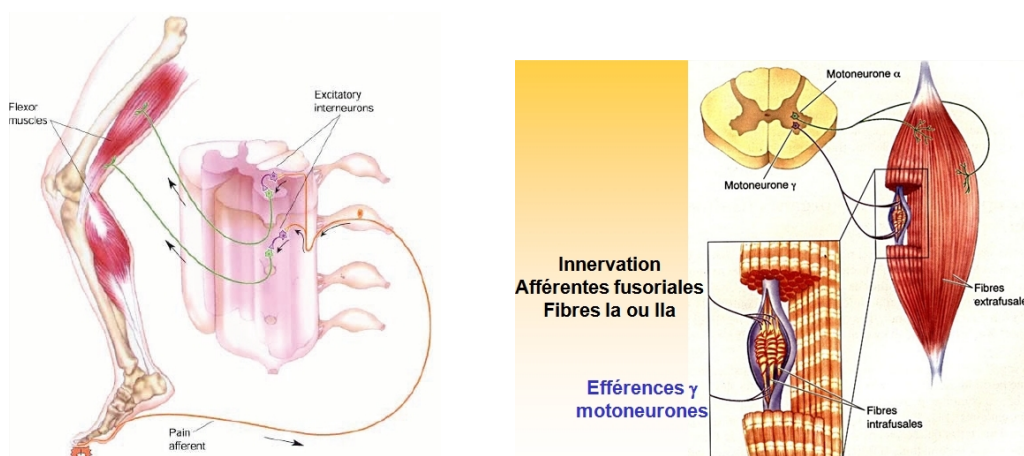


FIGURE 4.3 – Système nerveux moteur (à gauche) et fuseau neuromusculaire (à droite) (Extrait : cours du collège de France, L. Damn).

La figure 4.3 (à droite) donne le schéma de principe de la boucle neuromusculaire. Les motoneurones α envoient depuis la moelle épinière les signaux efférents sur le muscle via leurs axones qui se ramifient sur les fibres pour établir les contacts synaptiques. L'activation des motoneurones α est responsable de la production de force dans le muscle, provoquant ainsi sa contraction. Une fibre musculaire n'est innervée que par un seul motoneurone, et l'ensemble formé par un motoneurone α et la fibre musculaire qu'il innerve représente une unité motrice. L'unité motrice est le plus petit élément contractile que le système nerveux peut commander.

Les fibres afférentes *Ia* sont issues des *fuseaux neuromusculaires* qui sont intercalés entre les fibres musculaires. Les fibres *Ia* véhiculent les informations proprioceptives sur la longueur musculaire. Contrairement aux motoneurones α , l'activité des motoneurones γ ne modifie pas la longueur musculaire mais informe le système nerveux de la vitesse du changement de longueur du muscle. Par la boucle fermée qu'ils forment avec les fibres *Ia*, les neurones γ sont responsables du réflexe myotatique (contraction réflexe d'un muscle déclenchée par son propre étirement). Celle-ci présente une activité de fond qui est le niveau zéro caractérisant l'état d'excitabilité du motoneurone α , c'est-à-dire son point de fonctionnement. Cette boucle dite gamma est considérée comme un dispositif de servo-contrôle intervenant dans la contraction musculaire.

Concrètement, la transmission des influx nerveux provenant de la moelle épinière via les motoneurones est effectuée au niveau des jonctions neuromusculaires. La figure 4.4 illustre bien la complexité de cette jonction où les motoneurones sont connectés aux fibres musculaires via l'interface membranaire post-synaptique.

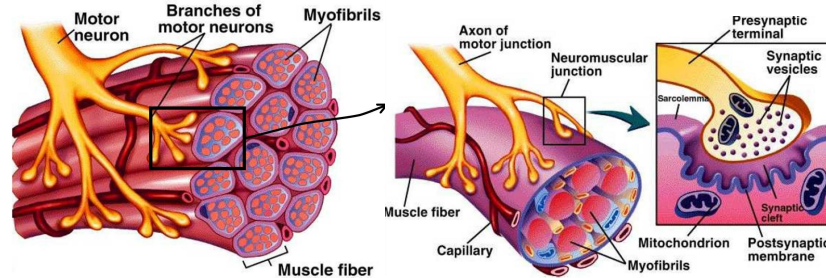


FIGURE 4.4 – Fonctionnement de la jonction neuro-musculaire. À gauche l'unité motrice, à droite un zoom de la jonction neuromusculaire sur la fibre musculaire.

Dans la présentation succincte qui précède, nous avons montré que le système musculo-squelettique et le système nerveux moteur sont liés par les terminaisons nerveuses neuronales. Le neurone qui est au centre des boucles sensori-motrices est donc l'élément clé de la communication dans le système nerveux moteur. Les contrôleurs neuronaux biologiques qui nous concernent en robotique sont principalement constitués des types suivants de neurones qui amènent à l'élaboration des différents modèles que nous utilisons dans nos recherches :

- les *motoneurones* qui sont des neurones moteurs qui engendrent le mouvement par la contraction des muscles,
- les *neurones sensoriels* qui sont les capteurs extéroceptifs ou proprioceptifs,
- les *interneurones pré-moteurs* plus nombreux que les deux premiers et qui, grâce à leur propriété d'inhibition, sont au cœur de la genèse des rythmes locomoteurs.

4.1.3 Propriété de Plasticité

La plasticité des réseaux de neurones est une de leurs propriétés les plus fondamentales. Beaucoup de recherche vont actuellement dans ce sens et de nombreuses découvertes sont encore à faire dans ce domaine. En biologie, la plasticité recouvre un grand nombre de phénomènes cellulaires qui se manifestent à des niveaux anatomiques et sur des échelles de temps très variés. La plasticité neuronale est intéressante pour le roboticien car elle est le fondement des mécanismes d'apprentissage qui ouvrent la voie à une meilleure compréhension des phénomènes d'adaptation que nous avons présenté en introduction. En effet, l'apprentissage repose sur la capacité des neurones à modifier de façon durable leur transmission synaptique. Le principe de base de l'apprentissage repose sur le postulat que Hebb a proposé en 1949 (Hebb, 1949). Cette loi de Hebb dit que si deux neurones sont actifs en même temps, alors les synapses liant ces neurones seront renforcées. C'est donc les connections entre les neurones qui sont renforcées ou affaiblies selon l'activité de ces neurones. Grâce à cette plasticité, le réseau de neurones est capable de réorganiser sa topologie en fonction des retours sensoriels ou de certains mécanismes de modulation. Tous les modèles d'apprentissage proposés, supervisés ou non, sont inspirés par ce postulat de Hebb qui lie la valeur des synapses à l'activité des neurones.

Les phénomènes de plasticités font partie des bases cellulaires de l'apprentissage, de la mémoire et des comportements adaptatifs. Il existe 2 grands mécanismes de plasticité :

- La plasticité synaptique à long terme :
On définit la plasticité à long terme par les mécanismes de potentialisation et de dépression à long terme (PLT et DLT). Une stimulation électrique répétitive en amont d'une synapse excitatrice induit un accroissement (PLT) ou une réduction (DLT) durable de l'efficacité synaptique de la transmission synaptique (poids synaptique). La durée du phénomène varie de quelques dizaines de minutes à plusieurs jours.
- La plasticité synaptique à court terme :

Réduction ou accroissement momentané de l'amplitude du potentiel pos-synaptique lors de décharges pré-synaptiques temporellement contiguës (dépression et facilitation à court terme). Cette variation est fonction de l'historique récent de l'activité pré-synaptique uniquement. Les constantes de temps sont de l'ordre de quelques millisecondes à quelques minutes.

La DLT joue un rôle important dans l'apprentissage du mouvement en corrigeant les procédures motrices lors de l'apprentissage d'un savoir-faire. Elle intervient dans le cervelet, où les réseaux neuronaux impliqués dans des mouvements erronés sont inhibés par la mise sous silence de leurs connections synaptiques, et elle limite le renforcement dû à la PLT. En effet, sans la DLT, les synapses atteindraient toutes en peu de temps un degré maximal d'efficacité (saturation des neurones) et il serait alors impossible de coder de nouvelles informations.

4.1.4 Mouvements réflexes et mouvements rythmiques.

La complexité des structures en réseaux que les neurones constituent dans le système nerveux moteur est liée au type de mouvement mis en œuvre par ces structures. Du point de vue biologique, l'apprentissage moteur peut être défini comme «*un processus d'adaptation cognitivo-moteur, relié à la pratique et à l'expérience, favorisé par des conditions d'apprentissage qui mènent à des changements permanents de la performance et de l'habileté motrice*» (Chevalier et Taktek, 2004). L'apprentissage moteur progresse lentement par rapport à la bande passante du système qu'il anime, c'est-à-dire que plusieurs essais sont nécessaires. Mais une fois appris, l'apprentissage reste acquis pour une période plus ou moins longue par rapport à l'échelle temporelle du système.

Trois types de mouvements peuvent être définis : les mouvements volontaires, les mouvements réflexes et les mouvements rythmiques.

Les mouvements volontaires sont des mouvements non stéréotypés initiés, inhibés et contrôlés par le cortex. Ils sont souvent intentionnels et s'améliorent avec la pratique. Les gestes du toucher et de la saisie sont deux exemples de mouvements volontaires.

Les mouvements réflexes sont initiés dans la moelle épinière ou le tronc cérébral. Ils ont des patrons d'activité nerveuse stéréotypés et déclenchés par stimuli externes. Ils peuvent être appris ou désappris (ils sont plastiques), sont modulables et difficiles à inhiber (le test médical du réflexe myotatique d'étirement en est un bon exemple). Il y a deux types de mouvements réflexes, les réflexes spinaux initiés dans la moelle épinière (s'étirer...) et les réflexes vestibulaires initiés dans le tronc cérébral qui sont par exemple liés à l'équilibre et au maintien du regard lorsqu'on tourne la tête.

Les mouvements rythmiques sont initiés et inhibés par le cortex. Ils ont des patrons d'activité nerveuse stéréotypés et périodiques. Ces rythmes sont générés et contrôlés dans la moelle épinière. Les patrons rythmiques subissent des modulations induites par les boucles sensorielles locales comme l'étirement des muscles ou la perception de l'effort de pression sur la voute plantaire. Ainsi, la marche et la mastication sont deux exemples très différents de mouvements rythmiques. Mais il convient de les distinguer des mouvements de respiration ou du rythme cardiaque qui ne sont pas des mouvements rythmiques au sens où ils sont engendrés par le système nerveux autonome.

Dans notre travail et compte tenu de nos objectifs scientifiques, nous nous inspirons des mouvements réflexes et rythmiques pour construire des architectures de contrôle à base de réseaux de neurones artificiels. Comme dans la nature, ces architectures seront à long terme composées de plusieurs niveaux, certains dédiés à l'apprentissage de réflexes pour le contrôle de la posture, d'autre à la genèse de mouvements rythmiques pour la locomotion. Les mouvements volontaires ne rentrent donc pas dans le cadre de mes activités de recherche même s'ils ont été étudiés par ailleurs au sein du laboratoire LISV (Bensmail, 2005) et le sont à ETIS (Lagarde *et al.*, 2010; Lagarde, 2010).

4.1.5 Architecture sensori-motrices pour la locomotion

La locomotion est une activité motrice automatique complexe. Les réseaux neuronaux générateurs de rythmes sont présents chez les mollusques, les vertébrés inférieurs et les mammifères, pour la locomotion mais aussi pour d'autres phénomènes rythmiques comme par exemple la mastication.

Il est ainsi admis que ces patrons générateurs centraux dénommés CPG (Central Pattern Generator), et situés dans la moelle épinière, sont les centres de contrôle de la locomotion des insectes et des mammifères et qu'ils sont composés de sous-ensembles de neurones. Ces sous-ensembles (ou unités) sont capables de générer de façon autonome un signal périodique contrôlant les muscles fléchisseurs et extenseurs autour d'une articulation et de se synchroniser avec d'autres sous-ensembles par des couplages synaptiques afin d'assurer la coordination de plusieurs membres (figure 4.5).

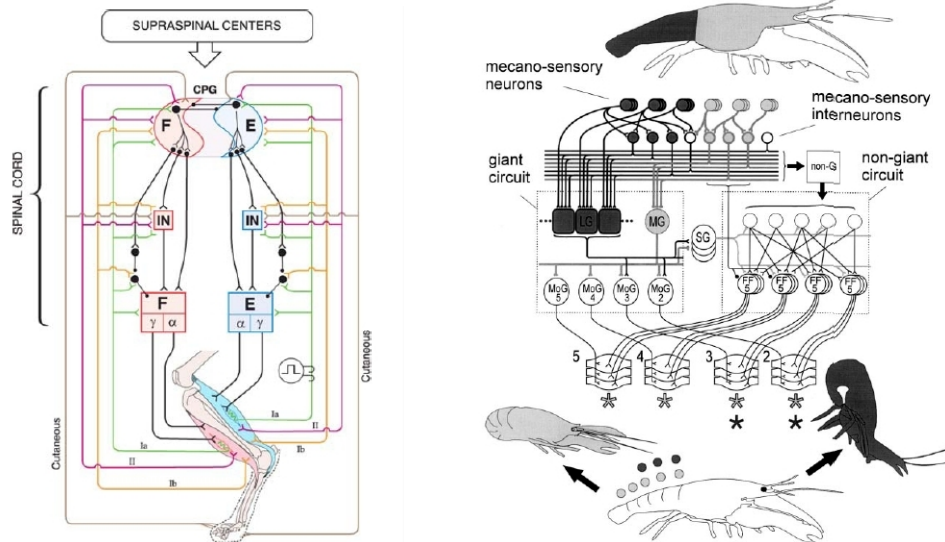


FIGURE 4.5 – Schéma général de la boucle sensori-locomotrice du membre arrière du chat (extrait de (Frigon et Rossignol, 2006)) et de l'écrevisse dans la phase de saut (extrait de (Cattaert et LeRay, 2001)).

Dans le système nerveux des invertébrés (principalement des insectes), l'organisation des réseaux de neurones responsables de la coordination des organes articulés a été étudiée (pour une revue, voir (Büschges, 2005)). Beaucoup de neurones ont été identifiés, leurs propriétés et leur organisation synaptique analysées. Plusieurs simulations réalistes d'insectes qui se déplacent ont été réalisées pour simuler ces modèles (Cruse *et al.*, 1995)(Cruse *et al.*, 1998). Par exemple, l'organisation du système nerveux responsable de la marche des écrevisses est largement modulaire (Cattaert et LeRay, 2001). Chaque patte est commandée par son propre réseau neuronal central et il y a un grand nombre de neurones pour une liaison articulaire. Les biologistes ont recensé différents types de neurones qui sont fondamentaux pour produire un motif rythmique cohérent : les neurones rythmiques, les interneurones prémoteurs, et les motoneurones. Les motoneurones appliquent les signaux de contrôle musculaires aux fibres musculaires. Les neurones rythmiques produisent des oscillations qui sont modulées par les interneurones prémoteurs. Les interneurones sont impliqués dans le contrôle local de l'articulation. Certains interneurones sont des neurones sensoriels qui « mesurent » la position angulaire, la vitesse, ou les forces internes de l'articulation. La plupart des interneurones sont des neurones non-impulsionnels. Les biologistes ont démontré que les connexions synaptiques dans les réseaux neuronaux contrôlant l'articulation d'une patte sont de trois sortes (Marchand *et al.*, 1997) : des synapses chimiques classiques activées par la présence d'une activité pré-synaptique (certaines sont excitatrices, d'autres sont inhibitrices en fonction de neurotransmetteurs et de récepteurs postsynaptiques) ; des synapses chimiques toniques qui ne nécessitent pas de libérer des neurotransmetteurs, et les connexions électriques entre les motoneurones d'un même groupe de terminaisons sensorielles d'un même organe, entre les interneurones prémoteurs et les motoneurones.

4.2 Modèles computationnels.

Les connaissances acquises sur les systèmes nerveux moteur des animaux étudiés en biologie ont permis de dégager quelques grands principes dont le plus fondamental : « *tous les mouvements*

rythmiques sont générés par des CPGs dont l'action peut être modifiée et ajustée par les influences sensorielles » (Grillner, 1985) (Bässler et Büschges, 1998). De plus, nous savons que ce système nerveux moteur des animaux à pattes est modulaire et symétrique et qu'il est commandé par un niveau supérieur. Il peut engendrer une activité tonique pour le contrôle de la posture et une activité rythmique responsable de la marche. Il possède des propriétés de plasticité et la perception du contact avec le sol est essentielle pour la stabilité du rythme. Enfin, certains animaux qui possèdent la propriété « d'autonomie », c'est-à-dire la capacité à perdre volontairement un membre afin d'échapper à un prédateur, voient leur système nerveux moteur se reconfigurer pour s'adapter à la nouvelle morphologie de l'animal. Cette dernière propriété est très intéressante si l'on considère l'autonomie d'un robot qui doit continuer à réaliser sa tâche même après la perte d'un organe moteur.

Depuis deux décennies, différents modèles ont été mis en œuvre pour concevoir des architectures de neurones artificiels capables d'engendrer des cycles de marche, pour des robots à pattes, bipèdes, réels ou simulés de différentes tailles ou pour reproduire la marche de certains insectes ou animaux vertébrés Ekeberg (1993) (Cruse *et al.*, 1995, 1998). Dans cette section, je résume les avantages et inconvénients des modèles principaux qui sont intéressants pour la robotique (citons les travaux les plus représentatifs (Taga *et al.*, 1991) (Wadden T. et A, 1993) (Ijspeert *et al.*, 1998) (Worgotter *et al.*, 2006)), en insistant sur la complexité de mise en œuvre et leur robustesse. Nous verrons comment certaines de ces solutions permettent d'incorporer dans le contrôleur, ou dans les éléments qui le composent, des mécanismes adaptatifs, voire d'apprentissage inspirés de la loi de Hebb.

La difficulté principale liée à ces modèles provient du calcul de la valeur optimale de leurs paramètres ou des poids synaptiques qui leur sont attachés. En effet, plus le modèle est biologiquement plausible, plus il est complexe mathématiquement et plus les algorithmes d'apprentissage classiques, comme la rétropropagation, sont difficiles à mettre en œuvre sur ces modèles. Quand un algorithme d'apprentissage ne peut pas être mis en œuvre, une approche très répandue consiste donc, plutôt que de chercher les valeurs de ces paramètres par une approche empirique, à les optimiser par un algorithme génétique. Nous verrons que cette approche peut être très utile pour explorer des solutions ou des classes de solutions, mais qu'elle ne permet pas une compréhension complète du fonctionnement du neurocontrôleur. C'est pourquoi les travaux de recherche dans ce domaine ont évolué au cours des années vers différents niveaux de modélisation qui permettent soit une synthèse manuelle des paramètres ; soit une synthèse automatique par algorithme génétique. Selon moi, ces modèles peuvent donc être classés en trois niveaux principaux de modélisation :

- le niveau le plus bas où l'élément de base est le neurone biologique souvent modélisé comme un système du premier ordre non linéaire
- un niveau intermédiaire où l'élément de base est un oscillateur non linéaire, modélisant le fonctionnement d'un sous-ensemble de neurones,
- le niveau le plus haut, où l'élément de base est un oscillateur sinusoïdal

Je présente ici les modèles de CPG les plus répandus dans la littérature et les plus intéressants selon moi pour la genèse de mouvements rythmiques en robotique.

4.2.1 Modèles de neurone

Modèle dynamique du premier ordre.

Le modèle de neurone le plus répandu, est une version dynamique du premier ordre (équation 4.1) du neurone de Mc Culloch et Pitts utilisé dans les perceptrons multicouches. Lorsque plusieurs neurones de ce type sont interconnectés entre eux, ce modèle de neurone, appelé souvent « leaky-integrator », forme les CTRNN (*continuous time recurrent neural networks*). Ce modèle a été très étudié (Grossberg, 1969; Hopfield, 1984; Beer, 1995b,a, 2006a), et il permet de reproduire assez fidèlement ce qui est observé en biologie pour la locomotion :

$$y_i = -\tau_i \frac{dy_i}{dt} + \sum_{j=1}^{N_i} w_{ij} o_j \quad \text{et} \quad o_i = \text{sigmo}(y_i + \theta_i) \quad (4.1)$$

Ce neurone, que nous appelons pour simplifier dans la suite du document, modèle ou neurone « de Beer », calcule la moyenne de la fréquence de décharge d'un neurone. Dans ce modèle, o_i est la sortie du neurone i , τ_i est la constante de temps, w_{ij} est le poids synaptique du neurone j

vers le neurone i . Ce modèle est le plus répandu car il permet d'appliquer aisément la plupart des algorithmes d'apprentissage, y compris la rétropropagation du gradient (rétropropagation temporelle ou *back propagation through time*, (Pineda et Fernando, 1987)) comme nous l'avons fait (voir section 5.1.1).

Les CTRNN ont été très utilisés dans des domaines variés pour leur capacité à être des approximateurs universels (Doya et Yoshizawa, 1989; Beer, 1995a; Chow et Xiao-Dong, 2000). Leur propriétés dynamiques ont été explorées et continuent encore à l'être (Beer, 2006b; Siri *et al.*, 2008; Nakamura et Nakagawa, 2009).

Cependant, le modèle de Beer étant un système du premier ordre, il n'est intrinsèquement pas rythmique. Pour produire un système dynamique oscillant avec un tel modèle, il convient donc d'interconnecter judicieusement au moins deux de ces neurones.

Neurone rythmique de Matsuoka

Le modèle de Matsuoka (Matsuoka, 1987) est un modèle intrinsèquement rythmique. Il est composé de deux cellules régies chacune par deux équations différentielles couplées par une fonction non linéaire (fonction seuil) :

$$\begin{cases} \tau_{ai}\dot{u}_i = -u_i - \beta f(v_i) + \sum_j w_{ij}f(u_j) + u_0 + Q_i + S_i \\ \tau_{bi}\dot{v}_i = -v_i - f(u_i) \\ f(u_i) = \max(0, u_i) \end{cases} \quad (4.2)$$

Dans ce modèle, u_i est la sortie de l'oscillateur, τ_{ai} et τ_{bi} ses constantes de temps, β_i le biais. Ce modèle est capable de produire différents comportements dont des oscillations stables. C'est cette propriété qui a été exploitée par Taga (Taga *et al.*, 1991; Woosung *et al.*, 2008) pour la marche bipède (voir sous-section 4.3.3). Dans ce modèle, Q_i représente les entrées de couplages internes au réseau et S_i les entrées sensorielles articulaires.

Neurone rythmique de Rowat Selverston

Le modèle de neurone de Rowat et Selverston (Rowat et Selverston, 1993) est un modèle intéressant car c'est un modèle généralisé de l'oscillateur de Van Der Pol. Il peut exhiber les 4 phases principales d'un neurone rythmique biologique : oscillateur endogène, potentiel en plateau, amorti, et quiescent (Marder et Bucher, 2001) comme le montre la figure 4.6.

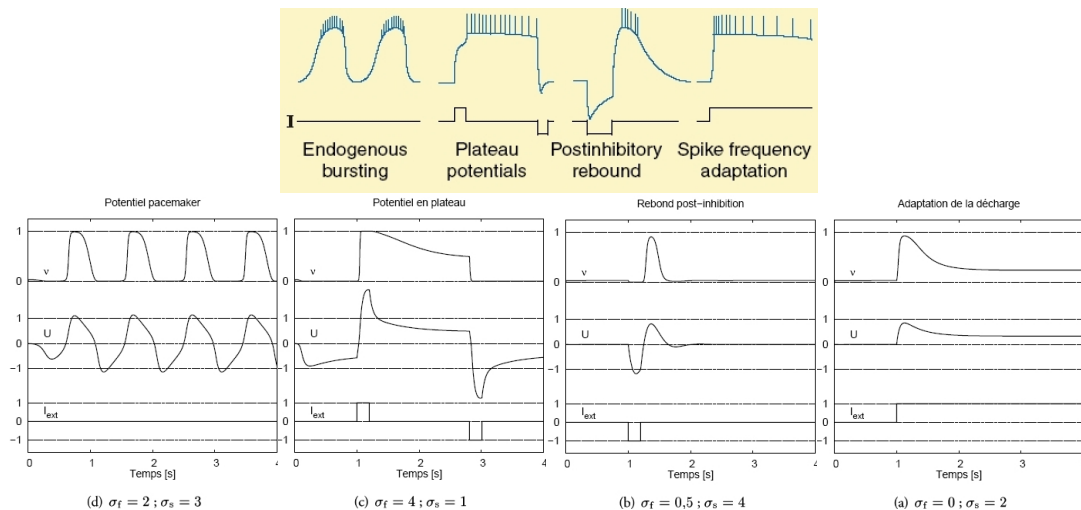


FIGURE 4.6 – Propriétés intrinsèques des neurones rythmiques biologiques en haut (figure extraite de (Marder et Bucher, 2001)), en en bas simulation correspondante (extrait de (Hoinville, 2007)).

Ces phases étant contrôlables par 2 paramètres σ_s et σ_f indépendants. La fréquence et l'amplitude des oscillations peuvent aussi être ajustées (pour plus de précisions sur ce modèle voir entre autre notre article (Amrollah et Hénaff, 2010) en annexe 7.1.1).

Ce modèle est basé sur 2 équations différentielles du premier ordre, qui correspondent aux variations lentes et rapides des courants induits dans les canaux ioniques des neurones biologiques (équations 4.3). Le couplage entre ces deux fonctions est réalisé par une fonction non-linéaire paramétrable incluant une tangente hyperbolique.

Ainsi le potentiel membranaire U est régi par les équations :

$$\begin{cases} \tau_m \frac{dU}{dt} = -I_{rapide} - I_{lent} - I_{syn} + I_{ext} \\ I_{rapide} = U - A_f \tanh\left(\frac{\sigma_f}{A_f} U\right) \\ \tau_s \frac{dI_{lent}}{dt} = \sigma_s U - I_{lent} \text{ avec } \tau_s > \tau_m \end{cases} \quad (4.3)$$

Le courant I_{syn} , dû aux afférences synaptiques, est la somme des courants inhibiteurs I_- et excitateurs I_+ :

$$I_{\pm} = \sum_i w_i \nu_i (U - E_{\pm}) \text{ et } \nu_i = \frac{1}{1 + \exp(-\alpha(U_i - \theta_i))} \quad (4.4)$$

Les paramètres E_{\pm} sont les tensions d'équilibre (potentiel de Nernst) qui caractérisent les réponses synaptiques.

Ce modèle de neurone permet donc de reproduire les phases principales des neurones rythmiques endogènes décrits par les biologistes et peut jouer le rôle d'attracteur afin de maintenir des modes rythmiques stables. De plus, le fait que les paramètres du neurone permettent de contrôler et de commuter entre ses phases, est intéressant pour réaliser des attracteurs dynamiques dont le point d'équilibre peut changer, et donc de réaliser des couplages dynamiques efficaces. Ces paramètres peuvent de plus être modulés par des mécanismes de neuromodulation par exemple liés à la perception d'une déficience interne du robot ou d'une perturbation externe qu'il subit.

Malgré ses propriétés intéressantes, ce modèle de neurone a été peu étudié (citons (Bem et J., 2004)). Les premiers travaux sur son utilisation en robotique sont ceux de T. Hoinville dans le cadre de sa thèse que j'ai co-encadrée sur la locomotion myriapode (Hoinville, 2007). Nous avons depuis utilisé ce modèle pour la commande de la marche bipède (Nassour *et al.*, 2009, 2010a,b; Amrollah et Hénaff, 2010).

4.2.2 Modèles d'architectures

L'architecture globale du réseau de neurones biologique chargé de la commande motrice d'une patte peut être décrit en trois couches principales interconnectées (Mc Crea et Ryback, 2008) : la couche rythmique, chargée de la création des signaux rythmiques, la couche formatrice chargée de la modulation de ces signaux et la couche régulatrice chargée du contrôle de la stabilité de la boucle sensorimotrice via les retours sensoriels. Malgré cette complexité, il est possible de réduire l'architecture nerveuse d'un réseau neuronal biologique contrôlant une liaison articulaire à un modèle qui respecte ces propriétés fondamentales. Mc Crea et Ryback suggèrent une organisation intéressante des neurones dans lequel un centre générateur de rythmes (CPG) est construit en 3 niveaux (voir figure 4.7) : un générateur de rythmes (RG) basé sur des neurones oscillatoires, un étage de formation (modulation) des patrons rythmiques (PF) basé sur des interneurons responsables de l'activation des motoneurons et la couche motrice contenant les motoneurons et leur régulation par les retours sensoriels.

Certains mécanismes biologiques situés au cœur des unités neuronales présentes dans les CPG peuvent être assimilés à des paramètres de contrôle ou à des mécanismes de régulation qui permettent d'assurer la stabilité du CPG ou de transiter vers un autre patron de coordination plus stable. Du fait de la dynamique des neurones, de leur non-linéarité, et du fort couplage synaptique, les CPG biologiques sont des systèmes dynamiques complexes dont plusieurs niveaux de modélisation ont été proposés en locomotion robotique comme on le verra à la section suivante.

4.2.3 Modèles de mécanisme adaptatif : plasticité

Le travail de synthèse sur les différents modèles mathématiques de plasticité que nous avons effectué avec Thierry Hoinville (voir le premier chapitre de sa thèse (Hoinville, 2007)) est un

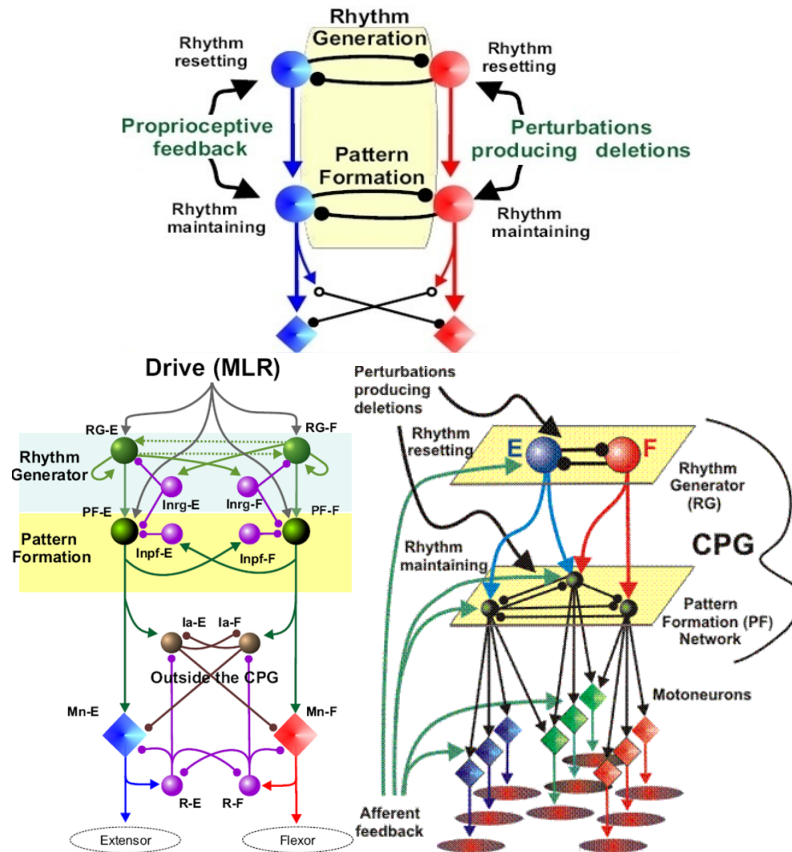


FIGURE 4.7 – Modèles de CPG décrits par Mc Crea et Ryback (figures extraites de (Mc Crea et Ryback, 2008)).

excellent moyen pour comprendre à la fois la complexité des mécanismes mis en jeu et la possibilité toutefois d'en proposer des modèles avec un certain degré d'approximation. Je résume ici les grands principes qui régissent ce mécanisme fondamental.

Plasticité hebbienne :

La plasticité hebbienne est le mécanisme le plus important. Différents modèles de plasticité hebbienne existent (plasticité fréquentielle, plasticité impulsionnelle : voir (Hoinville, 2007) pour plus de détails). Je rappelle ici le principe fondamental des modèles parmi les plus intéressants pour la robotique, afin de montrer comment on peut passer d'un principe neurobiologique à un modèle mathématique « implémentable » dans un algorithme en vue de commander un robot.

Le principe général des modèles de la loi de Hebb lie les variations temporelles du poids synaptique (efficacité synaptique) à une fonction paramétrée des sorties (taux de décharges) des neurones pré et post-synaptiques :

$$\tau \frac{dw}{dt} = F(w, \nu_{pré}, \nu_{post}) \quad (4.5)$$

Ainsi, la loi hebbienne fondamentale exprimant la plasticité est :

$$\tau \frac{dw}{dt} = \nu_{pré} \nu_{post} \quad (4.6)$$

Cependant, cette loi ne peut pas être facilement utilisée en pratique, car elle est divergente si elle est appliquée en continu, c'est-à-dire que la valeur des poids synaptiques peut tendre vers une valeur infinie. Pour pallier ce problème, plusieurs auteurs ont proposé d'autres formulations (Grossberg, 1969; Gerstner et Kistler, 2002) en intégrant dans le modèle un terme de limitation de l'efficacité synaptique et un terme d'oubli représentant la dépression synaptique :

$$\frac{dw}{dt} = \tau^{-1}(w_{max} - w)\nu_{pré}\nu_{post} - \tau_{oubli}^{-1}w \quad (4.7)$$

Un autre moyen de limiter les effets de saturation est de prendre en compte le mécanisme de compétition synaptique par des contraintes qui tendent à stabiliser l'efficacité synaptique globale d'un groupe de neurones. Un formalisme possible consiste par exemple à insérer un terme multiplicatif sur le poids synaptique, à modifier ce terme en prenant en compte l'activité totale et la somme des poids synaptiques (équation (4.8)).

$$\tau \frac{dw_i}{dt} = \nu_i \nu_{post} - w_i \frac{\sum_i \nu_i}{\sum_i w_i} \nu_{post} \quad (4.8)$$

Plasticité homéostatique

La plasticité homéostatique, ou homéostasie, est une propriété fondamentale du vivant. Elle existe à tous les niveaux des systèmes biologiques et chez toutes les espèces. C'est la capacité d'un organisme à conserver un équilibre dynamique de fonctionnement face aux perturbations internes ou externes. Les mécanismes homéostatiques exercent donc un rôle de régulation essentiel à la survie des espèces. Pour les réseaux de neurones, et dans une perspective d'application à la commande des systèmes robotiques, nous pouvons prendre en compte deux types de plasticité homéostatique : la plasticité synaptique et la plasticité neuronale intrinsèque.

La plasticité homéostatique des synapses régule l'efficacité synaptique en fonction de l'activité post-synaptique. Ce mécanisme évite la saturation des neurones qui peuvent alors engendrer des signaux incohérents pour la commande. La plasticité synaptique consiste à modifier de manière pondérée la valeur des poids synaptiques en entrée d'un neurone de façon à réguler l'activité de ce neurone (voir figure (4.8)). Lorsque l'une des entrées excitatrices (provenant de neurones pré-synaptiques) d'un neurone post-synaptique est jugée suffisante pour l'activation du neurone post-synaptique, la synapse impliquée est potentialisée par la plasticité hebbienne, résultant de l'activité post-synaptique supérieure et ainsi de suite. Si cette activité post-synaptique trop élevée dure trop longtemps, les mises à l'échelle synaptiques agissent sur toutes les synapses d'entrée proportionnellement à leur poids afin de revenir à une activité normale.

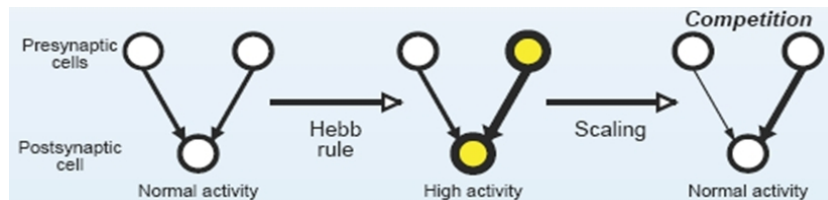


FIGURE 4.8 – Plasticité synaptique (mécanisme de mise à l'échelle). Un neurone post-synaptique reçoit des entrées excitatrices de deux neurones pré-synaptique. Si l'activité post-synaptique est trop élevée, les mises à l'échelle synaptiques agissent proportionnellement à leur poids. Ce mécanisme de concurrence synaptique induit une potentialisation dans cette synapse et une dépression dans l'autre.

La plasticité homéostatique intrinsèque neuronale consiste à réguler l'activité du neurone par la modification de ses paramètres internes (biais, constantes de temps, pente de la sigmoïde). Nous avons par exemple mis en œuvre le principe de la régulation de l'excitabilité du neurone (voir figure 4.9) dans la commande rythmique d'un robot monopode et nous avons montré que ce mécanisme de régulation intrinsèque permet d'améliorer les performances du cycle de marche en terme d'amplitude des mouvements et de robustesse (voir section 6.2.1).

Neuromodulation

Dans la nature, la neuromodulation est provoquée par une sensibilité de certains neurones à des substances, appelées neuromodulateurs, qui entraîne une cascade de réactions chimiques qui vont modifier les caractéristiques de l'activité du neurone post-synaptique ou l'efficacité de ses

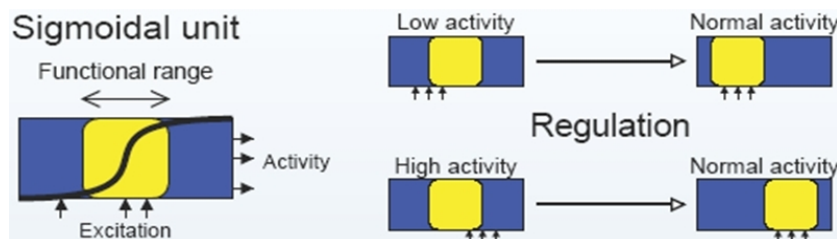


FIGURE 4.9 – Régulation homéostatique de l'excitabilité neuronale. Pour un neurone, la fonction d'activation sigmoïdale lie l'activité totale reçue en entrée synaptique au taux moyen d'activité en sortie. Selon l'activité synaptique reçue, une partie seulement de la fonction d'activation est utilisée. Si cette partie correspond à une activité neuronale trop élevée ou trop basse, la fonction d'activation est déplacée pour revenir à un niveau d'activité « normale » : l'excitabilité est adaptée.

synapses (figure 4.10). Ces substances peuvent être émises par d'autres neurones qu'on appelle neurones modulateurs. Des neuromodulateurs bien connus sont par exemple la dopamine et l'adrénaline qui, par leur impact local au niveau cellulaire, peuvent modifier de manière considérable le comportement global du système nerveux moteur.

Pour la robotique, le mécanisme de neuromodulation est intéressant car il permet de modifier les caractéristiques d'un groupe de neurones en fonction d'un signal particulier provenant par exemple d'un neurone modulateur. Un neurone peut être plus ou moins sensible à un ou plusieurs signaux neuromodulateurs ce qui permet dans un même réseau de neurones d'avoir différents types de sous réseaux. On peut par exemple imaginer que certains dysfonctionnements brusques (amputation de patte, rupture de transmission...) détectés par des capteurs déclenchent l'activation de neurones modulateurs qui modifieront alors le comportement du neurocontrôleur et donc du rythme locomoteur.

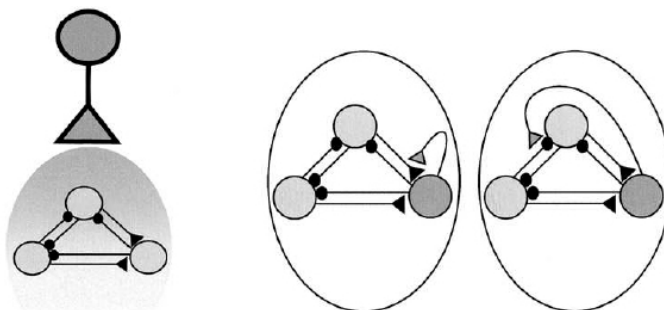


FIGURE 4.10 – Neuromodulation extrinsèque (à gauche) et intrinsèque (à droite). Les neurones classiques sont en gris clairs et les neurones modulateurs en gris foncé (Extrait de (Marder et Bucher, 2001)).

4.3 Modèles de neurocontrôleurs pour la locomotion

4.3.1 Modèle de CPG pour la nage de la lamproie

Dans ses travaux sur la modélisation du contrôle de la nage de la lamproie, Ekeberg propose en 1993 (Ekeberg, 1993) un modèle d'architecture neuronale contrôlant le mouvement rythmique de la lamproie simulée, qu'il appelle modèle neuro-mécanique. Il applique les propriétés observées en biologie sur le système nerveux de l'animal. Son architecture possède des symétries latérales (dont l'intérêt a été démontré depuis dans (Golubitsky *et al.*, 1999)) et pour chaque articulation des segments neuronaux identiques. Enfin, il montre l'importance des retours sensoriels au niveau articulaire dans la robustesse du contrôle de l'animal simulé. Dans cette architecture, (équation 4.9), le neurone de base est inspiré du modèle de Beer :

$$\begin{cases} \dot{\xi}_+ = \frac{1}{\tau_D} (\sum_{i \in \Psi_+} w_i u_i - \xi_+) \\ \dot{\xi}_- = \frac{1}{\tau_D} (\sum_{i \in \Psi_-} w_i u_i - \xi_-) \\ \dot{\vartheta} = \frac{1}{\tau_A} (u - \vartheta) \end{cases} \quad u = \begin{cases} 1 - e^{-(\theta - \xi_+)\Gamma} - \xi_- - \mu\vartheta & (u > 0) \\ 0 & (u \leq 0) \end{cases} \quad (4.9)$$

Dans ce modèle, qui sépare les entrées attachées aux synapses inhibitrices (ξ_-) de celles attachées aux synapses excitatrices (ξ_+), u est la sortie du neurone, θ est son seuil d'activation. Ekeberg a montré l'efficacité et la robustesse de ce modèle de contrôleur neuronal notamment par le rôle des cellules sensorielles qui permettent au CPG de réagir comme une boucle fermée (nage contre un courant perturbateur). Cependant, le contrôleur neuronal final est complexe (plus de 100 segments soit 1000 neurones), il n'inclut pas de mécanisme de régulation ou de plasticité et les valeurs des paramètres neuronaux et synaptiques ont été réglées empiriquement.

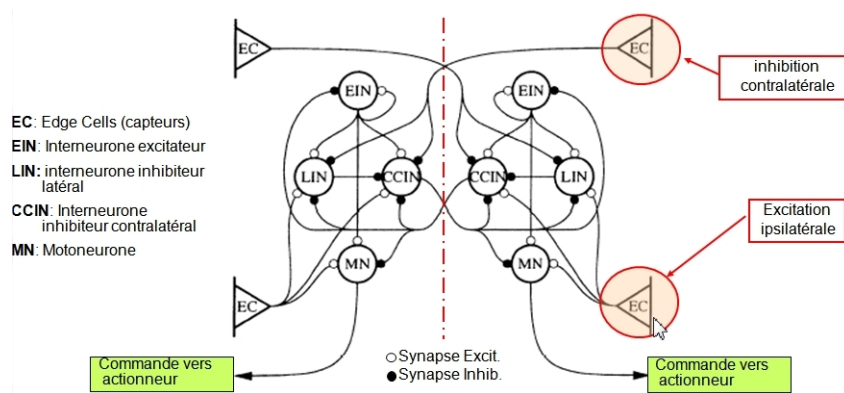


FIGURE 4.11 – Modèle de CPG proposé par Ekeberg pour la nage de la lamproie (figure modifiée de (Ekeberg, 1993)). Ce modèle inclut différents types d'interneurones inhibiteurs et excitateurs, deux motoneurones et des neurones sensoriels articulaires (Edge Cells). On voit bien la symétrie latérale dans la topologie du réseau et dans les interconnexions inhibitrices et excitatrices qui permettent de créer un mouvement rythmique par des actions antagonistes sur les motoneurones. Ce segment neuronal se répète à chaque articulation.

4.3.2 Modèles de CPG pour le contrôle de la marche et de la nage de la salamandre.

Afin de pallier ce problème du calcul de la valeur des paramètres du réseau de neurones, A. J. Ijspeert propose en 2001 (Ijspeert, 2001), d'optimiser ces paramètres par un algorithme génétique. Il applique son approche au contrôle de la marche et de la nage de la salamandre. Le contrôleur global, construit entièrement à la main est en trois niveaux (figure 4.12). Il est constitué de 40 segments oscillateurs (CPG de 8 neurones) qui contrôlent chacun une liaison articulaire. L'optimisation des paramètres neuronaux et synaptiques se déroule en trois étapes successives, du niveau le plus bas, vers le niveau le plus haut. Le contrôleur neuronal final obtenu par cette méthode permet de reproduire les modes de marche et de nage de la salamandre. Cependant, ce modèle est trop complexe pour être implémenté dans une unité de contrôle embarquée sur un « robot salamandre ».

Les CPG sont constitués d'un groupe de neurones interconnectés qui exhibent un comportement équivalent à un oscillateur constituant lui-même un bassin d'attraction pour des lois de commandes stables. Par d'autres travaux sur les oscillateurs non linéaires (Righetti *et al.*, 2006), A. J. Ijspeert propose, pour simplifier l'architecture du neurocontrôleur précédent, d'utiliser un oscillateur non linéaire comme modèle de CPG (Ijspeert *et al.*, 2007). Dans ce modèle (équations 4.10), deux variables (θ_i et r_i) représentent la phase et l'amplitude de l'oscillateur et deux autres (R_i et v_i) permettent de contrôler sa fréquence et son amplitude. Le couplage entre les CPG est défini par des

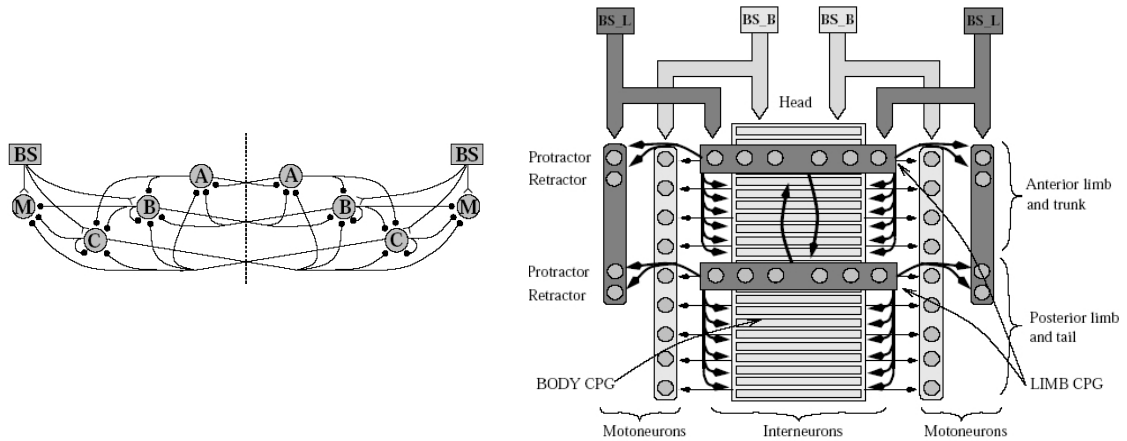


FIGURE 4.12 – Modèle de contrôleur neuronal de la salamandre proposé par A. J. Ijspeert (figures extraites de (Ijspeert, 2001)). Le CPG (à gauche) est constitué de 3 types d’interneurons (A, B, C) et de motoneurons M. Les entrées BS (Brain Stem) provenant du niveau supérieur de la commande activent les neurones. L’architecture du contrôleur neuronal (à droite) est en 3 niveaux.

pois synaptiques (w_i) et des déphasages (ϕ_i). L’architecture du nouveau neurocontrôleur respecte les mêmes règles de construction que celle du précédent. Elle est moins complexe tout en restant biologiquement plausible (voir figure 4.13).

$$\begin{cases} \dot{\theta}_i = 2\pi v_i + \sum_j r_j w_{ij} \sin(\theta_j - \theta_i - \phi_{ij}) \\ \ddot{r} = a_i \left(\frac{a_i}{4} (R_i - r_i) - \dot{r}_i \right) \\ x_i = r_i (1 + \cos(\theta_i)) \end{cases} \quad (4.10)$$

Ce modèle de contrôleur a été implémenté sur un « robot salamandre » réel construit par l’équipe de l’EPFL. Cette équipe a pu montrer ainsi que le réseau de neurones oscillateurs permet au robot de basculer d’un mode de locomotion à l’autre avec des changements de phases et de fréquences, passant de la nage à la marche en reproduisant parfaitement l’allure de l’animal.

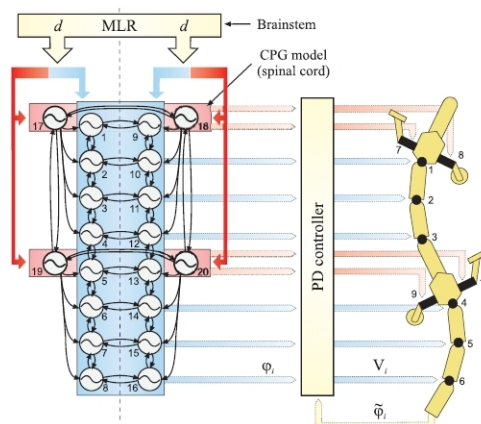


FIGURE 4.13 – Architecture du CPG à base d’oscillateurs non linéaires (figure extraite de (Ijspeert *et al.*, 2007)).

4.3.3 Modèles de CPG pour la marche bipède :

Modèles à bases de neurones :

En 1991, Taga et al. (Taga *et al.*, 1991) proposent un modèle de CPG à base de neurones pour commander la marche et l'équilibre d'un robot bipède plan simulé. Ils utilisent le modèle de Matsuoka (équations 4.2) comme un CPG par liaison articulaire et 2 neurones par CPG, l'un pour le mouvement de flexion, l'autre pour l'extension (figure 4.14). L'architecture de contrôle proposée respecte les principes de modularité (un CPG par liaison articulaire) et de symétrie latérale. L'architecture intègre des retours sensoriels articulaires locaux qui contrôlent l'activité rythmique et d'autres globaux liés au contrôle de l'équilibre. Les auteurs montrent qu'il y a une simultanéité de convergence entre l'équilibre et les trajectoires articulaires composées de cycles limites stables imposés par le CPG. Cependant, le contrôleur global est sensible aux variations de ses paramètres (poids synaptiques difficiles à régler) et ne possède pas de mécanisme de plasticité.

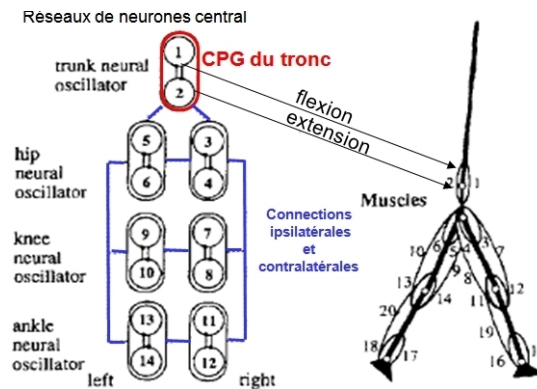


FIGURE 4.14 – Architecture neuronale à base de CPG proposée par Taga et al. (figure modifiée de (Taga *et al.*, 1991)).

Une approche très intéressante sur le contrôle bio-inspiré de la locomotion bipède a été proposée par F. Worgotter (Worgotter *et al.*, 2006). Cette méthode est basée sur les principes du réflexe articulaire couplé à une détection du contact avec le sol pour les phases de transfert et de poussée des jambes (Figure 4.15). Cette approche a été validée sur un petit prototype de robot réel bipède avec de très bonnes performances. Les neurones moteurs sont des neurones dynamiques (modèle de Beer) et les neurones sensoriels sont statiques (modèle proposé par Wadden et Ekeberg (Wadden *et Ekeberg*, 1998)).

Modèles à bases d'oscillateurs non linéaires

L'utilisation d'oscillateurs non-linéaires pour synthétiser des CPG, permet donc de simplifier les architectures neuronales, tout en respectant les principes de la biologie pour la genèse de rythmes locomoteurs. Plusieurs travaux ont été menés dans ce sens, particulièrement sur l'utilisation des oscillateurs de Van der Pol (Dutra *et al.*, 2003) et de Rayleigh (De Pine Filho *et al.*, 2005) mais sans l'introduction de mécanisme adaptatif et sans retour sensoriel lié au contact avec le sol. L'équipe de A. J. Ijseert a exploité d'autres modèles d'oscillateur qu'elle a validé pour la commande de la marche d'un robot bipède (Righetti et Ijspeert, 2006). L'oscillateur utilisé est celui de Hopf (équations 4.11).

$$\begin{cases} \dot{x} = \gamma(\mu - r^2)x - \omega y + \epsilon F(t) \\ \dot{y} = \gamma(\mu - r^2)y + \omega x \\ \dot{w} = -\epsilon F(t) \frac{y}{r} \end{cases} \quad (4.11)$$

Dans ce modèle, w correspond à la pulsation de l'oscillation r à son amplitude, $F(t)$ est le signal d'erreur entre les sorties du CPG et le signal de synchronisation.

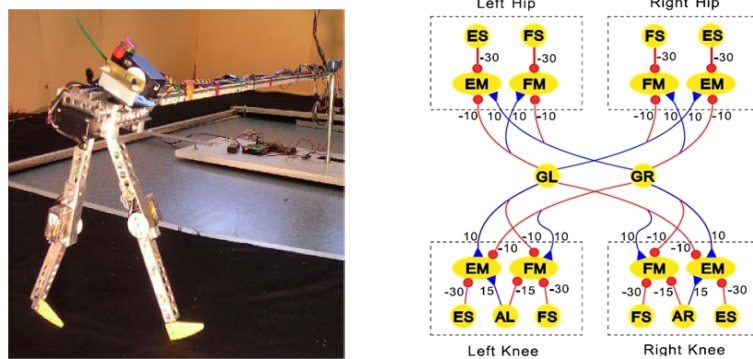


FIGURE 4.15 – Modèle de contrôleur proposé par F. Worgotter et robot bipède utilisé pour valider la méthode. FS et ES sont des neurones sensoriels détecteurs de flexion et d’extension. FM et EM sont les neurones moteurs de flexion et d’extension, GR et GL sont des neurones sensoriels (droit et gauche) détecteurs du contact pied-sol. (Extrait de (Worgotter et al., 2006)).

L’originalité de ce travail réside dans la capacité d’adaptation des oscillateurs grâce à un mécanisme d’apprentissage qui leur donne la propriété de se synchroniser sur un signal périodique. De cette façon, on peut moduler la fréquence et l’amplitude des oscillations du contrôleur et donc modifier l’allure de la marche du robot.

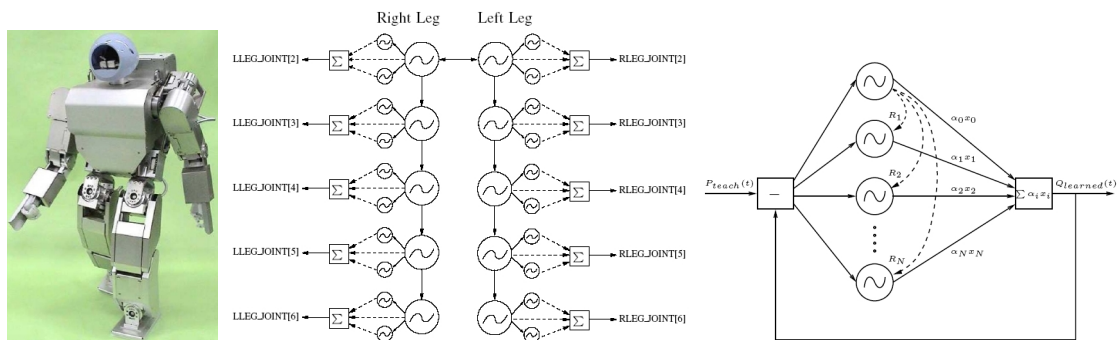


FIGURE 4.16 – Architecture du neurocontrôleur à base d’oscillateurs de Hopf qui a permis de commander plusieurs allures de marche du robot Hoap-2 . La boucle de retour permet d’adapter la fréquence de chaque oscillateur au signal de synchronisation $P_{teach}(t)$ (figures extraites de (Righetti et Ijspeert, 2006)).

Modèle à base d’oscillateurs linéaires :

La solution très intéressante proposée par l’équipe de G. Cheng et al. (Morimoto *et al.*, 2006) consiste à simplifier encore le modèle de contrôleur par l’utilisation de fonctions sinusoïdales. Chacune de ces fonctions est un CPG qui pilote la position angulaire d’une liaison articulaire. Ces oscillateurs sont déphasés entre eux suivant la liaison commandée. Ils sont tous synchronisés à la phase de la dynamique du robot bipède modélisé comme un pendule inverse constitué par son centre de masse et son centre de pression. La phase de la dynamique du robot est mesurée suivant la vitesse \dot{x} du centre de pression par rapport à sa position x selon l’équation (4.12) :

$$\phi_r(x) = -\arctan\left(\frac{\dot{x}}{x}\right) \quad (4.12)$$

Cette fonction peut être assimilée à un neurone sensoriel qui perçoit l’équilibre et dont le modèle a été proposé par Wadden et Ekeberg (Wadden et Ekeberg, 1998). La fréquence d’oscillation de chaque contrôleur dépend de l’erreur entre sa propre phase ϕ_c et celle du robot selon l’équation (4.13) :

$$\dot{\phi}_c^i = w_c + K_c \sin(\phi_r(x) - \phi_c^i + \alpha_i) \quad (4.13)$$

où α_i correspond à différentes valeurs de déphasage entre les liaisons articulaires afin de coordonner les mouvements des jambes. De cette façon il est possible de décrire la position articulaire désirée pour chaque liaison par rapport à la posture courante, en précisant l'amplitude du mouvement. Enfin, une loi supplémentaire de contrôle, qui exprime l'écart entre le roulis mesuré du robot et une valeur maximale admissible, vient moduler l'amplitude des oscillateurs liés aux mouvements dans le plan frontal. Cette méthode a été validée expérimentalement sur le robot humanoïde Sarkos (de taille humaine) et sur le robot humanoïde QRIO de Sony.

4.4 Conclusion

L'objectif de ce chapitre était de présenter succinctement, d'une part les principes fondamentaux et les modèles principaux des mécanismes biologiques mis en œuvre dans les architectures neuronales responsables de la locomotion, et d'autre part des applications concrètes de ces principes à la locomotion robotique. Il apparaît clairement que ces structures et leurs interactions avec le système commandé sont des systèmes dynamiques complexes possédant des couplages internes, ou externes via les boucles sensori-motrices. Ces boucles régulatrices existent à tous les niveaux des architectures nerveuses et mettent en œuvre une grande variété de phénomènes complexes. Néanmoins, depuis de nombreuses années, de grands progrès ont été réalisés sur la compréhension du fonctionnement du système nerveux central.

Cependant, comme l'explique A. J. Ijspeert dans son article de synthèse (Ijspeert, 2008), si nous voulons doter les robots humanoïdes de capacités d'apprentissage de la locomotion, nous devons avancer encore plus loin dans la compréhension du fonctionnement des systèmes locomoteurs et dans leur modélisation. Les travaux des biologistes sur les animaux doivent permettre aux roboticiens de proposer des solutions nouvelles pour les robots. Comme pour un robot, un appareil musculo-squelettique est constitué d'une série de pattes articulées et doit produire des mouvements coordonnés permettant à l'animal de marcher sur un terrain irrégulier, même si des dommages apparaissent sur certains organes moteurs. Une commande centrale est générée et des capteurs permettent une correction en ligne afin d'adapter les mouvements aux contraintes internes ou externe (Marder et Bucher, 2001). Toutefois, les questions sont beaucoup plus difficiles si l'on s'adresse à la marche de systèmes pour lesquels les réseaux de neurones doivent générer des activités motrices complexes et contrôler simultanément des organes articulés et la posture globale du système.

Les travaux que je vais présenter dans la suite de ce manuscrit visent à proposer, pour les robots humanoïdes, des modèles neuronaux capables de produire des mouvements rythmiques adaptatifs pour la locomotion et de contrôler l'équilibre de ces robots. Ma démarche a consisté à explorer plusieurs voies sur la genèse des rythmes locomoteurs et sur le contrôle de l'équilibre en utilisant l'expérimentation et la simulation. Les expériences menées sur le robot ROBIAN du LISV nous ont permis d'expérimenter l'apprentissage en-ligne du contrôle de l'équilibre en position debout et d'étudier par la même occasion l'implémentation temps-réel des modèles neuronaux. En utilisant une simulation dynamique de ROBIAN (sous OpenHRP) nous avons exploré l'utilisation de contrôleurs neuronaux pour implémenter une commande articulaire compliant émulant un muscle et montrer que celle-ci améliore l'équilibre des robots bipèdes pendant la marche. Nous avons d'autre part utilisé la simulation pour explorer, par les algorithmes génétiques, l'efficacité de différents modèles de contrôleurs neuronaux et différents mécanismes adaptatifs pour des robots pouvant posséder un nombre de pattes variable. Enfin, les modèles qui nous ont semblé les plus pertinents ont permis de proposer une architecture sensori-motrice pour l'apprentissage de la marche bipède en simulation.

Pour plus de précisions, le lecteur trouvera en annexe les articles principaux liés aux différents travaux synthétisés dans ce manuscrit.

Chapitre 5

Apprentissage du contrôle de l'équilibre et de la compliance articulaire pour les robots bipèdes

Ce chapitre présente mes travaux sur l'utilisation de méthodes d'apprentissage pour l'amélioration de l'équilibre des robots bipèdes. Dans une première section, une approche expérimentale permet de valider la méthode d'apprentissage par rétropropagation temporelle pour le contrôle de l'équilibre du robot ROBIAN du laboratoire LISV de l'Université de Versailles St Quentin. Dans un second temps un modèle d'apprentissage de comportement musculaire est proposé pour introduire de la compliance dans les articulations de ce robot et améliorer son équilibre pendant la marche en simulation. Enfin, un mécanisme d'apprentissage, basé sur la notion de vigilance par rapport à la chute est introduit et validé sur une simulation du robot bipède de l'Université de Gottingen (Worgotter *et al.*, 2006).

5.1 Mise en œuvre expérimentale d'un contrôleur CTRNN et de la rétropropagation temporelle du gradient

Les travaux que nous avons développés dans le cadre de la thèse de V. Scesa, est une suite logique de mon travail effectué sur l'apprentissage en-ligne d'un robot mobile par la rétropropagation classique (section (3.6)). La thèse avait pour objectif d'évaluer d'une part la capacité des contrôleurs neuronaux de type Continuous Time Recurrent Networks (CTRNN) à apprendre en ligne par la méthode de rétropropagation temporelle du gradient (BPTT) une loi de contrôle d'équilibre, et d'autre part à étudier l'implémentation temps-réel de ces modèles de neurocontrôleurs.

L'intérêt de la rétropropagation temporelle est que, comme la rétropropagation classique, l'apprentissage s'effectue par la minimisation d'une fonction de coût qui exprime mathématiquement la tâche à réaliser. Cette technique d'apprentissage peut donc être rapprochée des méthodes de commande optimale dont les outils théoriques de preuve de stabilité existent, mais n'ont pas encore pu être appliqués à de tels algorithmes d'apprentissage lorsque le contrôleur neuronal possède plusieurs entrées et plusieurs sorties. La validation expérimentale de l'apprentissage de l'équilibre a été effectuée sur le robot ROBIAN du LISV. L'étude des performances des algorithmes a aussi été menée sur ROBIAN et en plus sur le simulateur de route de l'entreprise BIA, entreprise qui a financé la thèse de V. Scesa, (figure (5.1)). Cette machine hydraulique permet de reproduire les effets de la non planéité de la route sur des amortisseurs de véhicule. Sa cinématique est celle d'une plateforme de Steward horizontale dont la partie mobile pèse 680 kg et peut subir des accélérations de 2 g dans toutes les directions. Le robot et cette machine sont pilotés chacun par une baie de commande temps-réel identique.

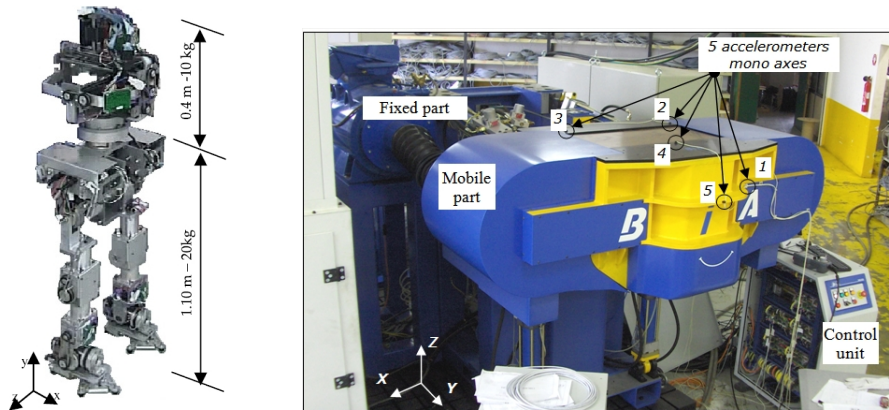


FIGURE 5.1 – Plateformes expérimentales ayant servi à étudier les contrôleurs de type CTRNN avec l'algorithme de rétropropagation temporelle du gradient (BPTT) : à gauche le bipède ROBIAN du LISV, à droite le simulateur de route de la société BIA.

5.1.1 Apprentissage en ligne de l'équilibre pour un robot bipède en posture debout

L'apprentissage de l'équilibre du robot se fait par interaction physique avec l'humain qui vient exercer à la main des efforts de poussée mesurés par un capteur d'efforts situé à la base du torse du robot. En apprenant à déplacer les masses du torse, le réseau de neurones peut compenser les efforts de perturbation et donc contrôler l'équilibre. Une étude approfondie sur l'implémentation temps-réel de ces algorithmes dans une baie de commande industrielle a été réalisée. Ces travaux ont fait l'objet de plusieurs publications dans des conférences internationales (Scesa *et al.*, 2005, 2006a,b) et un article (Hénaff *et al.*, 2011) présenté en annexe (III) dans le journal international *Control Engineering and Practice*.

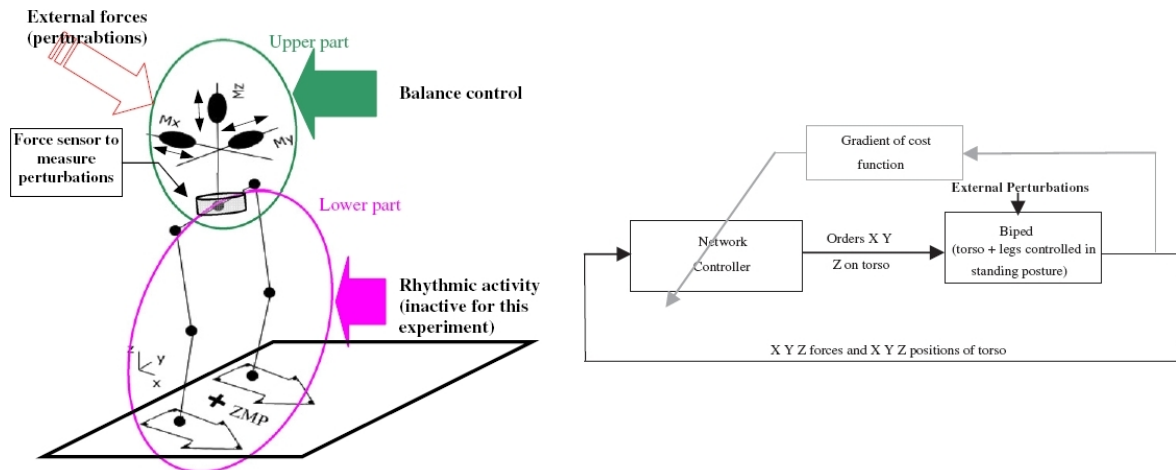


FIGURE 5.2 – Principe de l'apprentissage du contrôle de l'équilibre. A gauche, position du capteur de force entre les membres inférieurs et le torse, à droite schéma d'apprentissage en ligne.

La figure 5.2 présente le principe d'apprentissage mis en œuvre : le CTRNN doit apprendre à commander les masses du torse afin que l'équilibre du bipède soit maintenu. Cet équilibre est déterminé par une fonction de coût égale à la somme quadratique des efforts mesurés par le capteur sur les 3 axes :

$$E = \int_{t_0}^t [F_x^2(t) + F_y^2(t) + F_z^2(t)] dt \quad (5.1)$$

L'algorithme de rétropropagation temporelle permet d'apprendre à minimiser en ligne cette fonc-

tion de coût, quelles que Les modification apportées au réseau CTRNN à chaque itération concernent les poids synaptiques w_{ij} , le biais b_j et la constante de temps T_j attachés de chaque neurone j (équation (5.2)).

$$\Delta w_{jk} = -\eta \frac{\partial E}{\partial w_{jk}} \quad \Delta b_j = -\eta \frac{\partial E}{\partial b_j} \quad \Delta T_j = -\eta \frac{\partial E}{\partial T_j} \quad (5.2)$$

Le détail du calcul du gradient présenté dans l'article (Hénaff *et al.*, 2011) montre que les variations à apporter à ces trois types de paramètres dépendent des états passés du neurocontrôleur et des efforts mesurés par le capteur (équations (5.3)(5.4)(5.5)).

$$\frac{\partial E}{\partial w_{jk}} = \frac{\Delta t}{T_k} \int_{t_0}^t [Z_k(\tau) f'(x_k(\tau - \Delta t)) y_j(\tau - \Delta t)] d\tau \quad (5.3)$$

$$\frac{\partial E}{\partial b_j} = \frac{\Delta t}{T_j} \int_{t_0}^t [Z_j(\tau) f'(x_j(\tau - \Delta t))] d\tau \quad (5.4)$$

$$\frac{\partial E}{\partial T_j} = \frac{-\Delta t}{T_j^2} \int_{t_0}^t [Z_j(\tau) f(x_j(\tau - \Delta t)) - y_j(\tau - \Delta t)] d\tau \quad (5.5)$$

Dans ces expressions, le terme Z_j exprime le gradient attaché à chaque neurone. Il est composé de la somme d'un gradient provenant des poids synaptiques connectés au neurone, d'un gradient provenant directement de la sortie du neurone et d'un gradient provenant de la dynamique propre au neurone. Cette dynamique intrinsèque liée aux variations des constantes de temps étant souvent la cause d'instabilité de l'algorithme d'apprentissage.

Les forces extérieures sont appliquées « à la main » sur le torse du robot de façon aléatoire (figure 5.3), le robot restant asservi dans la position debout, mais sans mouvements sur ses pattes (l'apprentissage de l'équilibre pendant la marche devait être réalisé après la thèse). L'implémentation temps-réel est réalisée en organisant l'algorithme en plusieurs tâches, dont une pour la propagation et une autre pour la rétropropagation, en exerçant une passe d'apprentissage toute les 10 propagations (figure 5.5).

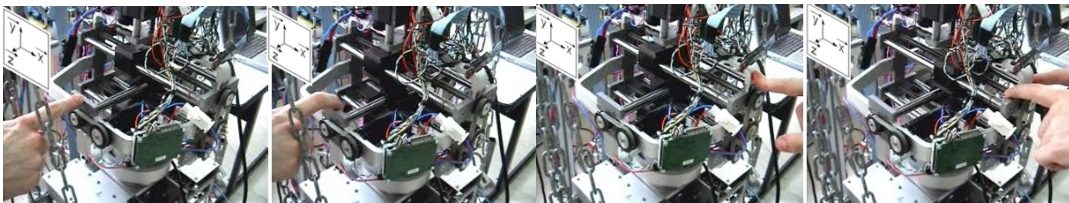


FIGURE 5.3 – Application manuelle des forces extérieures sur le torse de ROBIAN.

Les résultats montrent que l'algorithme d'apprentissage permet au CTRNN dans un premier temps de découpler la commande des masses puis, dans un second temps, de commander chaque masse en fonction de la direction des forces appliquées et mesurées par le capteur. Après l'apprentissage qui dure 2 minutes environ (voir figure 5.4), le CTRNN est capable de compenser les perturbations en réduisant les efforts de près de 50%. L'apprentissage en ligne par l'algorithme BPTT d'une loi de contrôle de l'équilibre est donc possible avec ce type de contrôleur. Cependant, la validation expérimentale de l'apprentissage durant la marche doit encore être réalisée.

5.1.2 Etude des performances temps-réel de la BPTT et améliorations

L'étude de l'implémentation temps-réel nous a permis d'évaluer les limites de l'algorithme BPTT en termes de stabilité et de rapidité de convergence. Cet algorithme étant en effet très délicat à stabiliser, une exploration de ses limites de fonctionnement sur la baie électronique temps-réel BIA a été menée : influence de la taille de la fenêtre temporelle, expression de la constante de temps des neurones, mémorisation des poids synaptiques.

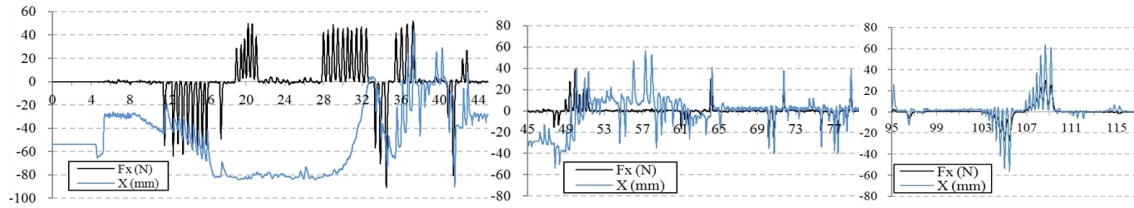


FIGURE 5.4 – Apprentissage de l'équilibre (extrait des enregistrements pour l'axe X uniquement). Au cours du temps le CTRNN apprend à commander le mouvement des masses pour compenser les forces mesurées (Hénaff *et al.*, 2011).

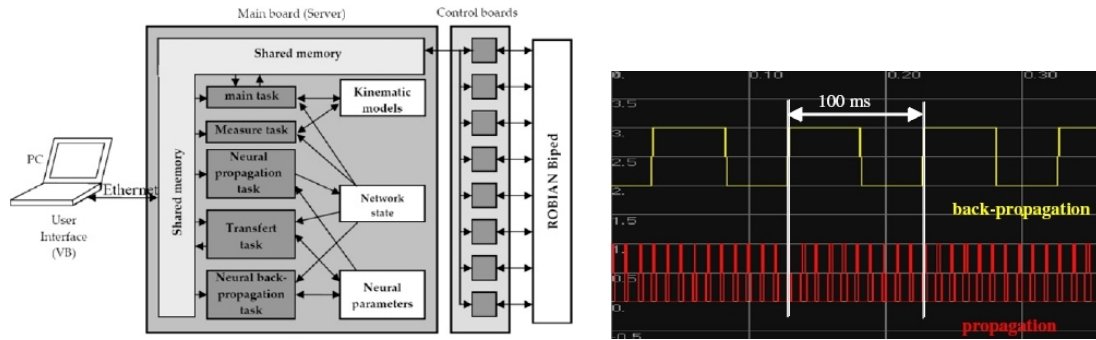


FIGURE 5.5 – Architecture temps-réel et ordonnancement propagation/rétropropagation (extrait de Scesa (2006)).

L'étude consiste à identifier le comportement dynamique entre les déplacements du ZMP sur le sol et la commande des masses du torse du robot. Un fichier d'apprentissage et un fichier de test ont été construits (figure (5.6)), en appliquant différents types de mouvements aux masses du torse : échelons d'amplitudes différentes mais de fréquence constante et sinusôides à amplitude constante mais de fréquence variable.

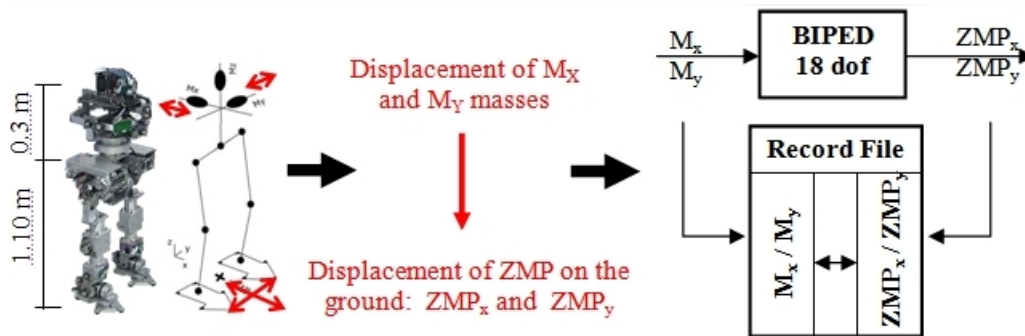


FIGURE 5.6 – Mise en œuvre expérimentale pour l'évaluation des performances de l'algorithme BPTT par l'identification de l'équilibre de ROBIAN. Les mouvements des masses du torse induisent des variations sur la position du ZMP au sol, ce qui permet de construire les fichiers d'apprentissage et de test.

Nous avons ainsi mis en évidence les conditions d'apparition d'instabilités de la BPTT suivant les paramètres de l'algorithme d'apprentissage : pas d'apprentissage, longueur de la fenêtre temporelle, mémorisation des poids le long de cette fenêtre, modèles de variation des constantes de temps des neurones (figure 5.7). De nombreuses expériences identiques nous ont permis d'élaborer des statistiques basées sur deux indices de performances : l'un reflétant la rapidité de convergence de la BPTT, l'autre le taux d'erreur sur le fichier de test. Cette étude statistique a permis de comparer la version classique et la version qui calcule l'apprentissage en tenant compte de l'historique de la valeur des poids synaptiques le long de la fenêtre temporelle. Sur la figure 5.8 qui montre les

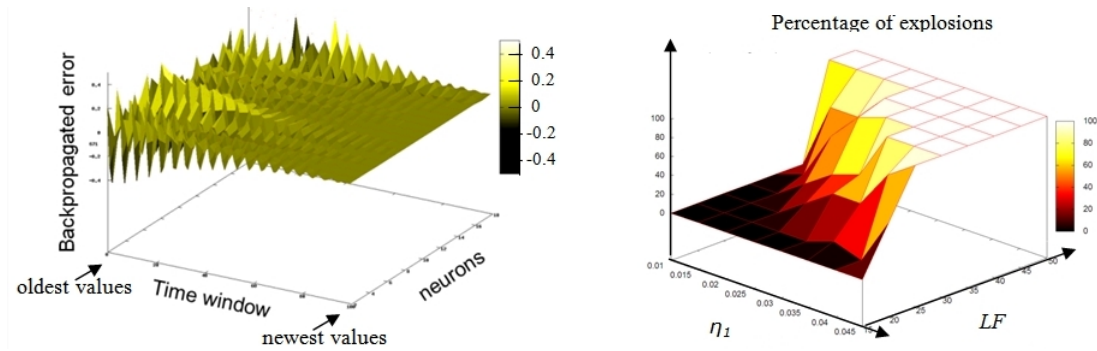


FIGURE 5.7 – Mise en évidence de l’instabilité de la BPTT due à une explosion du gradient le long de la fenêtre temporelle. A gauche phénomène d’explosion qui fait diverger l’algorithme, à droite influence du pas d’apprentissage et de la taille de la fenêtre sur cette divergence (extraits de (Scesa, 2006)).

résultats, on constate bien qu’en mémorisant la valeur des poids synaptiques le long de la fenêtre temporelle et en modifiant les équations d’apprentissage en conséquence, la taille de la fenêtre peut être augmentée permettant un apprentissage plus rapide et plus précis (Scesa *et al.*, 2006b).

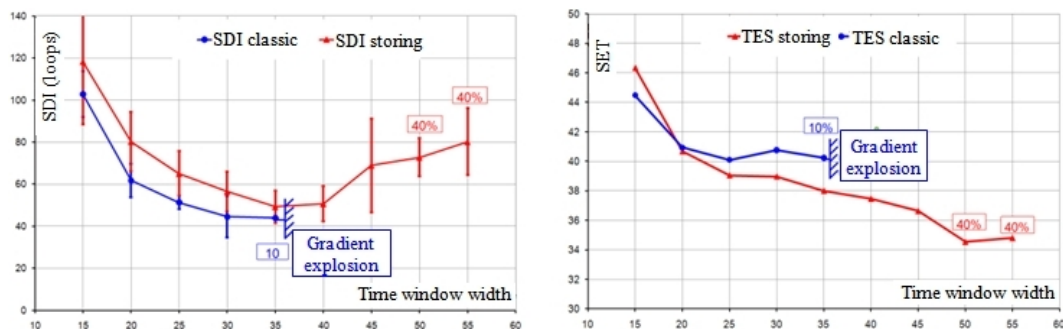


FIGURE 5.8 – Influence de la taille de la fenêtre temporelle sur la stabilité de l’algorithme BPTT. Les indicateurs SDI et TES correspondent à la qualité de l’apprentissage, respectivement la rapidité de convergence et le taux d’erreur sur un fichier de test. Les pourcentages expriment le nombre d’essais qui ont convergé lorsque tous n’ont pas convergé (extraits de (Scesa, 2006)).

5.1.3 Influence de la dynamique neuronale sur la stabilité de l’apprentissage

Enfin, nous avons montré aussi que la dynamique neuronale exprimée mathématiquement par la constante de temps des neurones a une grande influence sur la dynamique d’apprentissage. En effet, lorsqu’un algorithme d’apprentissage modifie cette constante de temps, la dynamique du neurone n’est pas équilibrée suivant les valeurs de la constante de temps. Ainsi plus la constante de temps diminue et plus le neurone sera sensible aux variations ce qui tend à déstabiliser l’apprentissage. A l’inverse, plus la constante de temps augmente, plus le neurone est lent et l’exploration s’en trouve ralentie. Quand on remplace la constante de temps linéaire par une expression sigmoïdale (solution proposée par (Nguyen et Cottrell, 1997)), l’analyse montre que les variations de la constante de temps ont un effet équilibré sur toute l’échelle du neurone, ce qui permet d’explorer pleinement la dynamique neuronale (voir figure 5.9 et en annexe (Scesa *et al.*, 2006a)).

Enfin, l’algorithme d’apprentissage modifié a été utilisé pour identifier en ligne le comportement dynamique du simulateur de route de la société BIA dont on peut voir l’extrait d’un essai expérimental figure (5.10). L’expérience a consisté en des essais dynamiques et harmoniques, par

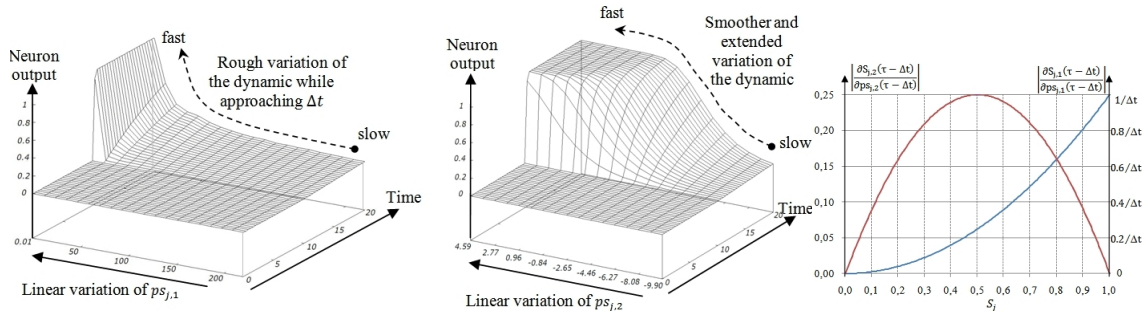


FIGURE 5.9 – En haut : mise en évidence du phénomène de limitation de la dynamique dans les CTRNN en fonction des variations de la constante de temps calculées par l'algorithme d'apprentissage. En bas, le gradient lié aux variations de la constante de temps joue comme un facteur d'amplification dans le gradient rétropropagé. On voit bien que si l'expression de la constante de temps est linéaire (en bleue), ce facteur est inversement proportionnel à la période d'échantillonnage du modèle numérique et tend vers l'infini. Si l'expression est sigmoïdale (rouge), ce facteur est équilibré sur toute l'échelle de la constante de temps (extraits de (Scesa, 2006)).

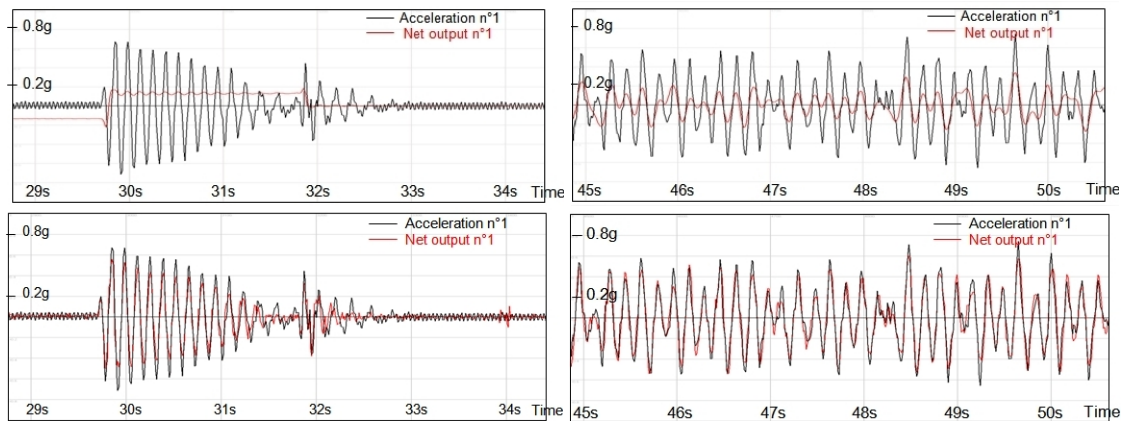


FIGURE 5.10 – Apprentissage en ligne de l'identification du comportement dynamique du simulateur de route de la société BIA : début d'apprentissage (en haut) et fin (en bas) après 300 boucles de rétropropagation temporelle sur la base d'apprentissage. Les accélérations sont celles mesurées sur la partie mobile dont la masse est 680 kg (ici une seule accélération est montrée)(extraits de (Scesa, 2006)).

application d'une série de consignes en échelon de position sur les verins hydrauliques ainsi qu'une série de sinusoides de fréquence variable. L'entrée du CTRNN est constituée des positions linéaires de chacun des six verins, la sortie des cinq accélérations mesurées (les accéléromètres sont placés selon la figure 5.1), la couche cachée possède quinze neurones. Ainsi, la stabilité et la rapidité de l'algorithme d'apprentissage BPTT ont été testées pour la commande de deux systèmes robotisés très différents dans leur dynamique et dans leur nature d'actionnement (électrique et hydraulique).

5.1.4 Conclusion

Ces expériences menées sur l'apprentissage en ligne de l'équilibre du robot bipède se voulaient, dans un premier temps, être découplées du problème de la marche. Cependant, dans une perspective à moyen terme de validation de l'apprentissage de l'équilibre pendant la marche, nous avons travaillé sur un algorithme de genèse de mouvement rythmique inspiré de l'analyse de la marche humaine. Après une étude descriptive approfondie du cycle de la marche humaine (figure 5.11), nous avons transposé à la structure cinématique de ROBIAN cette description en la codant sous forme d'une machine à états. Les résultats obtenus sont très intéressants car cette approche a permis d'obtenir en simulation sous OpenHRP un cycle de marche stable et rapide avec des transitions stables entre le

phases d'arrêt et de redémarrage du robot et pour différentes vitesses de marche (voir (Serhan *et al.*, 2008b)). Cette approche ne sera pas présentée dans ce document pour des raisons de clarté et de cohérence liée à l'approche neuronale mais le lecteur trouvera les détails dans les articles présentés et (Serhan *et al.*, 2008a,c) en annexe. Dans le but d'améliorer les performances de cet algorithme sur la stabilité de la marche, nous avons en même temps étudié l'intérêt que peuvent avoir les contrôleurs neuronaux pour apporter de la compliance dans le contrôle articulaire. L'ensemble de ce travail effectué dans le cadre de la thèse de Hayssam Serhan est résumé dans la section suivante.

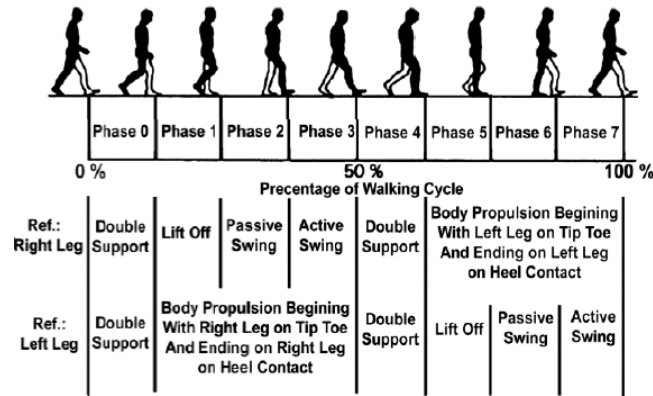


FIGURE 5.11 – Décomposition du cycle de la marche humaine en 8 phases élémentaires (Serhan *et al.* (2008b,a,c)).

5.2 La compliance articulaire améliore l'équilibre durant la marche (simulation).

La rigidité des asservissements articulaires est un obstacle à l'amélioration de la marche des robots humanoïdes. En effet, on sait maintenant que les actionneurs classiques, tels que les moteurs électriques à courant continu ne sont pas optimisés pour l'actionnement articulaire de ces robots dont les mouvements sont d'amplitude faible et souvent effectués à faible vitesse, comparative-ment aux robots manufacturiers. Actuellement, beaucoup de travaux concernent la recherche de nouveaux actionneurs électriques (Ben-Ahmed et Multon, 2007), pneumatiques (muscle de Mc Kibben), hydrauliques (Alfayad *et al.*, 2009), à base d'électro-polymères, ou d'actionnements hybrides (Shin *et al.*, 2010). On se référera à (OHalloran et OMalley, 2004) pour une revue complète sur la question. D'autres méthodes consistent à incorporer dans les structures articulaires des robots une certaine souplesse par des artifices mécaniques (compliance mécanique) (Mc Geer, 1990; Sentis *et al.*, 2010) ou par une loi d'asservissement articulaire adaptative (compliance contrôlée) (Lee et Gonzalez, 2008).

C'est cette dernière approche que nous avons choisi d'explorer pour la thèse de H. Serhan (Serhan, 2009). Nous avons proposé d'apprendre l'adaptation des gains des contrôleurs PID des moteurs à courant continu de telle façon que la réponse de l'articulation soit proche de celle qu'elle aurait si elle était contrôlée par un muscle (figure 5.12).

Nous avons d'abord indentifié un modèle de muscle par un réseau de neurones de type *Neural Network AutoRegressive eXogenous* (NNARX) (Serhan *et al.*, 2006). L'identification par NNARX consiste à optimiser les poids du réseau de neurones par une méthode ARX en utilisant des mesures passées comme entrées du réseau. Puis ce modèle a été utilisé comme référence pour l'apprentissage d'un perceptron qui adapte en ligne les gains proportionnel, intégral et dérivé de la boucle d'asservissement du moteur à courant continu de chaque genou du robot bipède ROBIAN (figure (5.13)). Nous avons d'autre part évalué la structure d'apprentissage avec un autre réseau de neurones qui apprend le modèle de l'actionneur lorsque celui-ci est supposé inconnu.

L'algorithme de rétropropagation du gradient est utilisé pour former le perceptron multicouches chargé d'adapter les gains du PID de la boucle d'asservissement de l'articulation. L'originalité du travail vient de l'algorithme lui-même. En effet, la sortie du perceptron est directement la valeur des

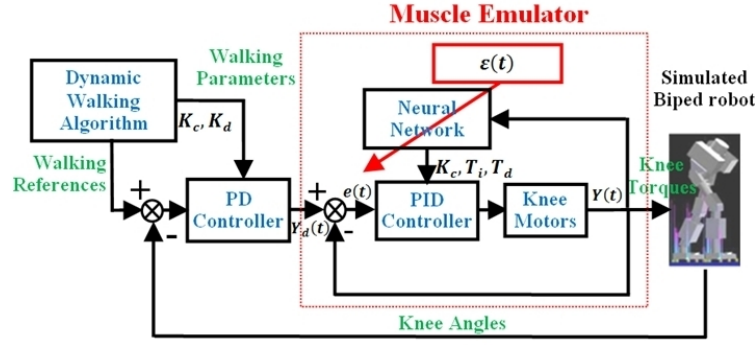


FIGURE 5.12 – Approche du contrôle articulaire compliant par réseau de neurones. L'émulateur de muscle adapte les gains articulaires à la dynamique de la consigne. L'algorithme de marche dynamique est basé sur une machine à états (Serhan *et al.* (2008b,a,c)).

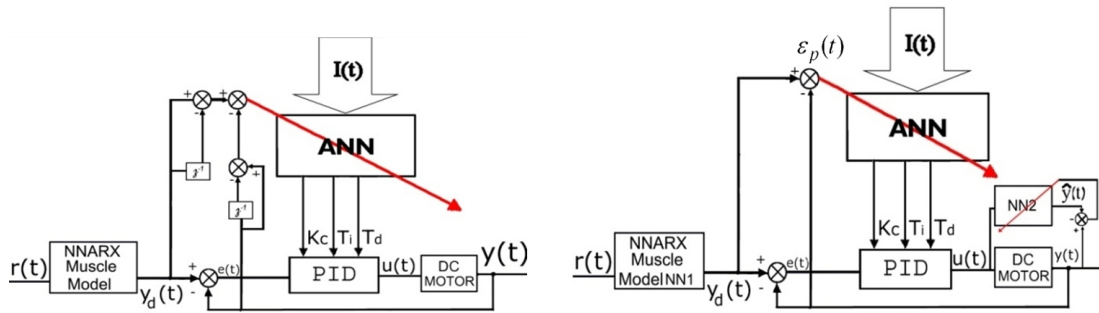


FIGURE 5.13 – Schéma de la structure d'apprentissage choisie pour l'émulateur de muscle. A gauche avec un modèle mathématique d'actionneur, à droite en apprenant un modèle neuronal de l'actionneur. $y(t)$ est le couple moteur, $y_d(t)$ est le couple du modèle de muscle (consigne), $r(t)$ est le signal de commande, et $I(t) = [e(t), e(t-1), e(t-2), \rho(t)]$ un vecteur composé de trois échantillons du signal d'entrée $e(t)$ et d'un signal image du sens de variation de $r(t)$.

gains proportionnel, intégral et dérivé de cette boucle (figure (5.14)). Comme la fonction de coût de l'apprentissage est la variation de l'erreur entre la sortie du moteur et la sortie désirée appliquée par modèle de muscle, les équations de rétropropagation montrent que le gradient rétropropagé peut être facilement calculé en prenant en compte le modèle du PID et le modèle de l'actionneur. En effet, si on prend comme fonction de coût :

$$\xi(t) = \frac{1}{2} \varepsilon_v^2(t) = \frac{1}{2} [\Delta y_d(t) - \Delta y(t)]^2 \quad \text{avec} \quad \Delta = 1 - z^{-1} (\text{opérateur dérivation})$$

Alors le gradient de cette fonction par rapport aux poids synaptiques peut s'écrire :

$$\frac{\partial \xi(t)}{\partial W_{jk}} = \frac{\partial \xi}{\partial \varepsilon_v} \frac{\partial \varepsilon_v}{\partial \Delta_y} \frac{\partial \Delta_y}{\partial \Delta_u} \frac{\partial \Delta_u}{\partial O_k} \frac{\partial O_k}{\partial W_{jk}} \quad (5.6)$$

Dans cette expression, tous les termes sont ceux habituels à l'algorithme, sauf le terme $\frac{\partial \Delta_y}{\partial \Delta_u}$ qui dépend du modèle de l'actionneur, et $\frac{\partial \Delta_u}{\partial O_k}$, qui dépend de la loi d'asservissement de l'actionneur. Ces deux gradients sont aisément calculables et permettent donc de réaliser l'apprentissage du perceptron afin que celui-ci fournisse à chaque instant les gains optimaux en fonction de l'erreur. Si le modèle de l'actionneur n'est pas disponible, on peut l'identifier par un autre réseau de neurones (figure (5.13)) au travers duquel la rétropropagation permet de calculer le gradient équivalent à celui obtenu par le modèle (pour plus de détail lire (Serhan *et al.*, 2010) en annexe (III)).

Nous avons validé cette méthode d'adaptation des gains des boucles d'asservissement à partir d'un modèle de muscle et en fonction de la consigne, dans le simulateur de ROBIAN sous OpenHRP. L'émulateur a été implémenté sur l'articulation du genou de chaque jambe. Les courbes de la figure 5.15 comparent les trajectoires articulaires de l'humain (ramenées à la dimension de ROBIAN),

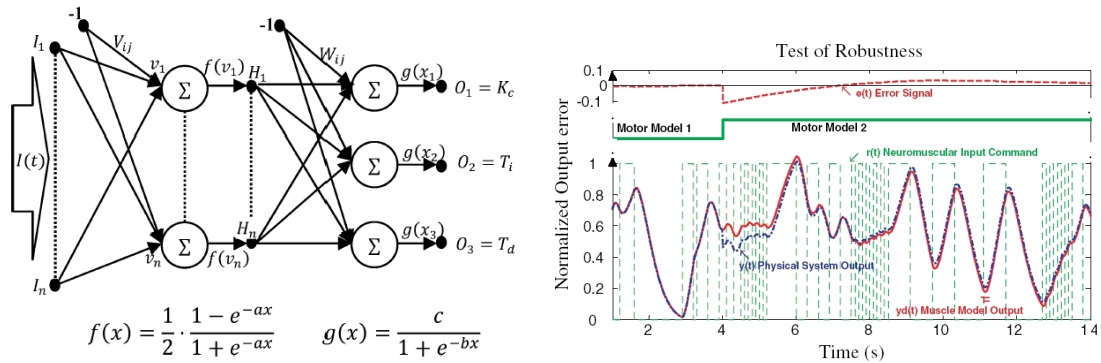


FIGURE 5.14 – A gauche : Structure du perceptron chargé de calculer les gains du PID de la boucle d’asservissement articulaire. A droite comparaison de la réponse du moteur DC avec le modèle de muscle, effet de l’apprentissage de ce perceptron en cas de variation de la dynamique du moteur.

avec celle de ROBIAN doté ou non de l’émulateur musculaire. On voit bien l’effet bénéfique de la compliance du genou sur les trajectoires articulaires du robot qui se rapproche plus de celles de l’homme. Les différences pour la hanche et surtout pour la cheville pendant les phases 1 et 2 proviennent de la rigidité de la semelle du robot qui contraint les mouvements articulaires. Des expériences réalisées, dans le cadre du projet PHEMA, à l’Hopital Raymond-Poincaré de Garches (Equipe du Pr. Bussel), ont montré que lorsqu’on contraint la semelle de l’homme à être rigide, les trajectoires de la cheville ont presque la même allure que celle du robot (Serhan, 2009).

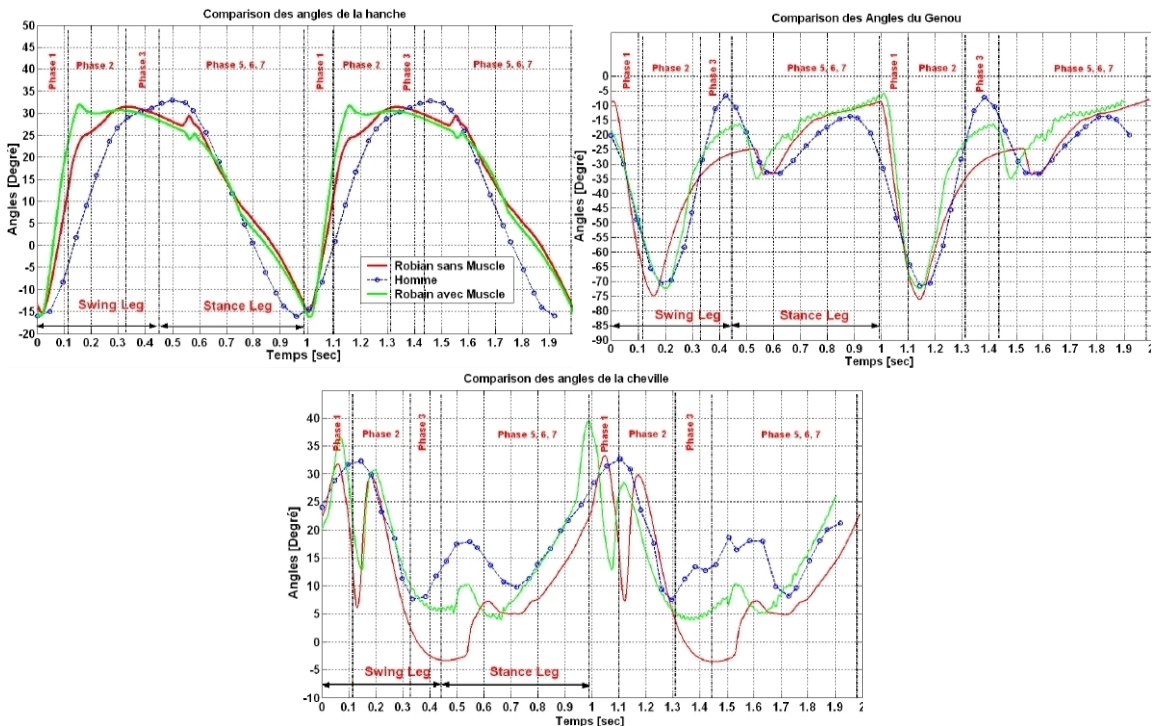


FIGURE 5.15 – Comparaison des trajectoires articulaires entre l’humain (marche moyenne d’un enfant de la taille de ROBIAN) et ROBIAN (jambe droite) avec et sans émulateur de muscle : angles de la hanche, du genou, et de la cheville pour deux cycles de marche. La vitesse du robot est 0.6 m/sec. Les phases indiquées sont celles de la figure (5.11) issues de la décomposition du cycle de la marche humaine.

Nous avons aussi montré en simulation que cette boucle d’asservissement qui émule une réponse musculaire permet d’améliorer la stabilité de la marche du robot bipède face à des perturbations

extérieures telles que des poussées latérales ou frontales, comme le montre la figure 5.16. La figure compare ainsi les trajectoires du robot lorsqu'il est soumis à des poussées dans le plan frontal. On constate une plus grande robustesse de la marche avec la compliance puisque le robot ne tombe pas après avoir été poussé. L'article présenté en annexe (III) et publié dans *International Journal of Neural Systems* décrit bien cette approche et les résultats obtenus.

Enfin, nous avons montré que la puissance totale consommée par les moteurs est réduite de façon importante par la compliance qui joue le rôle d'amortisseur en réduisant la raideur d'asservissement sur certaines parties du cycle de marche (Serhan, 2009).

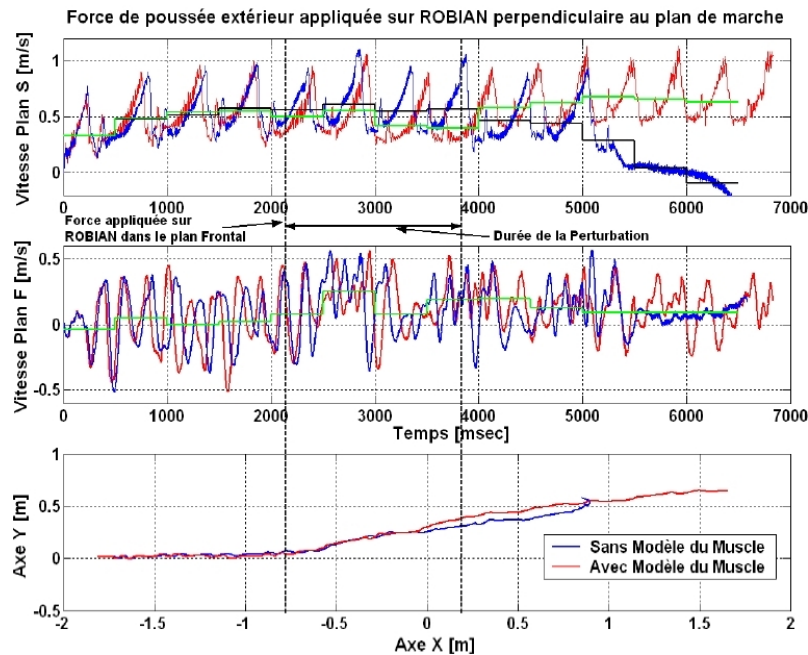


FIGURE 5.16 – Etude comparative du contrôle de l'équilibre avec et sans compliance contrôlée sur les genoux du robot bipède. Vitesses et position de ROBIAN dans le plan sagittal, après application d'une force de poussée perpendiculaire au sens de la marche (plan frontal). La courbe bleue correspondant à la configuration sans compliance montre la chute du robot : en haut la vitesse de la marche devient nulle, en bas la trajectoire est interrompue.

5.3 La dynamique neuronale articulaire influence la stabilité de la marche

Si l'on simule l'expérience menée par Worgotter et al. (figure 4.15) on constate que la robustesse du neurocontrôleur qu'il a proposé dépend de la dynamique des neurones, particulièrement ceux de la hanche. Nous avons donc exploré par la simulation le domaine de stabilité du contrôleur de la marche du bipède de Worgotter en fonction du paramètre dynamique (gain de la sigmoïde α_{Hip}) et de la sensibilité (extension maximale articulaire θ_{HipMax}) du neurone sensoriel extenseur *ES* de la hanche (équivalent au récepteur de Stretch en biologie). Les résultats montrent que suivant les valeurs de ces paramètres, le contrôleur ne peut pas assurer une marche stable et le robot tombe (figure 5.17). L'équilibre du robot est donc lié à l'allure de la marche, l'amplitude du pas dépendant de l'extension de la hanche, et sa vitesse du gain du neurone).

Dans le cadre de sa thèse qui est en cours, nous avons montré avec John Nassour que pour assurer une meilleure robustesse du contrôleur de Worgotter, une solution consiste à insérer un mécanisme de sélection de ce couple de paramètres en fonction d'une perception de l'équilibre du robot qui peut être perturbé par la pente variable du sol (Nassour *et al.*, 2009). La notion de vigilance (Ahn et Picard, 2005) permet d'introduire ce mécanisme de sélection de l'allure lorsqu'elle est couplée à une méthode d'apprentissage et de classification du risque de chute. En effet, en perturbant le robot par des variations aléatoires de la pente du sol, on peut déterminer l'ensemble des couples

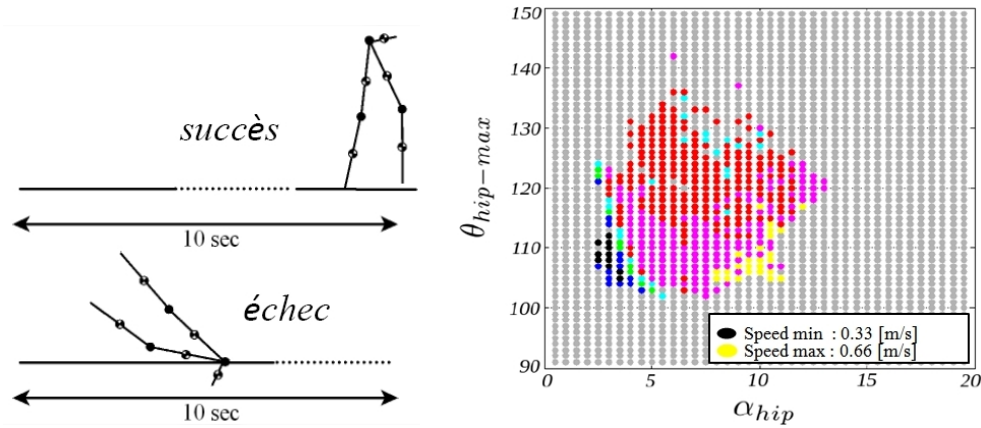


FIGURE 5.17 – Mise en évidence de l'instabilité du contrôleur proposé par F. Worgotter en fonction des paramètres du neurone sensoriel extenseur *ES* de chaque hanche. À gauche, critère utilisé pour valider ou non la stabilité de la marche. À droite, en gris : la dynamique contrôleur/robot est instable et le robot tombe. En couleur, cette dynamique est stable et la vitesse de la marche (exprimée ici par les couleurs) dépend de la valeur des paramètres.

$(\alpha_{Hip}, \theta_{HipMax})$ qui maintiennent l'équilibre et l'ensemble des couples qui engendrent la chute. On peut par la suite sélectionner le bon couple, ou une valeur qui s'en approche en fonction de la pente du sol et du risque que l'on souhaite prendre, c'est-à-dire résoudre le dilemme entre devoir marcher en risquant de tomber et ne pas marcher. Selon le seuil de vigilance, qui dénote la tolérance aux risques, on peut décider d'une allure qui prend des risques ou d'une autre qui n'en prend pas. Le mécanisme se rapproche de celui identifié en sciences cognitives qui joue le rôle d'alerte prévisionnelle dans le cerveau pour éviter de refaire les mêmes fautes. Ce mécanisme situé dans la région du Cortex Cingulaire Antérieur (Brown et Braver, 2005) pourrait participer à la sélection des actions en fonction des expériences passées et en fonction des résultats attendus (Matsumoto *et al.*, 2003).

Le mécanisme d'apprentissage proposé par John Nassour est basé sur ce principe d'apprentissage à partir de retours négatifs (la chute) et de retours positifs (la marche). Ce mécanisme est composé de deux phases : l'évaluation et la prise de décision (figure 5.18). Pour la phase d'évaluation, deux cartes auto-organisatrices de Kohonen sont utilisées pour apprendre en fonction de la pente γ du sol. L'un apprend les couples $(\alpha_{Hip}, \theta_{HipMax})$ qui amènent à l'échec (chute), et l'autre les couples qui amènent au succès (marche). La tolérance au risque joue un rôle important dans un tel mécanisme d'apprentissage. La prise de décision est basée sur un mécanisme d'alerte précoce qui permet d'éviter de répéter les erreurs passées. L'apprentissage est effectué sur 1500 essais de marche avec une pente du sol aléatoire. Le mécanisme d'apprentissage évalue la distance entre le vecteur d'état du système $v = [\alpha_{Hip}, \theta_{HipMax}, \gamma]$ et chaque carte. Selon la valeur du seuil de vigilance le mécanisme de prise de décision décide d'appliquer la marche avec ces valeurs $(\alpha_{Hip}, \theta_{HipMax})$, ou décide de rechercher un nouveau couple de valeurs. Une fois l'apprentissage effectué, le mécanisme identifie la distance entre l'état du système $(\alpha_{Hip}, \theta_{HipMax}, \gamma)$ et chaque carte et en déduit, selon le seuil de vigilance, le bon couple $(\alpha_{Hip}, \theta_{HipMax})$ des neurones sensoriels des hanches. Les résultats ont montré des différences dans la capacité d'apprentissage selon que le comportement accepte la prise de risque (i.e. la chute) ou l'évite. Ce travail est détaillé dans l'article (Nassour *et al.*, 2009) fourni en annexe de ce document. Les résultats montrent que les cartes de succès et d'échec apprennent mieux avec un seuil de vigilance plus tolérant au risque. Cela aboutit à un contrôleur plus robuste face à des variations de pente aléatoires comme le montre la figure 5.19.

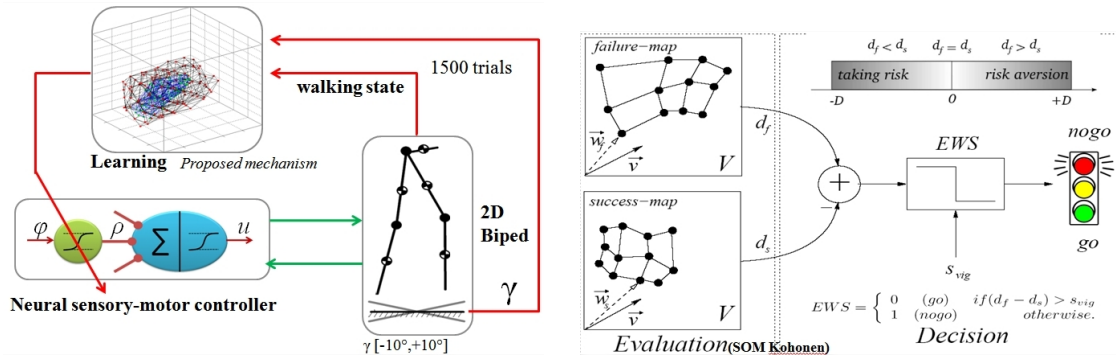


FIGURE 5.18 – Mécanisme d'apprentissage de la vigilance avec un système d'alerte prévisionnelle (EWS : Early Warning System). A gauche, l'apprentissage est effectué sur 1500 essais de marche avec une pente du sol aléatoire. A droite, le mécanisme d'apprentissage évalue la distance entre le vecteur d'état du système et chaque carte. Un changement de la pente conduit à faire basculer le système de décision entre les neurones pour assurer la marche.

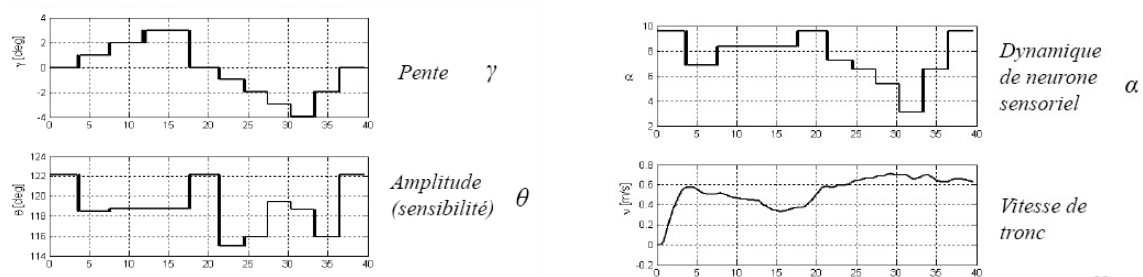


FIGURE 5.19 – Mise en évidence de l'efficacité du mécanisme d'adaptation : les paramètres des neurones sensoriels sont adaptés à la pente.

5.4 Conclusion

Les travaux présentés dans ce chapitre ont mis en évidence l'intérêt des contrôleurs neuronaux et des techniques d'apprentissage pour résoudre certains aspects du contrôle de la marche et de l'équilibre des robots bipèdes. Nous avons montré que l'algorithme de rétropropagation temporelle permet un apprentissage en ligne de l'équilibre qui est robuste et que, malgré sa complexité, il est possible de l'implémenter dans des systèmes d'exploitation temps-réel embarqués. Nous avons d'autre part montré que les techniques d'apprentissage peuvent fournir des modèles de contrôleurs bas niveau qui permettent d'obtenir une compliance articulaire qui améliore la robustesse de la marche des robots bipèdes tout en minimisant l'énergie dissipée par les actionneurs. Nous avons montré enfin que la dynamique du modèle de neurone utilisé a une influence importante sur la stabilité de la marche du robot et qu'il est essentiel d'introduire des mécanismes d'apprentissage qui adaptent cette dynamique en fonction des perturbations subies par le robot.

Les modèles de neurocontrôleurs utilisés dans ce chapitre ne possèdent pas les propriétés fondamentales des contrôleurs biologiques que nous avons présentées au chapitre 4, ni dans leur architecture, ni dans leurs propriétés de rythmicité, ni dans leur adaptivité intrinsèque. En effet, pour commander la marche et l'équilibre des robots humanoïdes par des contrôleurs bio-inspirés, nous devons proposer des modèles de contrôleurs qui possèdent ces propriétés fondamentales.

Le chapitre suivant est consacré à l'étude de ces modèles. Afin d'explorer la diversité des solutions, ceux-ci sont optimisés par évolution artificielle pour commander différents types de robots à pattes (monopode, hexapode, myriapode). L'évaluation des neurocontrôleurs est effectuée suivant différents types de scénarios qui correspondent à des situations de perturbation ou de dommage : poussées ou frottements lents ou brusques, blocages articulaires, amputations... .

Chapitre 6

Modèles de contrôleurs neuronaux adaptatifs pour la genèse de rythmes locomoteurs

Ce chapitre présente des travaux qui visent à répondre aux questions précédentes sur la plausibilité biologique des modèles de contrôleurs neuronaux. En partant du modèle CTRNN, nous montrons qu'il est possible de construire à la main ou par synthèse génétique des contrôleurs adaptatifs et robustes pour la locomotion, puis que des modèles de neurones rythmiques apportent des solutions très avantageuses pour la maîtrise des rythmes locomoteurs. Les modèles proposés sont appliqués dans un premier temps à la marche des robots à pattes (monopode, hexapode, myriapode) en simulation, car pour ces types de robot la question de l'équilibre est moins fondamentale, et les outils de simulation permettent alors de mieux évaluer et de tester les comportements adaptatifs par des scénarios variés. Dans un second temps, nous verrons que les résultats obtenus sur les robots myriapode aboutissent à des solutions très intéressantes qui peuvent être appliquées à la marche des robots bipèdes.

6.1 Synthèse manuelle de contrôleurs rythmiques pour les robots hexapodes

La synthèse manuelle consiste à construire, dans un premier temps, des blocs fonctionnels neuronaux capables de reproduire certains mécanismes biologiques élémentaires, puis d'associer ces blocs pour produire des rythmes locomoteurs stables et moduler ces rythmes en fonction des retours sensoriels. Dans un second temps, des interactions entre ces blocs par des connexions nouvelles et des mécanismes adaptatifs sont définis afin d'obtenir un contrôleur très évolué. C'est une méthode souvent longue et délicate qui n'explore pas l'ensemble des solutions et qui nécessite une maîtrise parfaite des outils connexionnistes. Nous avons mené avec Gildas Bayard le travail présenté dans cette section lors de sa thèse au LIP6, que je co-encadrai avec J. A. Meyer dans le cadre du projet IRON (thèse commencée en 2001 et interrompue en 2004).

Les résultats montrent que le contrôleur neuronal peut apprendre à s'adapter à la dynamique du robot afin qu'il se déplace efficacement et de façon fluide. Les caractéristiques des contrôleurs sont liées aux connexions qui peuvent être récurrentes, excitatrices ou inhibitrices, au neurone qui peut être dynamique et posséder des paramètres adaptatifs conformément aux modèles biologiques.

A l'image des architectures biologiques, le contrôleur est constitué d'un bloc neuronal par patte (figure 6.1). Ces blocs sont tous connectés entre eux afin de synchroniser les mouvements des pattes, et sur ces blocs se projette le signal de niveau supérieur *BS* (Brain Stem). Le principe de construction du contrôleur repose sur une décomposition du cycle d'une patte de l'hexapode pour une marche en tripode alterné. Chaque patte se trouve alternativement soit en contact avec le sol, soit en transfert (voir figure 6.1). Lors de la phase de contact avec le sol, la patte se déplace vers l'arrière, créant la poussée qui déplace le corps vers l'avant. L'entrée du réseau est une excitation

dont l'amplitude est une image de la vitesse désirée de la plate-forme. Les sorties du réseau sont les deux consignes de position articulaire de chaque patte (2 liaisons sont mécaniquement couplées).

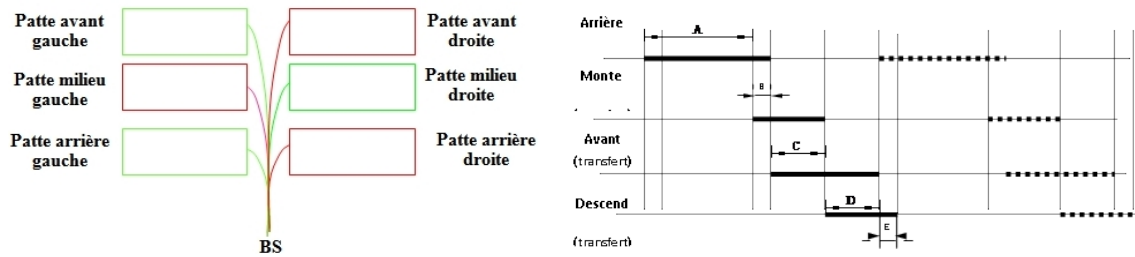


FIGURE 6.1 – Structure modulaire du contrôleur global de l'hexapode et chronogramme du cycle de la marche d'une patte de l'hexapode. Lors de la phase de transfert la patte n'est pas en contact avec le sol, elle revient en avant en se soulevant, puis en se baissant suivant 4 sous-séquences (B : perte de contact avec le sol, C : transfert montant, D : transfert descendant vers l'avant, E : descente et prise de contact avec le sol).

6.1.1 Modélisation des blocs fonctionnels neuronaux :

Les blocs fonctionnels neuronaux sont construits en couplant judicieusement les neurones (modèle de Beer) par des liaisons synaptiques excitatrices. La figure 6.2 montre les oscillateurs qui sont à la base du contrôleur complet. La figure 6.3 montre le centre générateur de rythme (CPG) pour une patte. La figure 6.4 montre le contrôleur global de l'hexapode. Chaque contrôleur de patte est synchronisé à l'autre via des « neurones π » dont le rôle est de moduler (par une opération de multiplication) les signaux issu d'une patte amont en fonction des signaux de la patte aval.

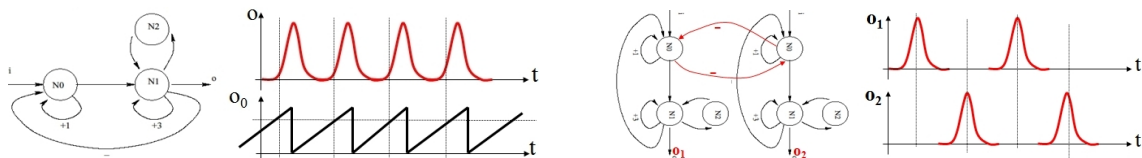


FIGURE 6.2 – Générateurs de rythmes. A gauche : oscillateur basique à 1 sortie. A droite : oscillateurs à 2 sorties en opposition de phase.

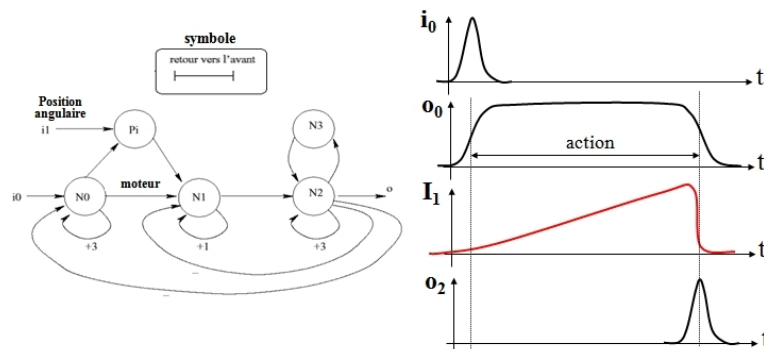


FIGURE 6.3 – Bloc fonctionnel commandant le mouvement vers l'avant d'une articulation. L'entrée I_0 est l'impulsion qui déclenche le mouvement, la sortie O_0 correspond à l'action, c'est-à-dire à la vitesse angulaire durant le mouvement, la sortie O_2 annonce la fin du mouvement, I_1 est le signal image de la position angulaire.

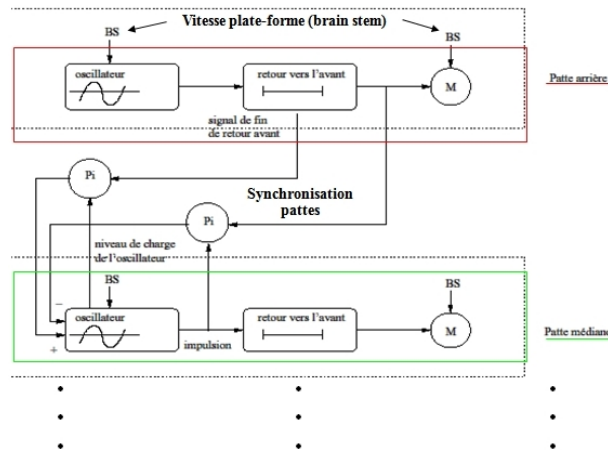


FIGURE 6.4 – Structure du contrôleur complet pour le cycle de marche de l'hexapode

6.1.2 Modèle de mécanismes adaptatifs

L'introduction de mécanismes adaptatifs permet d'adapter le cycle locomoteur à la taille du robot hexapode et à sa dynamique. Ils modifient les poids synaptiques en fonction notamment de l'état de la patte par rapport aux 4 phases du cycle locomoteur de la figure 6.1 et par rapport à celui de l'autre patte liée au mode tripode alterné. Par exemple, les poids seront modulés afin d'augmenter la fréquence du cycle d'une patte si celle-ci reste trop longtemps en butée. Pour calculer simplement les poids synaptiques, l'idée consiste à les initialiser suivant 4 valeurs qualitatives :

- inhibition normale = valeur négative de valeur absolue inférieure à 1,
- inhibition forte = valeur négative de valeur absolue supérieure à 1,
- excitation forte = valeur positive supérieure à 1,
- Excitation normale = valeur positive inférieure à 1.

Les mécanismes adaptatifs sont implantés suivant les quatre séquences de la phase de transfert de la patte décrits figure 6.1. Ces mécanismes, qui vont régler finement les poids en ligne et donc augmenter ou diminuer la sensibilité de chaque neurone concerné dans le cycle de la patte, sont des règles qualitatives du type :

```

Si (butéeAvant) {augmenter(fréquenceC); augmenter(fréquenceD);}
Si ((finD) et (!pasEnAvant)) {diminuer(fréquenceC); diminuer(fréquenceD);}
Si (ContactPendantDescente (D)) et (avantFin(E))
    {diminuer(fréquenceC); augmenter(fréquenceD);}
Si ((!contact) et (finE)) {augmenter(fréquenceC); diminuer(fréquenceD);}
Si ((contactBonSens) et (finE)) {augmenter(fréquenceE);}
Si ((!contactBonSens) et (finE)) {diminuer(fréquenceE);}

```

Les résultats obtenus en simulation (figure 6.5) montrent que le cycle de la marche est au début incohérent, puis devient coordonné. En effet, pendant les premiers pas (500 premières itérations), les 6 pattes sont en phase (figure 6.6) et le robot « tombe » contre le sol, puis les déphasages apparaissent, mais le cycle reste incohérent et peu efficace, la vitesse du robot est variable et discontinue. Grâce aux mécanismes adaptatifs, la fréquence d'activation des pattes est modifiée en fonction de la durée de blocage de la patte en butée (figure 6.7). Le cycle de chaque patte étant aussi lié à celui des autres pattes, l'adaptation locale d'une patte provoque une convergence globale du cycle des 6 pattes. Finalement, la marche devient coordonnée comme le montre la figure 6.6 et la vitesse du robot devient stable et continue.

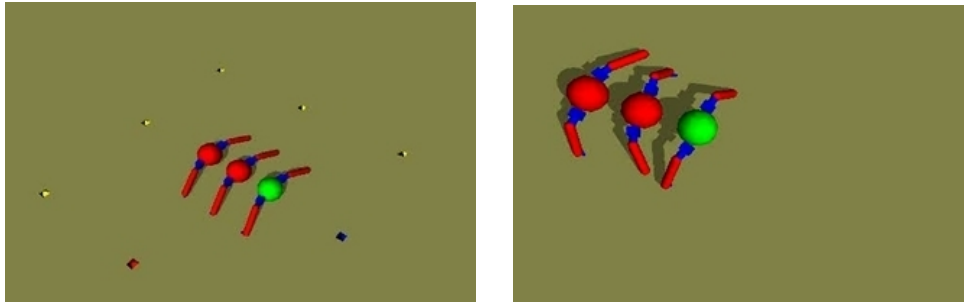


FIGURE 6.5 – Simulation dynamique de l'hexapode sous ODE (Open Dynamic Engine : <http://www.ode.org/ode.html>)

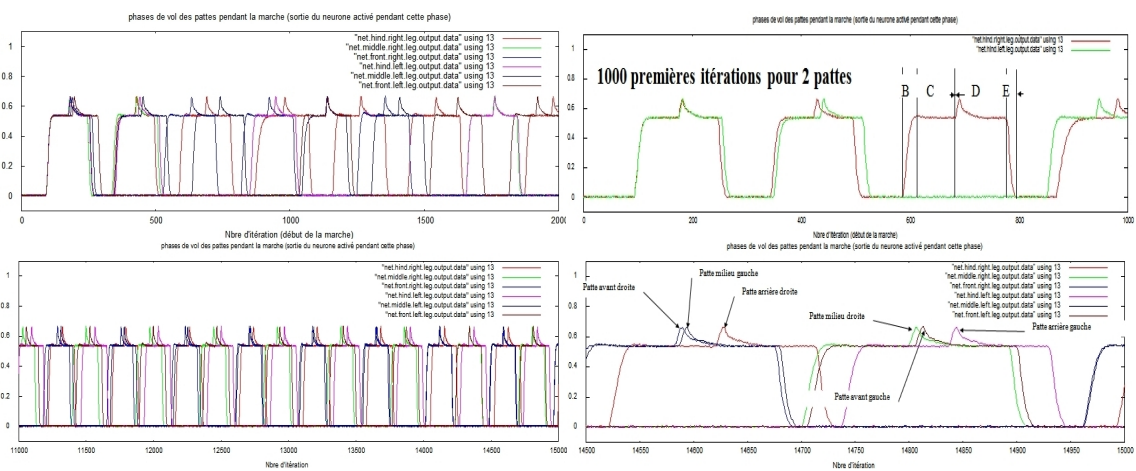


FIGURE 6.6 – Amplitude des mouvements des pattes lors du cycle de marche de l'hexapode. En haut, début du cycle et zoom sur 2 pattes où l'on aperçoit que l'une d'elle reste inerte entre les itérations 600 et 800. En bas cycle de marche après adaptation du contrôleur neuronal. On constate que le cycle tripode alterné est parfaitement obtenu (zoom à droite).

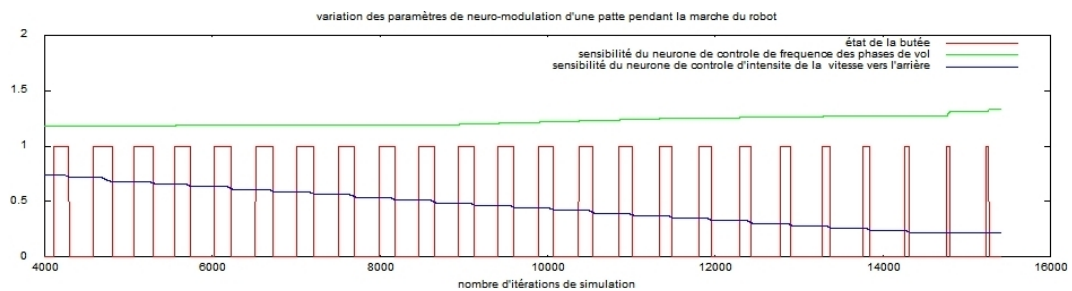


FIGURE 6.7 – Variation des paramètres de neuro-modulation d'une patte durant la marche du robot. Le temps pendant laquelle la patte reste en butée diminue car la sensibilité du neurone chargé de la fréquence du cycle augmente tandis que celui chargé du contrôle de la vitesse diminue. L'efficacité du cycle de la marche augmente et la coordination des pattes apparaît.

Ces simulations montrent l'efficacité, au niveau global, des mécanismes de régulation intrinsèques locaux. Ces mécanismes permettent en effet, en ajustant les paramètres synaptiques, de faire converger le cycle global vers une rythmicité stable et robuste, après un régime transitoire lié à au couplage dynamique entre le contrôleur neuronal et la dynamique du robot. Ce travail a montré par ailleurs la complexité et la difficulté de la méthode de synthèse manuelle de tels contrôleurs sans savoir si le contrôleur finalement obtenu est optimal. Cette question de la recherche de l'optimalité du contrôleur et de ses mécanismes intrinsèques est abordée dans la section suivante.

L'utilisation des algorithmes évolutionnistes permet en effet d'évaluer l'efficacité des contrôleurs neuronaux et des mécanismes adaptatifs qui leur sont incorporés.

6.2 Algorithmes génétiques pour l'évaluation de contrôleurs neuronaux plastiques

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 essaie donc de reproduire ce que la nature a su faire. Cependant, les travaux les plus significatifs (Gallagher *et al.*, 1996; Nolfi, 1997; Ijspeert *et al.*, 1998; Meyer *et al.*, 2002; Doncieux, 2003) 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 quel que 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, paramètres des neurones, types de synapses, règles d'adaptation. . .). C'est une technique d'optimisation avantageuse et puissante (Doncieux, 2010) car elle permet d'explorer automatiquement l'ensemble de l'espace de recherche et d'identifier des solutions efficaces qui n'auraient peut être 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.

Le travail présenté dans cette section que nous avons mené avec Thierry Hoinville dans le cadre de sa thèse, s'est inscrit dans le projet IRON soutenu par l'ex-programme ROBEA 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. Nous avons étudié la robustesse de différents types de 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 ces trois types de réseaux de neurones à complexité biologique croissante et dont les paramètres sont calculés ou non par évolution, nous mettons en évidence l'efficacité de deux mécanismes homéostatiques observés en biologie.

6.2.1 Mise en évidence de l'importance de la dynamique neuronale sur l'efficacité du rythme locomoteur

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 6.8) et son contrôleur peuvent être vu comme l'un des six constituants du robot hexapode (cela est souvent le cas dans les travaux qui cherchent à s'inspirer de la nature pour construire des contrôleurs de robots à pattes (Shinkichi *et al.*, 2006)).

Le réseau de neurones doit calculer les 3 consignes de position articulaire à appliquer à la patte de façon à engendrer 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. Le robot est simulé sous O.D.E. avec un modèle d'asservissement, basé sur un contrôleur PID, équivalent en boucle fermée à un système du second ordre correctement réglé. L'algorithme génétique est basé sur un modèle générationnel qui utilise la bibliothèque GA_Lib du MIT. Le nombre de générations est fixé à 1000 pour 200 individus par génération (voir (Hoinville *et al.*, 2007) pour plus de précisions).

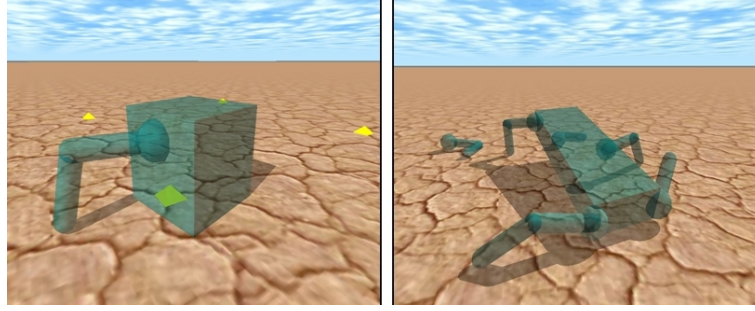


FIGURE 6.8 – Robot à une patte et le modèle d’hexapode correspondant lorsqu’il est amputé de 2 pattes

Nous avons comparé trois modèles de réseaux de neurones. L’architecture de ces 3 réseaux est identique, 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 afin de reproduire un mécanisme homéostatique. 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 conductivité. Enfin, le biais θ_i du neurone correspond au seuil d’activation de la membrane, w_{ij} est le poids synaptique de la connexion du neurone j vers le neurone i , N_i est le nombre de synapses entrantes du neurone i .

Les neurones du premier réseau sont classiques, sans comportement dynamique. Seuls les poids w_{ij} et le biais θ_i de chaque neurone sont calculés par l’évolution :

$$y_i = \sum_{j=1}^{N_i} w_{ij} o_j \quad \text{et} \quad o_i = \text{sigmo}(y_i + \theta_i) \quad (6.1)$$

Les neurones du second réseau sont des neurones qui possèdent une dynamique du premier d’ordre avec une non-linéarité liée à la fonction sigmoïde. C’est le modèle de CTRNN (Beer, 1995a,b, 2006a,b) très répandu.

$$y_i = -\tau_i \frac{dy_i}{dt} + \sum_{j=1}^{N_i} w_{ij} o_j \quad \text{et} \quad o_i = \text{sigmo}(y_i + \theta_i) \quad (6.2)$$

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 CTRNN mais chaque biais est déterminé par un mécanisme homéostatique appelé Center-Crossing (Boonyanit et Beer, 2002) qui 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 :

$$y_i = -\tau_i \frac{dy_i}{dt} + \sum_{j=1}^{N_i} w_{ij} o_j \quad o_i = \text{sigmo}(y_i + \theta_i) \quad \theta_i = -\frac{\sum w_{ij}}{2} \quad (6.3)$$

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

La figure 6.9 montre que les stratégies adoptées ne sont pas identiques selon le type de réseau synthétisé. Ainsi, celle des perceptrons récurrents standards consiste à positionner le bras en appui sur le sol et à appliquer un mouvement de faible amplitude à grande fréquence avec des consignes saturées. Les vibrations ainsi obtenues permettent de déplacer le corps du robot mais, dans la plupart des cas, le robot avance sur une faible distance puis s’arrête. Lorsqu’on remplace les neurones classiques par des neurones dynamiques, l’amplitude des mouvements est importante mais la commande reste saturée. Le bras se soulève lentement mais régulièrement pour reprendre appui

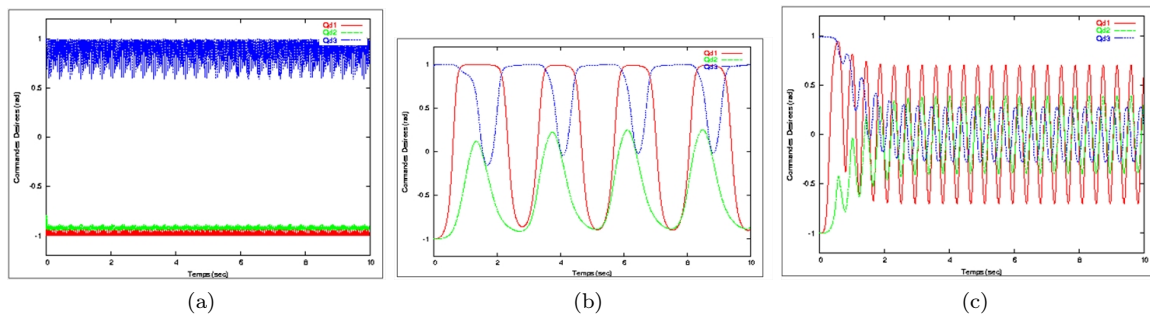


FIGURE 6.9 – Comparaison des commandes articulaires calculées par le meilleur neurocontrôleur d’une population : a) perceptron récurrent standard, b) perceptron récurrent avec neurones dynamiques de Beer. Dans les deux cas, les biais sont calculés par l’évolution. c) même réseau avec neurones de Beer, mais le biais est calculé par un mécanisme homéostatique.

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 un mécanisme homéostatique, applique une commande oscillante de fréquence plus importante que le second contrôleur. Les oscillations ne sont pas saturées et le régime transitoire qui précède le régime permanent montre la stabilité du contrôleur.

Sur la figure 6.10 (à gauche) on compare l’efficacité 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 un temps limité à dix secondes. On perçoit nettement l’efficacité des neurones dynamiques sur les neurones standards et l’efficacité du mécanisme de régulation dans le calcul des biais. On voit que les neurones classiques ne génèrent pas de cycle locomoteur sur la commande tandis que les neurones dynamiques permettent eux, de créer un cycle locomoteur qui se maintient à l’infini après les dix secondes. Lorsqu’on incorpore le mécanisme homéostatique, on observe en début du cycle, un régime transitoire puis une commande du robot à une vitesse importante et stable. La distance parcourue est finalement presque 3 fois plus longue que pour les deux premiers modèles.

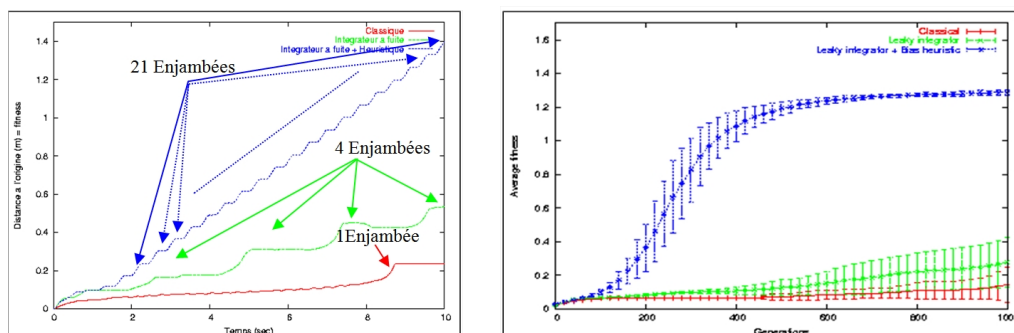


FIGURE 6.10 – Fonction d’évaluation pour les 3 types de réseau. L’indice de performance est la distance parcourue par le robot en 10s. A gauche : meilleur contrôleur d’une population. Chaque enjambée est visible par une variation rapide de la distance parcourue. A droite : évolution de la valeur moyenne de la fonction d’évaluation d’une population (1000 générations, 200 neurocontrôleur). L’écart type est calculé sur 10 expériences.

La qualité des résultats obtenus peut être analysée à travers la diversité des solutions trouvées par dix expériences d’évolution différentes (figure 6.10). On constate nettement que les réseaux de neurones dynamiques permettent d’obtenir une valeur de la fonction d’évaluation plus importante en fin d’évolution avec un faible écart type. De plus, le mécanisme homéostatique 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 le

mécanisme de régulation accélère fortement l'évolution en la focalisant sur une zone de l'espace de recherche où les solutions sont déjà efficaces grâce à ce mécanisme, rendant ainsi robuste la population de solutions.

Il est donc intéressant d'incorporer dans les neurones des mécanismes de régulation homéostatique plutôt que de laisser l'algorithme génétique calculer les valeurs optimales de chaque paramètre. 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. 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. La section suivante présente cette approche qui consiste à incorporer des mécanismes adaptatifs pour les neurones et les synapses.

6.2.2 Genèse de mouvements rythmiques stables pour la locomotion monopode

En robotique évolutive, les modèles plastiques de réseaux neuronaux s'avèrent prometteurs pour l'évolution des comportements adaptatifs. Nous avons vu précédemment que l'algorithme évolutionniste peut être 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 ». En particulier, il a été démontré que les neurocontrôleurs incorporant des synapses adaptatives sont utiles pour résoudre des tâches complexes nécessitant de la flexibilité. Toutefois, ces contrôleurs évolués peuvent présenter une instabilité du comportement, notamment lorsque la durée de la simulation est prolongée au-delà de la limite utilisée pour l'évolution, c'est-à-dire que le réseau de neurones « ne sait pas » contrôler le robot pour des situations non observées durant l'évolution, c'est-à-dire qu'il n'est pas adaptatif.

Nous montrons ici comment des modèles de neurones plastiques inspirés par les mécanismes homéostatiques présentés à la section (4.2.3) (figures 4.9 et 4.8), peuvent devenir des générateurs de rythmes flexibles et stables pour la commande des robots à pattes. Ces mécanismes 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. Ces modèles sont appliqués à la commande d'un robot monopode glissant sur un rail et soumis à des perturbations extérieures qui s'opposent à son avancement par des frottements variables (figure 6.11). L'article correspondant à ce travail (voir en annexe (Hoinville *et al.*, 2011)) démontre en détails l'intérêt d'incorporer dans les contrôleurs neuronaux ces lois de plasticité (lire aussi (Hoinville et Henaff, 2004a,b)).

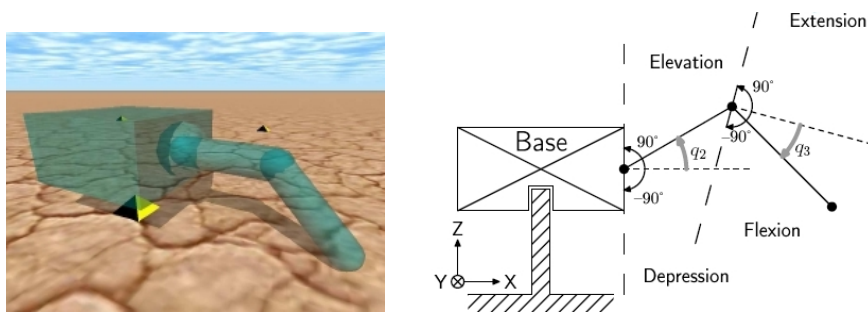


FIGURE 6.11 – Morphologie et débattements articulaires du robot monopode glissant sur son rail.

Les modèles de mécanismes homéostatiques sont de deux types. Le premier, introduit précédemment et dit *Center Crossing* a été proposé par (Mathayomchan et Beer, 2002). Il régule l'excitabilité du neurone en adaptant le seuil d'activation de chaque neurone en fonction de l'efficacité de ses afférences synaptiques, c'est-à-dire de ses poids, selon la loi :

$$\theta_i = \frac{\sum w_{ij}}{2} \quad (6.4)$$

Le second mécanisme, dit *synapses normalisées* et proposé par (Gerstner et Kistler, 2002) tend à stabiliser l'efficacité synaptique globale de chaque neurone, c'est-à-dire à normaliser la valeur totale de ses poids synaptiques :

$$\left\| \sum_{j=1}^{N_i} w_{ij} \right\| = 1 \quad (6.5)$$

En combinant ces deux mécanismes homéostatiques et les quatre lois d'apprentissage hebbien (équations 6.6), proposées dans (Floreano et Urzelai, 1999, 2000) (règle de Hebb, règle post-synaptique, règle pré-synaptique, règle de covariance). On obtient une diversité de choix dans le modèle de contrôleur.

$$\frac{d|w_{ij}|}{dt} = \begin{cases} \text{loi de Hebb simple :} & (1 - |w_{ij}|)o_j o_i \\ \text{loi pré-synaptique :} & (1 - |w_{ij}|)o_j o_i + |w_{ij}| o_j (o_i - 1) \\ \text{loi post-synaptique :} & (1 - |w_{ij}|)o_j o_i + |w_{ij}| (o_j - 1)o_i \\ \text{loi de covariance :} & \begin{cases} (1 - |w_{ij}|)\delta(o_j, o_i) & \text{si } \delta(o_j, o_i) > 0 \\ |w_{ij}| \delta(o_j, o_i) & \text{sinon} \end{cases} \end{cases} \quad (6.6)$$

En effet, les deux mécanismes aboutissent à quatre modèles de neurones (modèle classique, modèle avec *center crossing*, modèle avec *synapses normalisées*, et modèle avec *center crossing* et *synapses normalisées*), tandis que pour chaque synapse, ce sera l'une des quatre lois d'apprentissage qui sera sélectionnée par l'algorithme génétique. L'architecture du neurocontrôleur est par contre imposée : en sortie, les trois consignes angulaires articulaires ; en entrée l'erreur en vitesse du corps du robot par rapport à une consigne et l'information binaire de contact du pied contre le sol. Chaque neurone est connecté à tous les autres y compris à lui-même. Plusieurs scénarios imposés au robot pour l'évaluation du contrôleur sont déterminés par la forme du signal de consigne en vitesse et par la valeur de la perturbation (tableau 6.1). La fonction d'évaluation est la somme quadratique des trois erreurs moyennes en vitesse calculée pour chaque scénario.

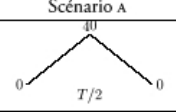
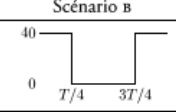
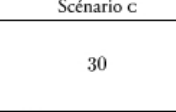

	Scénario A	Scénario B	Scénario C
$V_d(t)$ [cm/s]			
$k_{fr}(t)$ [kg/s]	10	10	

TABLE 6.1 – Trois scénarii d'évaluation des contrôleurs du monopode : différentes formes de consignes et différentes valeurs de frottements visqueux contre l'environnement.

Les résultats comparatifs montrent (figure 6.12) que les contrôleurs contraints par ces mécanismes homéostatiques sont plus performants en termes d'évolutivité et de stabilité que ceux qui sont non contraints. L'analyse du meilleur contrôleur évolué montre l'adaptabilité et la robustesse découlant des propriétés de plasticité incorporées. En effet, en appliquant à ce contrôleur différents profils de vitesse désirée du robot, 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 de neurones de suivre la consigne en vitesse. Les courbes des figures 6.13 et 6.14 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. 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

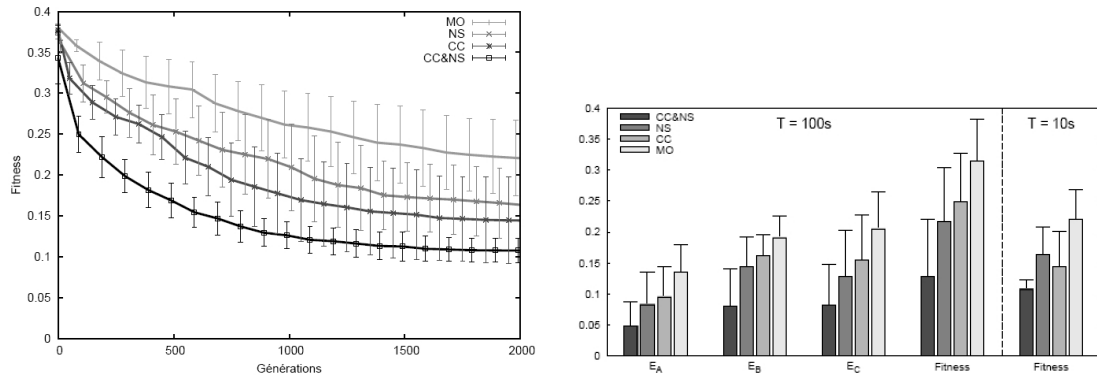


FIGURE 6.12 – Etude statistique de l’influence des différents modèles de neurones sur les performances des neurocontrôleurs à chaque génération (à gauche) et pour les quatre scénarios d’évaluation après évolution pour une durée de 100s.

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. Après application de la perturbation, l’amplitude des commandes est modifiée particulièrement sur le genou et la vitesse effective rejoint la vitesse de consigne. Il y a donc compensation de la perturbation par le neurocontrôleur. L’effet des mécanismes adaptatifs homéostatiques est visible sur les deux figures.

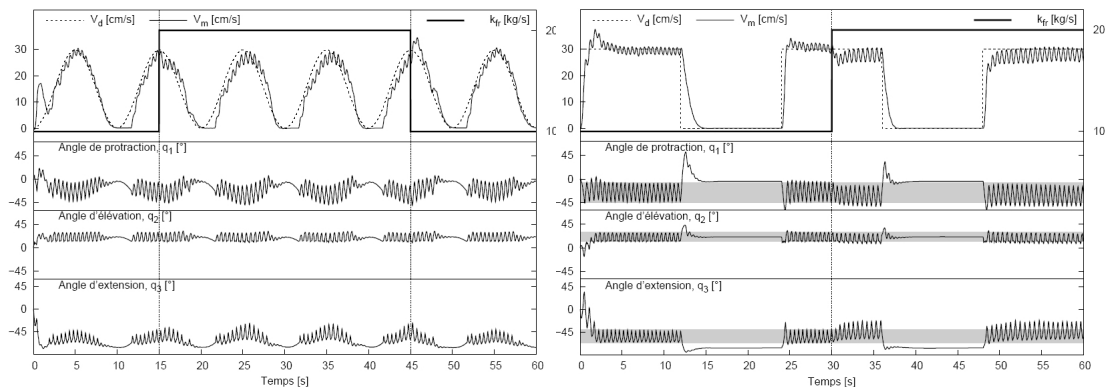


FIGURE 6.13 – Suivi de consignes de vitesse V_d avec ou sans perturbation (consignes sinusoïdales et en échelon) et variations temporelles des angles articulaires. Chaque oscillation de la vitesse mesurée V_m correspond à une foulée du robot. On note une phase transitoire pendant laquelle le contrôleur neuronal converge vers son cycle stable. La perturbation provoquée par l’augmentation des frottements visqueux induit une légère diminution de la vitesse.

Cette adaptation des commandes calculées par le réseau de neurones est aussi visible pour le cycle de locomotion (figure 6.15). En effet, du point de vue de la commande, on constate que pour compenser la perte de vitesse due au frottement, le contrôleur augmente l’amplitude de l’enjambée pour reculer le point d’appui de la patte sur le sol afin d’augmenter l’effort de poussée et maintenir la vitesse. D’autre part, les 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 reste maintenue lorsqu’il y a une perturbation.

Une analyse quantitative fine sur le fonctionnement des contrôleurs neuronaux nécessiterait de se doter d’outils d’investigation théoriques et puissants que nos plateformes de simulation ne possèdent pas encore. Une analyse qualitative effectuée sur les contrôleurs obtenus nous ont permis

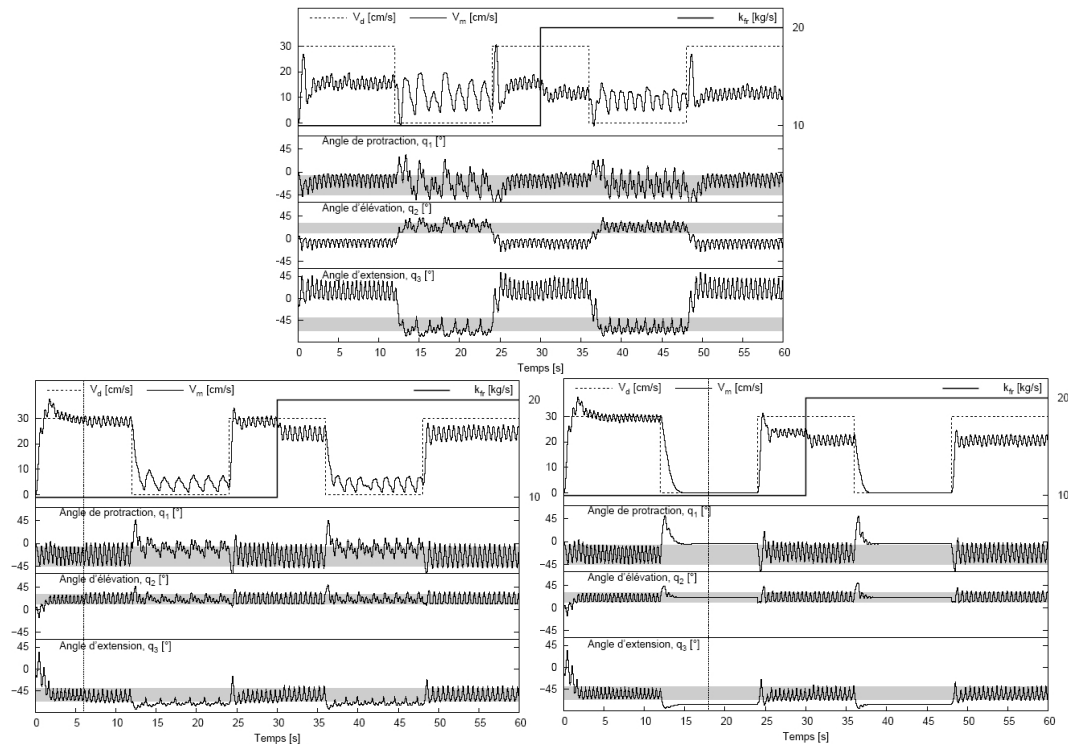


FIGURE 6.14 – Neutralisation de la plasticité du contrôleur aux instants $t=0s$ (à gauche), $t=6s$ (milieu) et $t=18s$ (à droite). Les bandes grises montrent le domaine de variation angulaire de chaque articulation en fonctionnement normal. On constate nettement que la rythmicité du contrôleur est fortement perturbée lorsque la plasticité disparaît : la consigne est non satisfaite (gauche et droite), ou le contrôleur est incapable d'exhiber des comportements d'arrêt (milieu).

cependant d'avancer quelques hypothèses intéressantes que l'on peut détailler ici. En effet, en observant un contrôleur performant (figure 6.16), on constate 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 mouvements rythmiques à partir des informations sensorielles. En effet, les synapses excitatrices de NS2 (erreur en vitesse) et NS1 (contact pied/sol) projettent sur NM3 et NM2 qui commandent respectivement l'articulation q_3 et q_2 responsables de la longueur d'enjambée, donc de la vitesse. La présence de connections inhibitrices allant de NM3 et NM1 vers NS2 confirme cette hypothèse puisqu'apparaît alors une structure en contre-réaction qui semble réguler le rythme en fonction de l'erreur en vitesse comme le montre la figure 6.16. 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 6.16, 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 dû servir à l'algorithme évolutionniste comme variable supplémentaire pour l'exploration de l'espace des solutions. L'algorithme évolutionniste, en sélectionnant à chaque génération une loi d'apprentissage pour chaque synapse, a donc abouti à une architecture de contrôleur qui respecte le principe d'un centre générateur de rythme modulé par une couche de neurones sensoriels et d'interneurones.

Ce travail a démontré l'intérêt d'incorporer dans les contrôleurs neuronaux des mécanismes de plasticité neuronale et synaptique. Les premiers, en exerçant des régulations locales au niveau du neurone permettent une robustesse du neurocontrôleur tandis que les seconds, en permettant une variation permanente des poids synaptiques apportent une capacité d'adaptation du contrôleur face aux perturbations externes. Toutefois, les modèles de neurone utilisés pour les rythmes locomoteurs

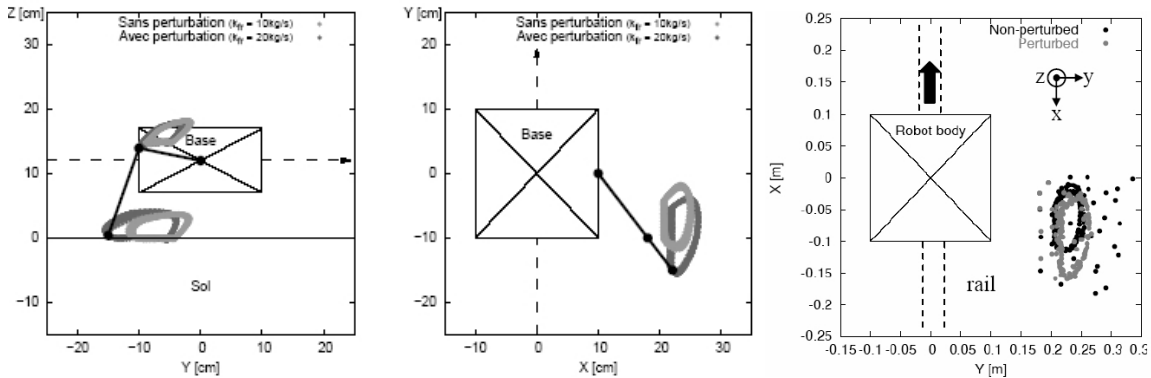


FIGURE 6.15 – Mise en évidence de l’adaptation du cycle de marche de la patte du monopode. En gris clair, sans perturbation, en gris foncé, avec perturbation (doublement des frottements sur le rail). On voit que les mécanismes homéostatiques provoquent un recul du point d’appui de la patte et un allongement de l’enjambée. La répartition des point d’appui sur le sol (à droite) montre les variations dans le cycle de la patte (extrait de Hoinville (2007)).

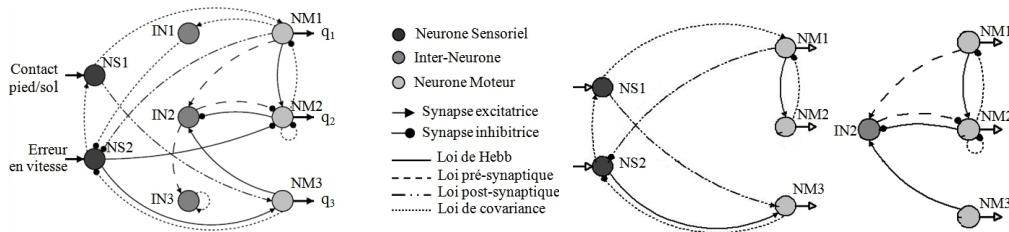


FIGURE 6.16 – Mise en évidence des sous-structures identifiées dans le neurocontrôleur obtenu (à gauche). Quand on observe attentivement la structure de ce contrôleur, on peut distinguer une sous-structure en boucle fermée régulatrice de la vitesse (au milieu) et une autre sous-structure coordinatrice du mouvement de la patte (à droite).

et les architectures de contrôleurs obtenues sont biologiquement assez peu plausibles en regard des travaux de Mac Crea sur les CPG (Mc Crea et Ryback, 2008). Il existe en effet, au cœur des CPG des neurones qui sont intrinsèquement rythmiques et qui peuvent exhiber des phases différentes (Marder et Bucher, 2001). La section suivante présente donc une extension des travaux précédents mais avec une approche biologiquement plus plausible, tant au niveau des modèles de neurone, des mécanismes plastiques incorporés que des modèles d’architecture.

6.2.3 Locomotion adaptative pour les robots myriapodes.

L’intérêt d’évaluer les modèles de neurocontrôleur sur une structure myriapode, est que celle-ci est dotée d’un grand nombre de pattes donc de degrés de liberté qui permettent d’exhiber une grande variabilité de situation d’adaptation en incorporant des dysfonctionnements mécaniques dans les pattes ou en modifiant à volonté la morphologie du robot (amputation de patte, de corps). Dans la continuité du projet IRON, nous avons donc suivi la démarche de simuler un robot dont on peut choisir à volonté le nombre de pattes et provoquer de tels dysfonctionnements (figure 6.17).

L’objectif de cette plateforme de simulation est de démontrer que l’on peut construire un contrôleur neuronal qui commande le robot myriapode vers une cible visuelle perçue par deux neurones sensoriels (droite et gauche) sur la tête (6.18), quels que soit les dysfonctionnements que peut subir le robot (blocage d’articulation, amputation de patte, amputation de corps). Pour ce faire l’architecture neuronale du myriapode respecte les principes fondamentaux observés en biologie :

- **modularité** : les contrôleurs de patte sont identiques entre eux et se répètent le long de la structure
- **sensitivité** : des neurones sensoriels existent pour toutes les valeurs angulaires des articu-

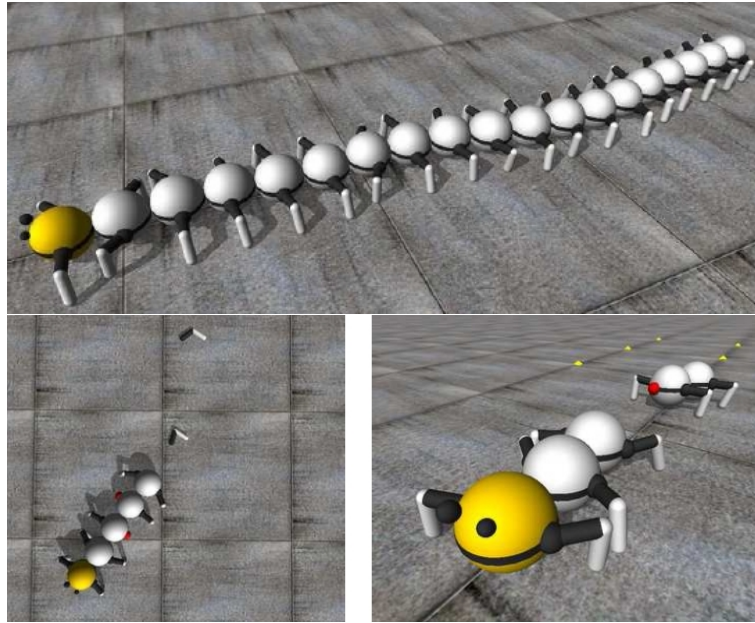


FIGURE 6.17 – Modèle de robot myriapode utilisé dont le nombre de modules est variable. Exemple d’une structure de robot à 19 modules (longueur 3m80). En bas, deux types de déficiences simulées : amputations de pattes et du corps. (extrait de (Hoinville, 2007)).

lations, pour les forces de poussée de chaque patte contre le sol et pour la détection de la direction, pour les forces de poussée de chaque patte contre le sol et pour la détection de la direction de la cible. Ces neurones sensoriels sont statiques et non linéaires contrairement aux capteurs usuels en robotique. Un bon modèle est celui proposé par Ekeberg, du type sigmoïde (ou tanh), donc quasi linéaire pour les valeurs moyennes (signaux de faible amplitude) et non linéaires pour les valeurs extrêmes.

- **symétrie** : les contrôleurs de pattes sont liés par des connections synaptiques selon un schéma biologiquement plausible respectant les contraintes de symétrie établies par Golubitsky et al. (Golubitsky *et al.*, 1999) : pour un animal à $2n$ pattes, l’architecture du contrôleur se compose de $4n$ sous-réseaux couplés, dont une moitié est utilisée pour contrôler les pattes, l’autre pour transmettre l’activité.
- **rythmicité** : un contrôleur de patte est composé de motoneurones, d’interneurones excitateurs et inhibiteurs, et de neurones rythmiques reproduisant les quatre propriétés intrinsèques qui caractérisent les neurones rythmogènes composants les CPG. Le modèle choisi est celui proposé par Rowat et Selverston présenté dans la section 4.2.1.
- **neuromodulation** : ce mécanisme de plasticité neuronale et synaptique est généré par un type de neurone supplémentaire dont le rôle est de moduler les paramètres intrinsèques (σ_s, σ_f) et les poids w des neurones rythmiques. Cette neuromodulation est caractérisée par une vitesse d’activation et de désactivation liées à deux constantes de temps. Chaque neurone rythmique peut être sensible à plusieurs neuromodulateurs selon une formule de barycentre. La sensibilité d’un neurone à un neuromodulateur est une interpolation linéaire entre le paramètre modulé et la variable modulante (pour plus de détails, lire (Hoinville, 2007)).

Méthode d’évaluation du modèle de contrôleur

Le contrôleur complet du myriapode est obtenu en synthétisant le contrôleur d’un hexapode soumis à différents scénarii pendant l’évolution artificielle. Ce contrôleur d’hexapode est ensuite « étendu » en respectant les contraintes de symétrie citées plus haut pour satisfaire à la cinématique du robot myriapode qui peut aller jusqu’à une centaine de pattes. Afin de faire évoluer les contrôleurs neuronaux de l’hexapode, celui-ci est posé sur un tapis roulant dont la vitesse varie suivant le scénario (figure 6.18). La fonction d’évaluation du contrôleur correspond à la durée pendant

laquelle le module de tête du robot reste maintenu dans un « domaine de viabilité¹ » représenté par les contours du carré rouge. Ainsi le neurocontrôleur doit être capable d'adapter son rythme locomoteur à la vitesse du tapis. Ce principe de test de la structure neuronale en simulation est très proche de celui réalisé par Ski et Orłowski ((Shik *et al.*, 1966)) en 1960 sur les chats décérébrés posés sur un tapis roulant.

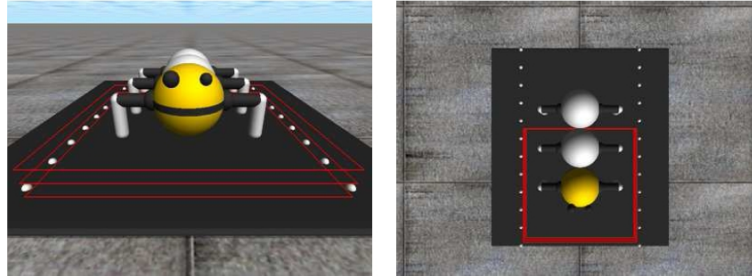


FIGURE 6.18 – Plateforme de d'évaluation de chaque neurocontrôleur. La vitesse du tapis est variable. Le module de tête du robot (en jaune) doit resté maintenu dans un « domaine de viabilité » représenté par le carré rouge (extrait de (Hoinville, 2007))

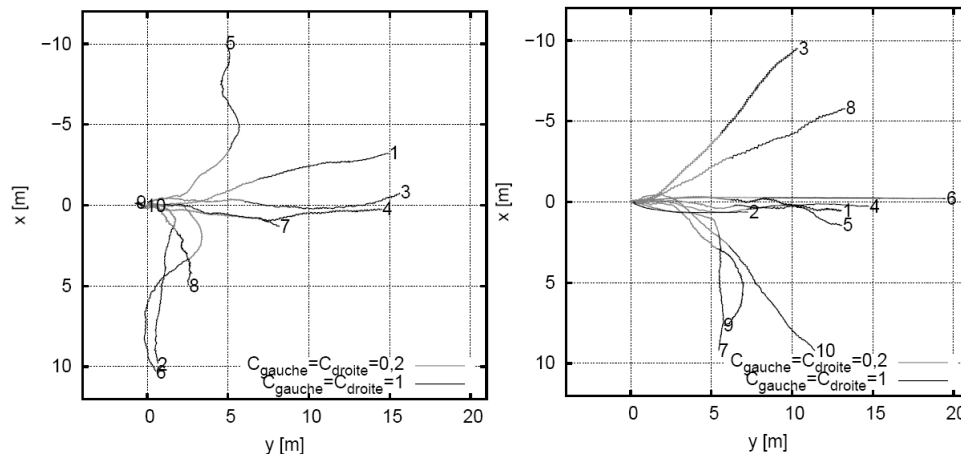


FIGURE 6.19 – Trajectoires obtenues après évolution : à gauche pour l'hexapode, à droite pour un décapode. La consigne reçue par les capteurs sensoriels de direction est d'effectuer une trajectoire rectiligne à faible vitesse ($C=20\%$) pendant les 30 premières secondes, puis à vitesse maximale ($C=100\%$) pendant les 30 suivantes. Une trajectoire rectiligne est donc un bon résultat. Le régime transitoire correspondant à la mise coordination des pattes provoque un désalignement de la trajectoire du robot par rapport à l'axe y visible sur la figure.

Les résultats (voir (Hoinville, 2007)) ont montré l'efficacité du modèle de neurone rythmique dans la genèse de rythmes locomoteurs. En effet, le fait que ce modèle soit intrinsèquement rythmique permet à l'algorithme génétique d'obtenir dès la première génération des neurocontrôleurs exhibant une activité rythmique apte à déplacer le robot. De plus, les architectures obtenues sont robustes à l'amputation de pattes, voire du corps, car les rythmes locomoteurs persistent et restent assez bien coordonnés après l'apparition de ce type de dommage jusqu'à une certaine limite. La figure

Ceci montre bien la robustesse du modèle de CPG basé sur le neurone rythmique et les boucles sensori motrices. Cependant, l'efficacité des mécanismes de neuromodulation sur les paramètres de plasticité du neurone pour rendre le neurocontrôleur adaptatif, doit encore être démontrée. Mais ce point est abordé au chapitre suivant, car nous avons par la suite exploité les propriétés de ce modèle pour la marche des robots bipèdes en adaptant les paramètres intrinsèque du neurone rythmique pour maintenir la posture du robot.

1. la théorie de la viabilité a été proposée par J. P. Aubin en 1991 (Aubin, 1991)

6.3 Conclusion

Ce travail sur l'évolution de contrôleurs neuronaux plastiques a montré l'intérêt de prendre en compte des modèles de neurones qui respectent les principes biologiques fondamentaux pour l'élaboration de contrôleurs neuronaux chargés de la locomotion des robots à pattes. Nous avons montré que le CPG qui contrôle une patte doit être basé sur des neurones intrinsèquement rythmiques, qu'il doit recevoir des retours sensoriels provenant des articulations, mais aussi du contact avec le sol et de l'environnement en général, qu'il doit inclure des mécanismes de régulation qui lui confère une certaine plasticité. Nous avons d'autre part établi que, si ces principes locaux sont respectés, on peut alors déployer ce modèle de CPG sur l'ensemble du robot myriapode, quel que soit le nombre de pattes, en respectant les principes de symétrie pour produire des mouvements de flexion/extension des articulations. Le modèle de contrôleur du robot myriapode est alors très robuste car il permet, lorsque le robot subit des amputations de patte ou de corps, de continuer à produire des mouvements rythmiques efficaces. Cependant, nous n'avons pas montré encore que nos modèles permettent de compenser pleinement ce type de dommage, c'est-à-dire que les rythmes obtenus après amputation permettent une marche stable et fluide du robot amputé.

Nous avons utilisé l'évolution artificielle comme un outil qui permet une étude exploratoire approfondie. Cependant, celle-ci ne permet pas de comprendre aisément pourquoi telle architecture de neurocontrôleur obtenue est plus performante que telle autre. En d'autres termes, l'approche évolutionniste dans notre travail nous permet d'explorer des solutions de contrôleurs originaux, de vérifier les performances des modèles mais elle ne nous permet pas d'expliquer le fonctionnement complet du contrôleur obtenu, ce qui est souvent pénalisant dans une vue d'automaticien.

Le chapitre suivant est consacré au transfert des modèles et des concepts précédents pour le contrôle de la marche bipède, mais en se basant sur une construction manuelle de l'architecture du neurocontrôleur et en exploitant au maximum les propriétés de plasticité du neurone rythmique.

Chapitre 7

Vers une architecture de contrôleur biologiquement inspiré pour la marche des robots bipèdes

Les travaux que j'ai présentés aux chapitres précédents nous ont permis de déduire les propriétés qu'une architecture bio-inspirée de contrôleur neuronal pour la locomotion doit respecter. Les travaux présentés au chapitre 4 sur les modèles computationnels de neurocontrôleurs proposés dans la littérature (Mac Crea, Ekeberg, Worgotter, Ijspeert...) mettent en avant les modèles de neurones et d'architectures. Les travaux que nous avons développés et qui sont présentés aux chapitres 5 et 6 ont mis en évidence l'influence de la dynamique neuronale sur la stabilité de la commande, l'apport des mécanismes homéostatiques neuronaux dans la robustesse de la dynamique neuronale, et l'intérêt d'utiliser dans les CPG des modèles de neurones intrinsèquement rythmiques qui incorporent ces mécanismes homéostatiques. Ces différentes propriétés peuvent être résumées ainsi :

- sur les modèles de neurones :
Différents types de neurones doivent exister dans le neurocontrôleur : les neurones rythmiques, les neurones sensoriels et les interneurons. Les neurones rythmiques et les interneurons doivent être dynamiques tandis que les neurones sensoriels peuvent ne pas l'être.
- sur la dynamique neuronale :
Le neurone rythmique doit pouvoir exhiber différents comportements dont la rythmicité endogène stable, l'hyper-amortissement, l'amortissement et la quiescence. Il doit par ailleurs inclure des mécanismes de plasticité qui stabilisent son comportement dans ces différentes phases de fonctionnement et qui le font transiter d'une phase à une autre.
- sur l'architecture du contrôleur pour une articulation
Le CPG doit inclure différents niveaux : celui où sont engendrés les rythmes, celui qui module les patrons rythmiques et le niveau moteur qui envoie les consignes aux actionneurs. Ces différents niveaux doivent recevoir en entrée des retours sensoriels proprioceptifs locaux (sur l'articulation : butées, vitesses, accélérations, forces) ou globaux (équilibre) ou extéroceptifs (centre de pression au sol, détection de l'horizon...). Ces retours sensoriels viennent moduler par différents mécanismes les modes de rythmicité du neurocontrôleur.
- sur l'architecture globale du contrôleur :
Celui-ci doit respecter des contraintes de symétrie liées à la cinématique du robot. L'actionnement antagoniste (Flexion/Extension...) présent dans les structures biologiques peut être fusionné par un neurone sommateur au niveau de la couche motrice pour piloter les moteurs électriques classiquement utilisés dans les robots humanoïdes. A noter que pour d'autres types d'actionneurs (verins, polymères électro-actif...) cette fusion peut ne pas être nécessaire. Enfin, le contrôleur doit recevoir des informations issues de niveaux supérieurs qui donnent une consigne globale pour le contrôle.

Dans ce chapitre, nous montrons comment prendre en compte ces propriétés afin de proposer un neurocontrôleur capable de produire une marche adaptative stable et robuste pour les robots bipèdes.

7.1 Modèle d'architecture sensori-motrice

Afin d'obtenir une coordination souple et dynamique dans le cycle de la marche, Pitti et al. (Pitti *et al.*, 2009) montrent que les CPG peuvent être vus comme des contrôleurs chaotiques qui maintiennent la synchronisation de phase dynamique entre eux et la dynamique du robot (figure 7.1). Une façon d'atteindre cet objectif consiste à utiliser des oscillateurs non linéaires. Ce type de modèle de CPG, comme ceux de Van der Pol (Dutra *et al.*, 2003) et Rayleigh (De Pine Filho *et al.*, 2005) a déjà été utilisé pour contrôler des robots bipèdes (en simulation). Néanmoins, à notre connaissance, le robot est souvent commandé en boucle ouverte par rapport à son environnement. Il n'y a pas de neurones sensoriels pour améliorer l'activité rythmique, particulièrement pas de boucle sensorielle liée au contact du pied avec sol, dont on sait en biologie qu'elle est essentielle pour la stabilité de la marche.

Dans le cadre de la thèse de T. Hoinville, nous avons utilisé le modèle de neurone rythmique de Rowat et Selverston (voir section 4.2.1) pour résoudre une tâche de locomotion (Hoinville, 2007). L'intérêt de ce modèle est qu'il permet aisément d'exhiber les modes fondamentaux des neurones rythmiques biologiques et qu'il permet d'obtenir des centres générateurs de rythmes robustes comme l'ont montré les résultats obtenus par l'évolution artificielle. Avec ce modèle, on peut donc construire une architecture de neurocontrôleur pour la marche rythmique des bipèdes en boucle fermée. De plus, on peut aisément y incorporer des mécanismes adaptatifs par la modulation de ses paramètres intrinsèques afin d'exhiber plusieurs types de patrons rythmiques de fréquence variable.

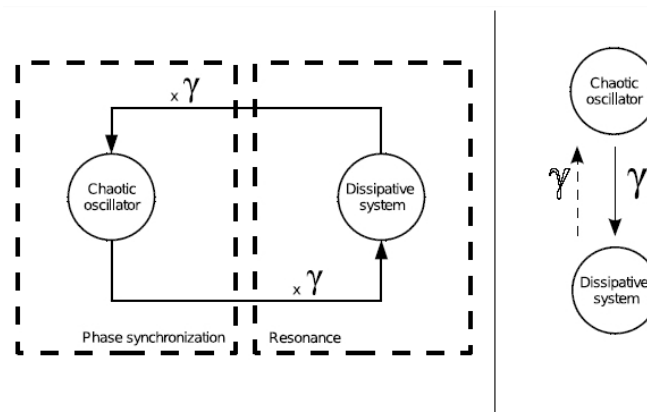


FIGURE 7.1 – Mécanisme de synchronisation de phase et boucle de résonance entre un système actif et un système dissipatif. Selon les valeurs du paramètre de couplage γ , l'oscillateur chaotique synchronise ou non la phase du système dissipatif (le robot). Figure extraite de (Pitti *et al.*, 2009).

7.1.1 Importance des retours sensoriels sur l'efficacité de la marche

Les travaux qui ont été présentés précédemment ont démontré le rôle important des boucles sensorimotrices entre le sol et le CPG pour la production d'un rythme locomoteur robuste et efficace. L'étude d'une première architecture de CPG pour la marche de robot bipède demande à mieux comprendre le modèle de neurone rythmique de Rowat et Selverston (Rowat et Selverston, 1993) en explorant ses différentes phases en fonction de ses paramètres et en étudiant l'importance du retour sensoriel pieds/sol sur l'efficacité du cycle locomoteur. Ce travail que j'ai effectué avec Elmira Amrollah au printemps 2008, dans le cadre de son projet de Master, a été publié dans le journal *Neurorobotics* de la série *Frontiers in Neurosciences* en décembre 2010 (voir l'article (Amrollah et Hénaff, 2010) en annexe). Nous avons repris le problème d'une patte glissant sur un rail (comme dans la section 6.2.2) que le neurocontrôleur doit coordonner pour obtenir une vitesse de déplacement. Comme le montre la figure (7.2), l'architecture proposée est inspirée de celle décrite par Mac-Crea et Ryback (voir section 4.2.2). Elle est composée d'une couche rythmique, d'une couche formatrice et d'une couche motrice. Des neurones sensoriels informant de la position angulaire de la hanche et de la force de poussée contre le sol viennent projeter sur les neurones

rythmiques flexeur et extenseur de Rowat et Selverston. Des interneurons informant de la position et de la vitesse du genou (modèle de Ekeberg) viennent moduler les signaux sortants des neurones rythmiques. L'étude statistique que nous avons menée figure (7.2) montre bien que le retour sensoriel provenant du sol améliore nettement la vitesse du robot par le couplage dynamique qui est créé lors de l'interaction avec le sol. Ce travail préliminaire sur une meilleure compréhension dans l'utilisation du modèle de Rowat et Selverston permet maintenant de proposer une architecture plus complète pour la marche bipède.

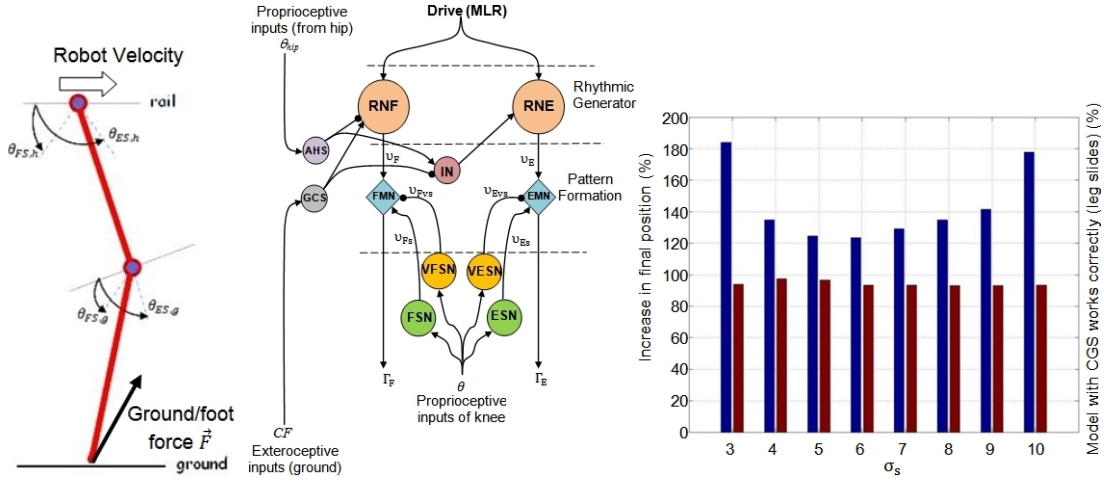


FIGURE 7.2 – Modèle de CPG, pour l'articulation du genou (le CPG au niveau de la hanche n'est pas représenté pour plus de clareté). Les entrées proprioceptives proviennent de la hanche (θ_{hip} angle d'extension) et les entrées extéroceptives du contact contre le sol (CF : intensité de la composante verticale de la force de contact \vec{F}). A droite, statistique comparative (247 configurations initiales différentes de la patte) pour huit valeurs du paramètre σ_s du neurone (paramètre pilotant la fréquence d'oscillation, voir équation 4.3). En rouge pourcentage d'essais aboutissant à un déplacement du robot. En bleu : valeur moyenne de l'augmentation de la distance parcourue avec le retour sensoriel du sol GCS par rapport à sans GCS .

7.1.2 Architecture neuronale complète pour la marche bipède

L'oscillateur rythmique de Rowat et Selverston peut exhiber différents modes suivant les valeurs de ses paramètres intrinsèques. Sa plage de fréquence dépend de ses deux constantes de temps et ses phases de ses deux paramètres de conductance σ_s et σ_f .

Modèle de CPG articulaire

Si l'on construit autour de ce neurone un CPG pour une articulation, comme celui de la figure 7.3a, l'étude systématique des modes rythmiques du CPG permet une classification de ses comportements suivant les valeurs des paramètres (σ_s, σ_f) des neurones. La figure 7.3b montre l'effet de ces paramètres sur un indice énergétique du mouvement appliqué sur l'articulation (image de l'énergie cinétique) :

$$\varepsilon = \int_{t_0}^t \dot{\theta}^2 dt \quad (7.1)$$

Le contrôle de ce couple de paramètres permet donc de contrôler les transitions de phases du CPG. On peut par conséquent imaginer de piloter des sauts de phase d'un contrôleur neuronal constitué de deux CPG pour une patte de robot, et donc de contrôler les allures de marche si chaque CPG articulaire est construit à partir de ce modèle de neurone (figure 7.3c). Contrôler l'allure de la marche du bipède revient donc à contrôler la position du point de coordonnées (σ_s, σ_f) sur cette surface.

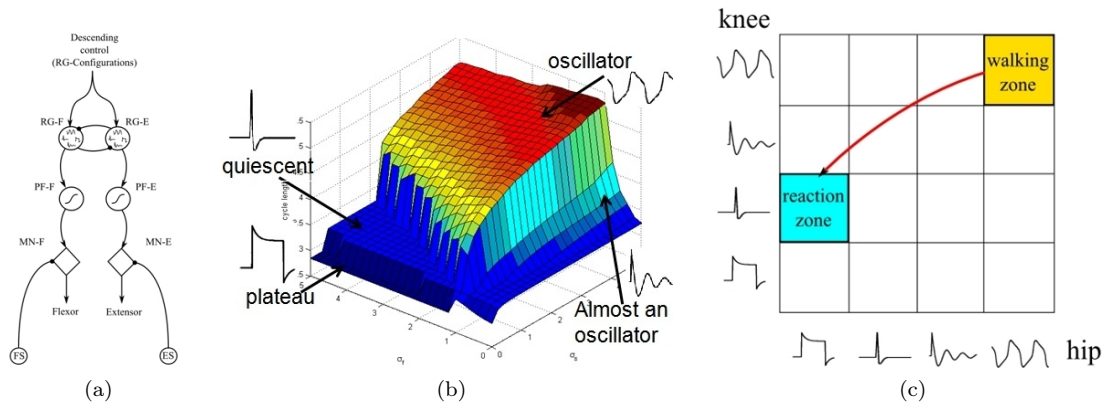


FIGURE 7.3 – (a) Structure du CPG à base de neurones rythmiques de Rowat et Selverston. (b) propriété du CPG en fonction de ses paramètres de conductance σ_s et σ_f . (c) sauts de phase entre deux CPG contrôlant une patte.

Modèle d'architecture du neurocontrôleur

C'est sur cette idée que nous avons proposé avec John Nassour une architecture de neurocontrôleur capable d'exhiber des changements d'allure notamment pour réagir face à une perturbation qui déstabiliserait la marche d'un robot bipède (pour plus de détails lire l'article Nassour *et al.* (2010a) en annexe). Cette architecture est construite autour du CPG élémentaire de la figure 7.3a. C'est une version améliorée du CPG de la figure 7.2 (essentiellement le couplage entre les neurones rythmiques) et comprend les couches rythmique, formatrice, et motrice et des neurones sensoriels. L'architecture du neurocontrôleur global est présentée figure 7.4.

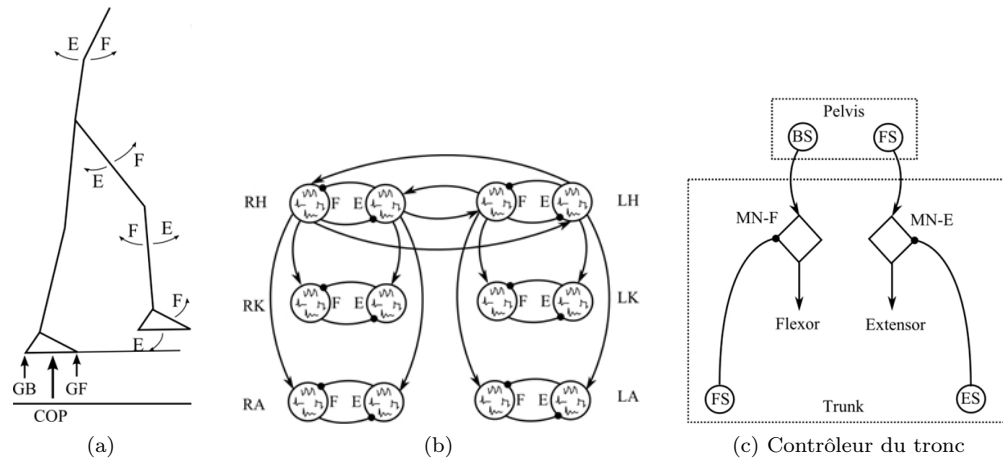


FIGURE 7.4 – Architecture du neurocontrôleur complet du bipède : (a) Modèle de bipède, (b) Couplages articulaires entre les neurones rythmiques, (c) Contrôleur du tronc

Toutes les articulations sont couplées aux deux hanches par des synapses excitatrices projetant sur les couples de neurones rythmiques de chaque CPG articulaire. C'est donc par les hanches que la coordination de la marche s'effectue par une projection « descendante ». Un mécanisme de contrôle du tronc permet de maintenir l'articulation du pelvis à la verticale par deux neurones sensoriels l'un pour détecter l'inclinaison vers l'avant (Front Sensor), l'autre pour détecter l'inclinaison vers l'arrière (Backward Sensor). Ensuite, pour chaque articulation, les retours sensoriels sont différents. Ainsi pour le genou (figure 7.5b), un neurone sensoriel d'extension de la hanche vient moduler les patrons rythmiques. Pour la cheville (figure 7.5c), deux types de neurones sensoriels projettent sur l'étage formateur : un neurone mesurant la force de poussée contre le sol à l'avant du pied (Ground Forward) et un autre à l'arrière (Ground Backward) et des neurones sensibles au risque de chute

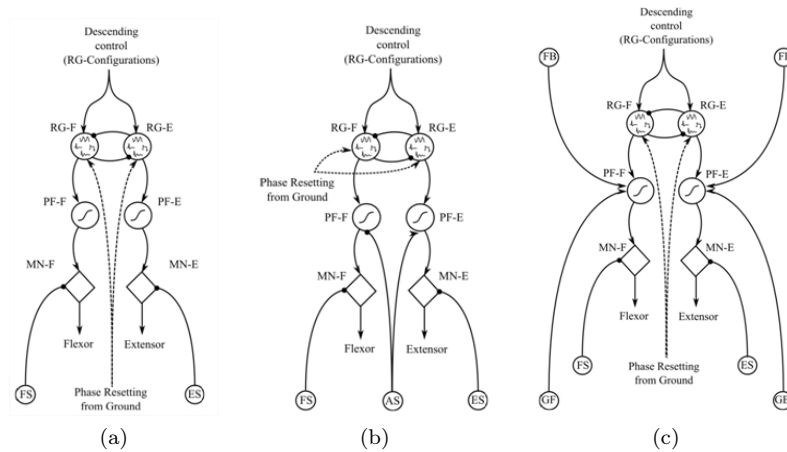


FIGURE 7.5 – Détail des CPG pour chaque articulation

vers l'avant (Fall Forward) ou vers l'arrière (Fall Backward). Ce risque étant proportionnel à la distance entre la projection du centre de masse du robot sur le sol et la position du centre de pression mesuré sur le pied. Enfin, un signal provenant du contact avec le sol vient réinitialiser la phase de chaque CPG articulaire au niveau des neurones rythmiques.

Modèle d'apprentissage du contrôle d'équilibre

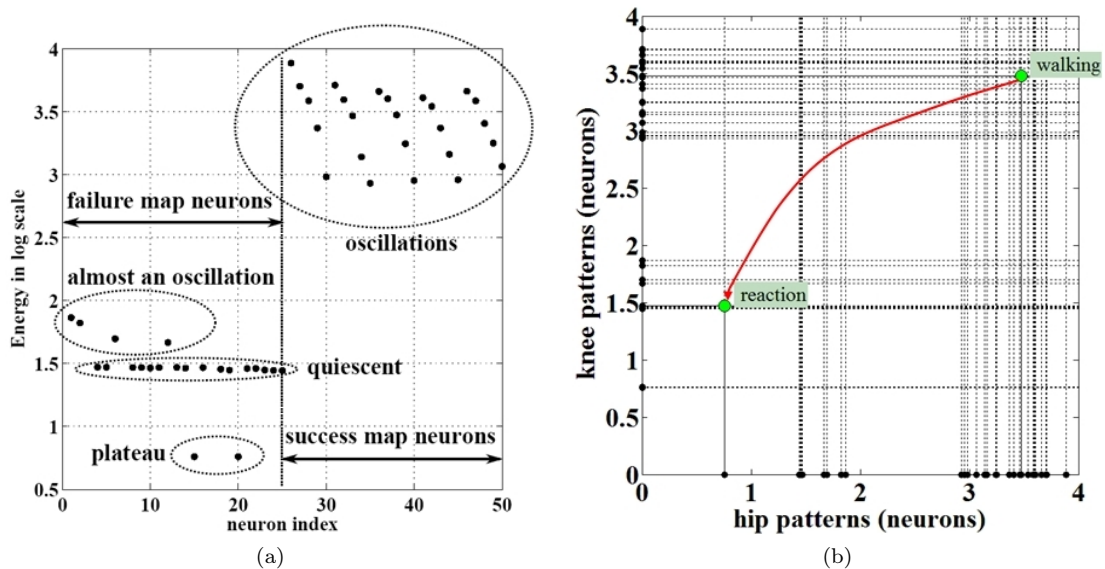


FIGURE 7.6 – Représentation matricielle 2D des modes rythmiques : (a) au niveau du CPG où les neurones de la carte de Kohonen sont classés suivant l'indice énergétique ε du CPG, (b) au niveau de la patte où le mouvement de la patte dépend de l'indice énergétique du neurone de la carte.

Nous avons introduit un mécanisme d'apprentissage basé sur la notion de seuil de vigilance identique à celui présenté à la section 5.3, l'apprentissage est mené de la même façon sur un sol de pente irrégulière. Cependant, les cartes de succès et d'échec de Kohonen représentent ici les couples (σ_s, σ_f) qui caractérisent le mode du neurone comme le montre la figure 7.6a (pour plus de détails, lire l'article (Nassour *et al.*, 2010a) en annexe). Chaque neurone d'une même carte topologique caractérise donc une phase du CPG, c'est-à-dire son patron rythmique dont l'indice est la valeur logarithmique de la valeur énergétique ε introduit plus haut pour la classification des phases. On

peut donc représenter l'allure de la marche par un tableau 2D où l'abscisse est l'indice du neurone classifieur de la hanche et l'ordonnée celui du genou (figure 7.6b). Cette représentation permet alors de contrôler les sauts de phase par le choix d'un neurone dans chaque carte topologique articulaire conformément à l'idée initiale représentée par la figure 7.3c.

Exemple de résultat :

Le modèle de robot bipède simulé est plan, à trois degrés de liberté par patte plus un degré sur le tronc, il pèse 22 kg (des études sont en cours pour passer à un modèle 3D implémentable sur le robot Nao). Comme le montre la figure 7.7a, l'architecture du neurocontrôleur permet d'obtenir une marche stable du bipède à 0,2 m/s tant que celui-ci n'est pas perturbé même sans mécanisme adaptatif. Sans mécanisme encore, si l'on applique une force de poussée de 45 Nm, on voit sur la figure 7.7b que le bipède perd son équilibre et tombe. Enfin, la figure 7.7c montre que le mécanisme d'adaptation des paramètres des neurones rythmiques produit le réflexe d'effectuer un grand pas vers l'avant pour éviter la chute.

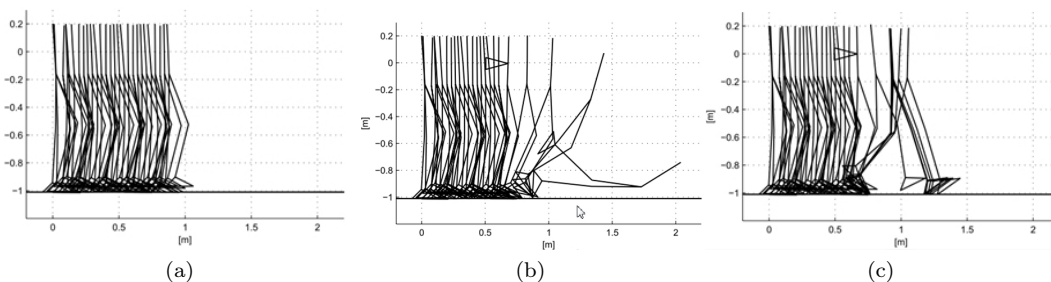


FIGURE 7.7 – Réflexe de changement d'allure suite à une perturbation. (a) sans mécanisme d'adaptation et sans perturbation la marche est stable, (b) sans mécanisme d'adaptation et avec perturbation le robot tombe, (c) avec mécanisme d'adaptation et avec perturbation le robot fait un grand pas vers l'avant pour rattraper son équilibre.

7.2 Conclusion

Nous avons proposé dans ce chapitre un modèle de contrôleur bio-inspiré pour la marche bipède. Nous avons montré qu'en contrôlant le comportement des neurones générateurs de rythme on peut produire un comportement adaptatif face aux perturbations externes. La transition entre un mode de marche normale et un mode réflexe s'avère simple à partir du concept d'apprentissage de l'échec et du succès. La démarche qui consiste à représenter chaque mode rythmique par son indice énergétique conduit à une méthode simple de classification des modes et réduit considérablement la dimension de l'espace de contrôle. Cet espace permet une commande du haut niveau de l'allure de marche et un apprentissage des sauts de phases qui engendrent des changements dans l'allure de marche, jusqu'à produire des réflexes d'équilibre. Le mécanisme d'apprentissage basé sur le seuil de vigilance, introduit sur le modèle de contrôleur proposé par Worgotter *et al.* produit avec le modèle proposé ici toute son efficacité. Cependant, afin de valider complètement cette approche, il conviendra de l'appliquer à la marche 3D et de la valider sur un robot réel.

Chapitre 8

Conclusion et perspectives de recherche

Mes travaux présentés dans ce document résument près de 10 années de recherche dans le domaine de la commande des robot marcheurs par réseaux de neurones. Tout au long de ces années, ma démarche a consisté sans cesse à garder un équilibre entre l'approche automatique et l'approche bio-inspirée sans les mettre dos à dos. Les bons résultats que j'ai obtenus proviennent de cet équilibre. En effet, non seulement l'automatique apporte des méthodes aux techniques bio-inspirées (dans la synthèse évolutionniste par exemple, en proposant des scénarios à base de signaux tests percussionnels, et en vérifiant le réalisme des solutions et des modèles obtenus), elle apporte aussi des méthodes dans l'observation et l'analyse des résultats en terme de précision et de stabilité. Cependant, l'automatique n'apporte pas encore d'outils mathématiques suffisants pour étudier théoriquement la stabilité des neurocontrôleurs proposés et de leurs algorithmes d'apprentissage.

L'objectif scientifique visé, c'est-à-dire l'obtention de neurocontrôleurs adaptatifs capables de contrôler un robot marcheur lorsque celui-ci subit des perturbations ou des dommages est un sujet vaste et complexe qui nécessite encore beaucoup d'investigations. Il reste encore à trouver des modèles de neurocontrôleurs robustes capables de compenser des dommages importants comme le sous-actionnement soudain ou l'amputation de membres moteurs. Cependant, les travaux que j'ai menés ont permis d'obtenir des résultats intéressants tant sur les modèles de contrôleurs que sur la méthodologie de leur construction.

Nous avons implémenté des algorithmes d'apprentissage complexes, basés sur l'optimisation d'une fonction de coût exprimant la tâche, sur des baies de contrôle temps-réel avec des temps de calculs compatibles avec la commande. C'est une approche inspirée par les méthodes de l'automatique et qui ne prend pas en compte les aspects cognitifs de l'apprentissage des êtres vivants. Nous avons ensuite montré que la dynamique interne des neurones et les mécanismes de plasticité associés sont essentiels pour la robustesse des contrôleurs et l'équilibre durant la marche. Nous avons testé et validé nos modèles en simulation sur différents types de robots marcheurs (monopode, bipède, et myriapode) soumis à des perturbations. Les modèles de contrôleurs que nous avons étudiés sur ces différentes structures cinématiques ont permis d'exhiber des comportements adaptatifs et de démontrer que les retours sensoriels entre le sol et les pattes permettent d'améliorer le couplage dynamique entre l'oscillateur qui commande le robot et le robot qui dissipe de l'énergie en interagissant avec le sol ou son environnement de façon générale. Nous avons démontré l'efficacité des mécanismes de plasticité intrinsèque sur la robustesse du contrôleur. Plus généralement, nous avons établi que les modèles de contrôleurs neuronaux, quand ils respectent les règles fondamentales du contrôle biologique, permettent d'implémenter facilement les concepts de commande et de régulation qui existent dans la nature et qui y ont fait leur preuve en terme de robustesse, d'adaptation et de stabilité.

Le roboticien trouvera dans les travaux que j'ai présentés, les similitudes entre les principes fondamentaux des boucles biologiques de contrôle et ceux des boucles technologiques de contrôle utilisées depuis longtemps dans la commande et la robotique. Ainsi, les structures nerveuses chargées de la commande motrice, sont constituées de boucles fermées locales (niveau musculaire, i.e niveau actionneur) et globales comme le sont les boucles fermées d'un actionneur électrique,

pneumatique ou hydraulique. De plus, l'information sensorielle est essentielle à la stabilité de ces boucles de contrôle biologique comme le capteur l'est pour une boucle d'asservissement. Enfin, les mécanismes de plasticité neuronale et synaptique permettent une adaptation des caractéristiques des contrôleurs neuronaux afin de les rendre plus robustes comme l'est le contrôle adaptatif. Nous savons que les structures neuronales chargées de la locomotion sont des systèmes non-linéaires qui constituent des attracteurs dynamiques permettant un couplage optimal entre le système commandé et le contrôleur. C'est la non-linéarité intrinsèque de ces structures biologiques de contrôle qui les rendent aussi puissantes. C'est une grande différence avec l'automatique par laquelle on cherche souvent à utiliser des modèles linéaires. Non seulement, ces attracteurs donnent une grande robustesse aux contrôleurs neuronaux, mais leur plasticité intrinsèque leur permet de transiter d'une phase à une autre rapidement, procurant ainsi au système nerveux moteur des capacités d'adaptation importantes allant jusqu'à une reconfiguration de l'architecture lorsque l'animal voit sa morphologie modifiée (propriété d'autotomie).

Toutefois, pour être crédibles, les modèles proposés dans ce document doivent être validés sur des robots humanoïdes réels. C'est une partie importante du travail qui reste à faire dans le cadre de cette recherche sur le contrôle bio-inspiré de la marche bipède. Plusieurs plateformes robotiques disponibles en France pourront servir de support expérimental. Citons les plateformes commerciales existantes (Nao et bientôt Romeo qui sera disponible dans le cadre du programme national ROBOTEX), ou les plateformes expérimentales de laboratoire (au LISV : ROBIAN et HYDROID, robot hydraulique conçu dans le cadre du projet ANR PHEMA ; ou le robot Européen iCub), sans oublier les robots du laboratoire mixte Franco Japonais (HRP2, HOAP).

De plus, les travaux présentés ici permettent d'envisager d'autres retombées dans la commande des systèmes robotiques. En effet, les recherches menées depuis quelques années dans la conception de robots bio-inspirés non marcheurs (robots anguilles, robots poissons, robots oiseaux...) s'accroissent à travers le monde. Le succès du Workshop International « International Workshop on Bio-Inspired Robots » organisé par l'IRCCYN à Nantes en avril 2011 en est une belle démonstration. L'objectif de ces projets est d'une part une meilleure compréhension du monde vivant en construisant des systèmes artificiels et d'autre part la création de systèmes artificiels capables d'explorer des zones hostiles ou inaccessibles. Or, si l'on sait assez bien construire des robots à l'image des animaux, il est souvent difficile de les doter de contrôleurs aussi performants que ceux du monde vivant. Ces systèmes artificiels seront donc vraiment performants lorsque nous saurons concevoir des contrôleurs bio-inspirés dignes de leurs capacités motrices (mécanique) et sensorielles (électronique). Les biologistes savent bien que les structures nerveuses qui commandent les animaux évolués que nous sommes ne sont pas très éloignées dans leurs fondements de celles qui commandent des vertébrés qui ont peu évolué durant des millions d'années comme la lamproie. A mon avis, il n'y a donc pas d'obstacle de fond pour continuer à suivre le chemin tracé afin de chercher dans la nature de nouvelles solutions pour la locomotion et la commande des robots humanoïdes du futur.

Toutefois, chercher à résoudre le difficile problème de la locomotion humaine et animale ne sera pas suffisant. En effet, les robots humanoïdes représentent sûrement les premiers systèmes artificiels à l'image de l'humain qui devront interagir avec lui physiquement et socialement et cela comme un être humain le fait naturellement depuis sa naissance avec ses congénères. Nous ne sommes qu'aux balbutiements de ce domaine de recherche sur l'interaction physique et sociale des robots humanoïdes (Alami *et al.*, 2006). En effet, les mécanismes biologiques d'apprentissage mis en jeu à différents niveaux et différentes échelles de temps sont variés et complexes. Ils sont directement liés au fonctionnement et à la nature même des structures nerveuses, et au développement cognitif de l'humain. Implémenter des mécanismes d'apprentissage dans les réseaux de neurones artificiels dépend aussi du modèle de neurone utilisé, de la tâche à apprendre et du nombre de degrés de liberté commandé, de la nature de l'actionneur... L'efficacité d'un mécanisme d'apprentissage peut se mesurer par sa capacité à permettre une généralisation de la notion apprise. Or, les algorithmes d'apprentissage et les modèles actuels de contrôleurs neuronaux ne permettent pas encore d'atteindre des capacités de généralisation telles qu'on les observe dans la nature et donc de développer l'autonomie et l'interactivité des robots. Un enfant de quelques mois est capable de généraliser très vite une tâche qu'il vient d'apprendre ou un mouvement qu'il vient de découvrir. Ainsi, grâce à ses capacités d'apprentissage, il acquiert de l'autonomie au cours de son développement en interagissant avec son environnement.

Cette approche développementale de l'apprentissage (suivie notamment au laboratoire ETIS)

permet de mieux comprendre les mécanismes cognitifs qui permettent à l'humain d'acquérir la maîtrise du geste, de la posture, de la locomotion et du langage. Ainsi les travaux de Shadmehr (Shadmehr, 2009) sur les mouvements biologiques volontaires, permettent de faire un rapprochement entre les modèles cognitifs computationnels sensori-moteurs et la commande optimale. Dans cette approche mon positionnement scientifique, se situe jusqu'à présent dans la compréhension des mécanismes sensori-moteur bas niveau, c'est-à-dire en lien avec le cortex moteur. Cependant, les travaux que j'ai mené sur l'apprentissage des allures de marches des bipèdes en incluant un mécanisme décisionnel lié à la prise de risque prennent en compte les mécanismes cognitifs haut niveau qui se situent dans la zone du cortex préfrontal.

Dans le cadre de la compréhension du contrôle moteur par l'étude d'un modèle computationnel complet utilisable en robotique humanoïde, mon objectif scientifique reste la compréhension des boucles sensori-motrices bas niveau et leur liens avec les niveaux supérieurs. La compréhension de ces mécanismes ne peut se faire sans prendre en compte les boucles d'interactions physiques présentées en introduction de ce document. Ces interactions permettent à l'humain d'apprendre, au cours du temps, à contrôler ses gestes, sa posture et ses émotions. La rythmicité joue un rôle essentiel dans cette interaction, et c'est vers cette nouvelle direction que je propose d'aller en m'appuyant sur le savoir acquis sur les modèles de neurocontrôleurs rythmiques pour la locomotion. Je décline cette direction de travail en trois points principaux : apprentissage et autonomie, apprentissage moteur et compliance articulaire, apprentissage et rythmicité.

8.1 Apprentissage et autonomie

L'adaptation aux dysfonctionnements ou plus généralement aux modifications morphologiques soudaines ou lentes, définitives ou réversibles, est une grande propriété des systèmes vivants qui leur confère une autonomie très importante. Je suis convaincu qu'on ne pourra pas obtenir de modèles de neurocontrôleurs robustes sans les confronter à de telles situations car cette propriété des systèmes naturels qui émerge des architectures nerveuses sensorimotrices correspond à un changement de phase lors d'une situation normale. Cela doit rester un objectif principal dans la recherche de lois de commande en robotique humanoïde. En effet, il sera dommage, à mon avis, de concevoir des machines aussi sophistiquées que les robots humanoïdes s'ils devaient rester incapables de s'adapter à des incidents ou accidents matériels qui n'attendent pas à leur « survie » et qu'ils ne puissent pas s'adapter à ce nouvel état pour continuer leur tâche en mode dégradé ou bien pour rentrer à leur base de maintenance pour y subir des réparations.

Les modèles de contrôleurs adaptatifs bio-inspirés restent donc un champ de recherche à explorer. Comme nous l'avons vu dans ce document, je propose de continuer à explorer l'une des pistes intéressante c'est-à-dire l'étude de mécanismes plastiques neuronaux et synaptiques, particulièrement la neuromodulation provenant des niveaux supérieurs dont le principe est assez simple, les modèles connus et l'efficacité démontrée en biologie.

8.2 Apprentissage moteur et compliance articulaire

Nous savons aujourd'hui, que les algorithmes de contrôle, inspirés par la biologie ou par la dynamique, auront beau être perfectionnés, ils se heurteront à la technologie des organes d'actionnement des robots humanoïdes tant que celle-ci n'intégrera pas la propriété de compliance. La compliance est une propriété « mécanique » des organes biologiques qui leur confèrent une souplesse ou une raideur variable selon les caractéristiques des mouvements et de la posture envisagés, de l'état émotionnel ou sensoriel. Comme nous l'avons rappelé au chapitre 5.2, beaucoup de travaux vont dans ce sens actuellement. J'ai montré qu'il est possible de commander cette compliance en apprenant à adapter la raideur d'asservissement articulaire, et que cette propriété permet d'améliorer les performances de la marche en terme de robustesse, de fluidité et de dissipation énergétique.

La notion de compliance a toutefois un sens plus large. Elle peut en effet être liée à la compliance d'un couplage mécanique d'une articulation qui voit sa raideur modifiée, ou à la compliance d'une interaction sociale entre un apprenant et un maître qui voient leurs rôles s'échanger. Les capacités d'apprentissage des robots du futur seront donc liées à leur propriété de compliance « mécanique » et « sociale ». Pour le contrôle moteur, il sera important de doter les centres générateurs de rythmes,

et les neurocontrôleurs en général, de mécanismes capables de contrôler cette compliance mécanique afin d'établir des modes de contrôles stables et cohérents qui permettront de favoriser l'interaction physique, voire sociale, des robots avec l'humain. J'ai entamé ce travail dans le cadre d'une thèse en cotutelle à ETIS sur le contrôle de la compliance articulaire lors de l'interaction physique entre un bras anthropomorphe et l'humain (Melnyk *et al.*, 2011).

8.3 Apprentissage et rythmicité

Même si nous sommes loin de pouvoir donner de telles capacités d'apprentissage aux robots humanoïdes, des pistes de recherche sont envisageables. En effet, il est admis en psychologie que c'est par l'interaction physique et sociale que l'apprentissage de l'enfant se fait. Les processus d'imitation dans le jeu, ou au cours d'une leçon avec un maître, sont essentiels pour l'apprentissage de nouveaux gestes. L'émotion joue aussi un grand rôle dans le renforcement des connaissances apprises.

On sait que les boucles sensorimotrices du système musculo-squelettique, c'est-à-dire le « bas niveau de la commande » sont aussi couplées aux aires corticales qui hébergent les centres émotionnels, c'est-à-dire le « haut niveau de la commande ». Des mécanismes de couplage dynamique existent donc aussi entre les centres générateurs de mouvements volontaires et les centres émotionnels. Un des rôles de l'apprentissage est de coupler la dynamique entre ces deux niveaux. Lorsque ce couplage devient optimal, l'apprentissage est effectif et l'apprenant peut devenir le maître dans l'interaction. Cette aptitude à un couplage dynamique pour établir l'interaction serait une propriété intrinsèque des structures neuronales sensorimotrices qui permettent par ailleurs la genèse des mouvements rythmiques.

La rythmicité globale de la personne définirait son identité individuelle (Troje *et al.*, 2006). En effet, il est admis en psychologie que la rythmicité est une propriété émergente de la dynamique d'interaction sociale et physique entre l'homme et son environnement. D'autre part, il semble aussi admis, que les mécanismes sous jacents aux modes de communication par le geste sont de même nature que ceux existant par les échanges verbaux. La rythmicité intrinsèque de ces mécanismes jouerait un rôle fondamental dans le couplage dynamique entre les individus, favorisant ainsi leur apprentissage. Cela doit donc aussi le devenir pour les robots humanoïdes du futur pour lesquels l'une des questions fondamentales à résoudre concerne les modèles qui seront à la base des mécanismes cognitifs qu'ils devront incorporer pour interagir physiquement et socialement avec l'humain, voire même entre eux. Pour atteindre cet objectif, il est donc important de mieux comprendre les soubassements communs aux capacités de communication et de locomotion de l'humain. Dans cette démarche suivie notamment par l'équipe Neurocybernétique de ETIS, je propose de chercher à modéliser certains de ces mécanismes afin de mieux comprendre ces fondements cognitifs et de proposer des algorithmes d'apprentissage et de commande qui permettront aux robots humanoïdes du futur d'interagir physiquement avec l'humain. Apprendre et maintenir une rythmicité globale et cohérente dans les mouvements des membres supérieurs et inférieurs est en effet essentiel pour le développement de la communication par le geste et pour l'apprentissage de la marche chez l'humain, il doit donc le devenir aussi pour le robot humanoïde.

Finalement, en couplant mécanismes d'apprentissage, rythmicité et compliance articulaire, il sera alors envisageable d'élaborer des signaux qui permettront au robot de percevoir et de juger de la qualité de son comportement et de se corriger par des mécanismes de renforcement. Nous montrerons ainsi que les interactions physiques haut niveau, les propriétés mécaniques et les boucles de contrôle bas niveau sont des éléments couplés, essentiels pour servir de base aux mécanismes de communication implicite inhérente à toute construction et acceptation sociale.

Bibliographie

- H. AHN et R. PICARD : Affective-cognitive learning and decision making : A motivational reward framework for affective agent. *In The 1st International Conference on Affective Computing and Intelligent Interaction*, 2005.
- R. ALAMI, A. ALBU-SCHAEFFER, A. BICCHI, R. BISCHOFF, R. CHATILA, A. De LUCA, A. De SANTIS, G. GIRALT, J. GUIOCHET, G. HIRZINGER, F. INGRAND, V. LIPPIELLO, R. MATTONE, D. POWELL, S. SEN, B. SICILIANO, G. TONIETTI et L. VILLANI. : Safe and dependable physical human-robot interactio in anthropic domains : State of the art and challenges. *In Proceedings IROS Workshop on pHRI, Physical Human-Robot Interaction in Anthropic Domains*, October 2006.
- S ALFAYAD, F.B. OUEZDOU et G. Cheng F. NAMOUN : Lightweight high performance integrated actuator for humanoid robotic applications : Modeling, design & realization. *In Robotics and Automation, 2009. ICRA '09. IEEE International Conference on*, pages 562–567, May 2009.
- E. AMROLLAH et P. HÉNAFF : On the role of sensory feedbacks in rowat-selverston cpg to improve robot legged locomotion. *frontiers in Neurosciences, Neurorobotics*, 4 :113, december 2010.
- J. P. AUBIN : *Viability Theory*. Birkhäuser, 1991.
- R BEER : On the dynamics of small continuous-time recurrent neural networks. *Adaptive Behavior*, 3(4):469–509, 1995a.
- R BEER : Parameter space structure of continuous-time recurrent neural networks. *Neural Computation*, 18:3009–3051, 2006a.
- Randall D. BEER : A dynamical systems perspective on agent-environment interaction. *Artificial Intelligence*, 72(1-2):173 – 215, 1995b. ISSN 0004-3702. URL <http://www.sciencedirect.com/science/article/B6TYF-4007FSV-5/2/e52765c96d8cb2e9319959cf6f0d761a>. Computational research on interaction and agency, part 1.
- Randall D. BEER : Parameter space structure of continuous-time recurrent neural networks. *Neural Computation*, 18(12):3009–3051, 2006b. URL <http://www.mitpressjournals.org/doi/abs/10.1162/neco.2006.18.12.3009>.
- T. BEM et Rinzel J. : Short duty cycle destabilizes a half-center oscillator, but gap junctions can restabilize the anti-phase pattern. *J Neurophysiol*, 91:693–703, 2004.
- H BEN-AHMED et B. MULTON : Les actionneurs électromagnétiques : quels choix pour la robotique ? *In Journées Nationales de la Recherche en Robotique*, octobre 2007.
- Soraya BENSMAIL : *Modélisation du neuro-contrôle dans la simulation et l'analyse du geste de manipulation fine*. Thèse de doctorat, Université de Versailles Saint Quentin, octobre 2005.
- M. BOONYANIT et R.D BEER : Center-crossing recurrent neural networks for the evolution of rhythmic behavior. *Neural computation*, 14:2043–2051, 2002.
- G BOURDON et P HENAFF : Fuzzy and neural control for mobile robotics experimentations. *In IEEE Int. Conference on Knowledge-Based Intelligent Systems*, 1997.

- J.W. BROWN et T.S. BRAVER : Learned predictions of error likelihood in the anterior cingulate cortex. *Science*, 307(5712):1118–1121, 2005.
- O. BRUNEAU et A. DAVID : *Analytical criterions for the generation of highly dynamic gaits for humanoid robots : dynamic propulsion criterion and dynamic propulsion potential*. Humanoid Robots book, new Developments, Advanced Robotic Systems,, 2007.
- A BÜSCHGES : Sensory control and organization of neural networks mediating coordination of multisegmental organs for locomotion. *Journal of Neurophysiology*, 93:1127–1135., 2005.
- U. BÄSSLER et A. BÜSCHGES : Pattern generation for stick insect walking movements-multisensory control of a locomotor program. *Brain Res Rev*, 27:65–88, 1998.
- D. CATTART et D. LERAY : Adaptive motor control in the crayfish. *Progress in Neurobiology*, 63:199–240., 2001.
- Joe CHALFOUN : *Prédiction des efforts musculaires dans le système main / avant bras. Modélisation, Simulation, Optimisation et Validation*. Thèse de doctorat, Université de Versailles Saint-Quentin, avril 2005.
- N. CHEVALIER et K. TAKTEK : *Apprentissage moteur : cahier d'accompagnement*. KIN2300. Librairie Coop. Université du Québec à Montréal, 2004.
- T.W.S. CHOW et L. XIAO-DONG : Modeling of continuous time dynamical systems with input by recurrent neural networks. *Circuits and Systems I : Fundamental Theory and Applications, IEEE Transactions on*, 47(4):575 –578, apr 2000.
- H. CRUSE, C. BARTLING, G. CYMBALYUK, J. DEAN et M DREIFERT : A modular artificial neural net for controlling a six-legged walking system. *Biol. Cybern*, 72:421–430, 1995.
- H. CRUSE, KINDERMANN, Schumm T., J. M., Dean et J SCHMITZ : Walknet-a biologically inspired network to control six-legged walking. *Neural Netw.*, 11:1435–1447, 1998.
- A. DAVID et O. BRUNEAU : Bipedal walking gait generation based on the sequential method of analytical potential (smap). *Journal of Multibody System Dynamics*, to be published, 2011.
- A. C. DE PINE FILHO, M. S. DUTRA et L. S. C. RAPTOPOULOS : Modeling of a bipedal robot using mutually coupled rayleigh oscillators. *Biol. Cybern*, 92:1–7, 2005.
- D. DIMITROV, P.B. WIEBER, O. STASSE, H. J. FERREAU et H. DIEDAM : An optimized linear model predictive control solver for online walking motion generation. *In In Proc. IEEE International Conference on Robotic and Automation*, 2009.
- Stéphane DONCIEUX : *Evolution de contrôleurs neuronaux pour animats volants : méthodologie et applications*. Thèse de doctorat, Université Pierre et Marie Curie, 2003.
- Stéphane DONCIEUX : *Robotique Evolutionniste : conception orientée vers le comportement*. Thèse de doctorat, Université Pierre et Marie Curie, décembre 2010.
- P. F. DOUBLIEZ, O. BRUNEAU et F. B. OUEZDOU : Force control of a biped robot's redundant leg producing large inertia effects to cross obstacles. *International Journal of Humanoid Robots*, to be published, 2011.
- Kenji DOYA et Shuji YOSHIZAWA : Adaptive neural oscillator using continuous-time back-propagation learning. *Neural Networks*, 2(5):375 – 385, 1989. ISSN 0893-6080. URL <http://www.sciencedirect.com/science/article/B6T08-485RHTR-5T/2/23a600b7fcbd555ec61abe1f347bd85d>.
- M. S. DUTRA, A. C. DE PINA FILHO et V. F. ROMANO : Modeling of a bipedal locomotor using coupled nonlinear oscillators of van der pol. *Biol. Cybern.*, 88:286–292., 2003.
- O. EKEBERG : A combined neuronal and mechanical model of fish swimming. *Biol. Cybern*, 69:363–374, 1993.

- E. FAUGLOIRE, B. G. BARDY et T. A. STOFFREGEN : The dynamics of learning new postural patterns. *Journal of Motor Behavior*, 3838(4):299–312, 2006.
- D. FLOREANO et J. URZELAI : Evolution of neural controllers with adaptive synapses and compact genetic encoding. In *Proceedings of the 5th European Conference on Artificial Life*, volume 1974, pages 183–194. Lecture notes In Computer Science, 1999.
- D. FLOREANO et J. URZELAI : Evolutionary robots : the next generation. In *Proceedings of the Seventh International Symposium on Evolutionary Robotics : From intelligent Robots to Artificial Life*, volume 9. AAI Book, T. Gomi Ed, 2000.
- A. FRIGON et R. ROSSIGNOL : Experiments and models of sensorimotor interactions during locomotion. *Biological Cybernetic*, 95:607–627, 2006.
- J.C GALLAGHER, Beer RD, Espenschied KS et Quinn RD : Application of evolved locomotion controllers to a hexapod robot. *Robotics and Autonomous Systems*, 19:95–103, 1996.
- W. GERSTNER et WM KISTLER : *Spiking Neuron Models. Single Neurons, Populations, Plasticity*. Gerstner W, WM Kistler, cambridge university press édition, 2002.
- M. GOLUBITSKY, I. STEWART, P. L. BUONO et J. J. COLLINS : Symmetry in locomotor central pattern generators and animal gaits. *Nature*, 401:693–695, 1999.
- A. GOSWAMI et V. KALLEM : Rate of change of angular momentum and balance maintenance of biped robots. In *Robotics and Automation, 2004. Proceedings. ICRA '04. 2004 IEEE International Conference on*, volume 4, pages 3785 – 3790 Vol.4, 26-may 1, 2004.
- F. GRAVEZ, O. BRUNEAU et F.B. OUEZDOU : Analytical and automatic modeling of digital humanoids. *International Journal of Humanoid Robotics*, 2(3):337–359, 2005.
- S. GRILLNER : Neurobiological bases of rhythmic motor acts in vertebrates. *Science*, 228:143–149, 1985.
- S. GROSSBERG : On learning and energy-entropy dependence in recurrent and nonrecurrent signed networks. *Journal of Statistical Physics*, 1:319–350, 1969.
- A. GUILLOT et J. A. MEYER : *la contribution de l'approche Animat aux sciences cognitives*. Agnes Guillot A et Jean Arcady Meyer, 2007.
- Herbert HATZE : The complete optimization of a human motion. *Mathematical Biosciences*, 28(1-2):99 – 135, 1976. ISSN 0025-5564. URL <http://www.sciencedirect.com/science/article/pii/0025556476900985>.
- D. HEBB : *The Organization of Behavior : A Neuro-physiological Theory*. New York. John Wiley and Sons., 1949.
- P HENAFF et S DELAPLACE : Using backpropagation algorithm for neural adaptive control : Experimental validation on an industrial mobile robot. In *Proc. of Int. Symp. CISM-IFTOMM Robots Manipulators Systems*, 1996.
- A.V. HILL : The heat of shortening and the dynamic constants of muscle. *Proceedings of the Royal Society*, B126:136–195, 1938.
- Patrick HÉNAFF, Vincent SCESA, Fethi Ben OUEZDOU et Olivier BRUNEAU : Real time implementation of ctrnn and bptt algorithm to learn on-line biped robot balance : Experiments on the standing posture. *Control Engineering Practice*, 19(1):89 – 99, 2011. ISSN 0967-0661. URL <http://www.sciencedirect.com/science/article/B6V2H-51COK89-1/2/17c7e19e4a2e614cc0a834e48c749ec0>.
- T. HOINVILLE, Tapia C. et P. HÉNAFF : Flexible and stable pattern generation by evolving constrained plastic neurocontrollers. *Adaptive Behavior*, to be published., 2011.

- T. HOINVILLE et P. HENAFF : Comparative study of two homeostatic mechanisms in evolved neural controllers for legged locomotion. *In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems(IROS 2004)*, volume 3, pages 2624–2629, 2004a.
- T HOINVILLE et P HENAFF : Evolving plastic neural controllers stabilized by homeostatic mechanisms for adaptation to a perturbation. *In Artificial life IX : proceedings of the Ninth International Conference on the Simulation and Synthesis of Living Systems (ALIFE9)*, 2004b.
- T. HOINVILLE, P. HÉNAFF et S. DELAPLACE : Etude sur l'intérêt des modèles biologiques de réseaux de neurones pour la synthèse de rythmes locomoteurs adaptatifs. *Journal Européen des Systèmes Automatisés, RS-JESA*, 41(3/4):413–435, 2007.
- Thierry HOINVILLE : *Évolution de contrôleurs neuronaux plastiques. De la locomotion adaptée vers la locomotion adaptative*. Thèse de doctorat, Université de Versailles St Quentin, octobre 2007.
- J. J. HOPFIELD : eurons with graded response properties have collective computational properties like those of two-state neurons. *Proceedings of the National Academy of Sciences*, 81:3088–3092, 1984.
- A. J. IJSPEERT : A connectionist central pattern generator for the aquatic and terrestrial gaits of a simulated salamander. *Biological Cybernetics*, 84(5):331–348, 2001.
- A. J. IJSPEERT : Central pattern generators for locomotion control in animals and robots : A review,. *Neural Networks*, 21, Issue 4:642–653, 2008.
- A J IJSPEERT, JCT HALLAM et D. J. WILLSHAW : From lampreys to salamanders : Evolving neural con-trollers for swimming and walking. *In Proceedings of the Fifth Intl Conf on Simulation of Adaptive Behavior*, pages 390–399, 1998.
- J. IJSPEERT, A., A. CRESPI, D. RYCZKO et J-M. CABELGUEN : From swimming to walking with a salamander robot driven by a spinal cord model. *Science*, 315, 2007.
- S. KAJITA, F. KANEHIRO, K. KANEKO, K. FUJIWARA, K. HARADA et andHirukawa H YOKOI, K. : Biped walking pattern generation by using preview control of zero-moment point. *In In Proc. IEEE Internationale Conference Robotic and Automation*, volume 2, pages 1620–1626, 2003.
- B. KOUCHMESHKY, W. AQUINO, H. LIPSON et J. C. BONGARD : Coevolutionary strategy for structural damage identification using minimal physical testing. *International Journal for Numerical Methods in Engineering.*, 69(5):1085–1107., 2006.
- M. LAGARDE, P. ANDRY, P. GAUSSIER, S. BOUCENNA et L. HAFEMEISTER : *Proprioception and Imitation : On the Road to Agent Individuation*, volume 264, pages 43–63. Springer Berlin / Heidelberg, 2010.
- Mathieu LAGARDE : *Apprentissage de nouveaux comportement : vers le développement épigénétique d'un robot autonome*. Thèse de doctorat, Universtité de Cergy Pontoise, 13 juillet 2010.
- C. S. LEE et R. V. GONZALEZ : Fuzzy logic versus a pid controller for position control of a muscle-like actuated arm. *Journal of Mechanical Science and Technology*, 22:1475–482, 2008.
- A. R. MARCHAND, W. J. P. BARNES et D. CATTAERT : Primary afferent depolarizations of sensory origin within contact- sensitive mechanoreceptive afferents of a crayfish leg. *J. Neurophysiol.*, 77:3340–3354, 1997.
- E. MARDER et D. BUCHER : Central pattern generators and the control of rhythmic movements. *Current Biology*, Vol. 11, Issue 23:986–996, 2001.
- B MATHAYOMCHAN et R BEER : Center-crossing recurrent neural networks for the evolution of rhythmic behavior. *Neural Computation*, 14:2043–2051, 2002.
- K. MATSUMOTO, W. SUZUKI et K. TANAKA : Neuronal correlates of goalbased motor selection in the prefrontal cortex. *Science*, 301(5630):229–232., 2003.

- K. MATSUOKA : Mechanisms of frequency and pattern control in the neural rhythm generators. *Biol Cybern*, 56:345–353, 1987.
- D. A. MC CREA et I.A. RYBACK : Organization of mammalian locomotor rhythm and pattern generation. *Brain Research Review*, 57, Issue 1:134–146, 2008.
- T. MC GEER : Passive dynamic walking. *International Journal of Robotics Research*, 9(2):62–82., 1990.
- W MCCULLOCH et W PITTS : A logical calculus of ideas immanent in nervous activity. *Bulletin of Mathematical Biophysics*, 5:115–33, 1943.
- Artem MELNYK, Patrick HÉNAFF, Sebastien RAZAKARIVONY, Volodymyr BORYSENKO et Philippe GAUSSIER : Adaptive behavior of an electromechanical arm robot in case of physical interaction with the human. *In proceedings of 2011 IEEE/ASME AIM International Conference on Advanced Intelligent Mechatronics*, Juillet 2011.
- JA MEYER, S. DONCIEUX, D. FILLIAT et A. Guillot A : *Biologically inspired robot behavior engineering. Evolutionary Approaches to Neural Control of Rolling, Walking, Swimming and Flying Animals or Robots*. Springer-Verlag, r.j. d. j. santos, and m. grana édition, 2002.
- J. MORIMOTO, G. ENDO, J. NAKANISHI, S. HYON et D. CHENG, G. and Bentivegna : Modulation of simple sinusoidal patterns by a coupled oscillator model for biped walking. *In Proceedings of the 2006 IEEE international conference on robotics and automation*, pages 1579–584, 2006.
- Franck MULTON : *HABILITATION À DIRIGER DES RECHERCHES "Analyse, Modélisation et Simulation du Mouvement Humain"*. Thèse de doctorat, Université de Rennes 1, 2006.
- Yuichi NAKAMURA et Masahiro NAKAGAWA : Approximation capability of continuous time recurrent neural networks for non-autonomous dynamical systems. *In Proceedings of the 19th International Conference on Artificial Neural Networks : Part II*, pages 593–602, Limassol, Cyprus, 2009. Springer-Verlag.
- J NASSOUR, P HENAFF, B OUEZDOU et G CHENG : Experience-based learning mechanism for neural controller adaptation : Application to walking biped robots. *In Proceedings of 2009 IEEE/RSJ International Conference on Intelligent Robots and Systems IROS 2009*, pages 2616–2621, 2009.
- J NASSOUR, P HENAFF, FB OUEZDOU et G CHENG : A study of adaptive locomotive behaviors of a biped robot : Patterns generation and classification. *In 11th International Conference on Simulation of Adaptive behavior : From Animals to Animals*. books.google.com, 2010a.
- J NASSOUR, P HENAFF, FB OUEZDOU et G CHENG : A study of adaptive locomotive behaviors of a biped robot : Patterns generation and classification. *In International Conference on Simulation of Adaptive behavior, SAB 2010*. hal.archives-ouvertes.fr, 2010b.
- Mai H. NGUYEN et Garrison W. COTTRELL : Tau net a neural network for modeling temporal variability. *Neurocomputing*, 15(3-4):249–271, juin 1997. ISSN 0925-2312. URL <http://www.sciencedirect.com/science/article/B6V10-3SNVKY1-6/2/1c3e8ed79abe39a80ab5f22123498b04>.
- NOLFI : Evolving non-trivial behaviors on real robots : a garbage collecting robot. *Robotics and Autonomous System*, 22:187–198, 1997.
- A. OHALLORAN et F. OMALLEY : *Materials and technologies for artificial muscle : A review for the mechanical muscle project*. Department of Electronic Engineering, National University of Ireland Galway, 2004. URL [Availableonlineathttp://www.tcd.ie/bioengineering/documents/ChapterVII000.pdf](http://www.tcd.ie/bioengineering/documents/ChapterVII000.pdf).
- PINEDA et J. FERNANDO : Generalization of back-propagation to recurrent neural networks. *Phys. Rev. Lett.*, 59(19):2229–, novembre 1987. URL <http://link.aps.org/doi/10.1103/PhysRevLett.59.2229>.

- A. PITTI, M. LUNGARELLA et Y. KUNIYOSHI : Generating spatiotemporal joint torque patterns from dynamical synchronization of distributed pattern generators. *Frontiers in Neurorobotics*, Vol 3, Issue 2, 2009.
- L. RIGHETTI et A.J. IJSPEERT : Programmable central pattern generators : an application to biped locomotion control. In *Proceedings 2006 IEEE International Conference on Robotics and Automation, 2006. ICRA 2006.*, pages 1585–1590, mai 2006.
- Ludovic RIGHETTI, Jonas BUCHLI et Auke Jan IJSPEERT : Dynamic hebbian learning in adaptive frequency oscillators. *Physica D : Nonlinear Phenomena*, 216(2):269 – 281, 2006. ISSN 0167-2789. URL <http://www.sciencedirect.com/science/article/B6TVK-4JV5M9M-2/2/1b5a464d7b45db2e81b181050ea3af61>.
- F. ROWAT et I. SELVERSTON : Modeling the gastric mill central pattern generator of the lobster with a relaxation-oscillator network. *Neurophysiology*, Vol. 70, No. 3:1030–1053, 1993.
- V. SCESA : *Contrôle adaptatif par réseaux de neurones : évaluation d'une implémentation temps réel sur un contrôleur industriel multiprocesseur pour le contrôle du comportement dynamique d'une structure mécanique complexe*. Thèse de doctorat, Thèse de Doctorat de l'Université de Versailles St Quentin, octobre 2006.
- V SCESA, P HENAFF, FB OUEZDOU et F NAMOUN : On the analysis of sigmoid time parameters for dynamic truncated bptt algorithm. In *Neural Networks, 2006. IJCNN'06. International Joint Conference on Neural Networks*. ieeexplore.ieee.org, 2006a.
- V SCESA, P HENAFF, FB OUEZDOU et F NAMOUN : Time window width influence on dynamic bptt (h) learning algorithm performances : Experimental study. In *Lecture Notes in Computer Science*. Springer, 2006b. URL <http://www.springerlink.com/index/14628378261k4r13.pdf>.
- V SCESA, B MOHAMED, P HENAFF et FB OUEZDOU : Dynamic recurrent neural network for biped robot equilibrium control : Preliminary results. In *Interbational Conference on Robotics and Automation, ICRA 2005*, 2005.
- L. SENTIS, J. PARK et O. KHATIB : Compliant control of multi-contact and center of mass behaviors in humanoid robots. *IEEE Trans. on Robotics*, 26(3):483–501, jun 2010.
- H. SERHAN, P. HENAFF, C. NASR et F. OUEZDOU : State machine-based controller for walk-halt-walk transitions on a biped robot. In *Proceedings of the 8th IEEE-RAS International Conference on Humanoid Robots, Humanoids 2008*, pages 535–540, Dec. 2008a.
- H SERHAN, P HENAFF, C NASR et F B OUEZDOU : Dynamic control strategy of a biped inspired from human walking. In *Proceedings of 2008 IEEE-RAS/EMBS Inter. Conference on Biomedical Robotics and Biomechatronics-BioRob*. ieeexplore.ieee.org, 2008b.
- H SERHAN, C NASR et P HENAFF : Designing a muscle like system based on pid controller and tuned by neural network. In *Neural Networks, 2006. IJCNN'06. International Joint Conference on Neural networks*. ieeexplore.ieee.org, 2006.
- H. SERHAN, C. NASR, P. HENAFF et F. OUEZDOU : A new control strategy for robian biped robot inspired from human walking. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2008*, pages 2479–2485, Sept. 2008c.
- H. SERHAN, C. NASR et P. HÉNAFF : Muscle emulation with dc motor and neural networks for biped robots. *International Journal of Neural Systems*, 10:341–353, 2010. URL <http://www.worldscinet.com/abstract?id=pii:S0129065710002450>.
- Hayssam SERHAN : *Approches Biomimétiques pour le contrôle de la marche des robots bipèdes : stratégie intuitive et intégration d'un modèle de muscle*. Thèse de doctorat, Université de Versailles St Quentin, juillet 2009.

- R. SHADMEHR : Computational approaches to motor control. *Encyclopedia of Neuroscience*, 3:9–17, 2009.
- ML SHIK, FV SEVERIN et GN ORLOWSKY : Control of walking and running by means of electrical stimulation of the mid- brain. *Biofizika*, 11:659–666, 1966.
- D. SHIN, I. SARDELLITTI, Y.-L. PARK, O. KHATIB et M. CUTKOSKY : Design and control of a bio-inspired human-friendly robot. *The International Journal of Robotics Research*, 29(5):571–584, apr 2010.
- I. SHINKICHI, Y. HIDEO, S. TAKANORI et Tamio A : Wave cpg model for autonomous decentralized multi-legged robot : Gait generation and walking speed control. *Robotic and Autonomous Systems*, 54:118–126, 2006.
- B. SIRI, H. BERRY, B. CESSAC, B. DELORD et M. QUOY : A mathematical analysis of the effects of hebbian learning rules on the dynamics and structure of discrete-time random recurrent neural networks. *Neural Computation*, 20(12):2937–2966, 2008.
- G. TAGA, Y. YAMAGUCHI et H. SHIMIZU : Self-organized control of bipedal locomotion by neural oscillators in unpredictable environment. *Biological Cybernetics*, 65:147–159, 1991. ISSN 0340-1200. URL <http://dx.doi.org/10.1007/BF00198086>. 10.1007/BF00198086.
- R. D. TLALOLINI, Y. Aoustin et C. CHEVALLEREAU : Design of a walking cyclic gait with single support phases and impacts for the locomotor system of a thirteen-link 3d biped using the parametric optimization. *Multibody System Dynamics*, 2009.
- N. F. TROJE, J. SADR, H. GEYER et K NAKAYAMA : Adaptation aftereffects in the perception of gender from biological motion. *Journal of Vision*, 6(8):850–857, 2006. URL <http://www.biotionlab.ca/adaptation.php>.
- M. VUKOBRATOVIC et B. BOROvac : Zero-moment point thirty five years of its life. *International Journal of Humanoid Robotics*, 1:157–173, 2004.
- T. WADDEN et O. EKEBERG : A neuro-mechanical model of legged locomotion : Single leg control. *Biological Cybernetics*, Vol. 79, No2:161–173, 1998.
- Ekeberg Ö. WADDEN T. et Lansner A : Towards ann based control of simulated legged locomotion. *In Proc. Mechatronical Computer Systems for Perception and Action*, pages 379–382., June 1-3 1993.
- Y. WOOSUNG, Ra S. Chang H. K. CHONG, N. Y. et J. Y. BUM : Self-stabilizing bipedal locomotion employing neural oscillators,. *In Proceedings of 8th IEEE-RAS International Conference on Humanoid robots*,, pages 8–15, 2008.
- F. WORGOTTER, T. GENG et B PORR : Fast biped walking with a sensor-driven neuronal controller and real-time online learning. *The International Journal of Robotics Research*, Vol. 25:243–259, 2006.

Troisième partie

Annexe

Chapitre 9

Sélection d'articles

- 9.1 Real time implementation of CTRNN and BPTT algorithm to learn on-line biped robot balance : experiments on the standing posture. Control Engineering and Practice, 2010.



Contents lists available at ScienceDirect

Control Engineering Practice

journal homepage: www.elsevier.com/locate/conengprac

Real time implementation of CTRNN and BPTT algorithm to learn on-line biped robot balance: Experiments on the standing posture

Patrick Hénaff^{a,*}, Vincent Scesa^b, Fethi Ben Oueddou^b, Olivier Bruneau^b

^a ETIS, UMR 8051, CNRS-ENSEA-UCP, University of Cergy-Pontoise, Site de ST Martin, 2 rue A. Chauvin, 95302 Cergy-Pontoise Cedex, F-9500, France

^b LISV, University of Versailles Saint-Quentin, France

ARTICLE INFO

Article history:

Received 7 December 2009

Accepted 6 October 2010

Available online 30 October 2010

Keywords:

Neural control

Learning algorithms

CTRNN

Real-time systems

Robotics

Biped robot

ABSTRACT

This paper describes experimental results regarding the real time implementation of continuous time recurrent neural networks (CTRNN) and the dynamic back-propagation through time (BPTT) algorithm for the on-line learning control laws. Experiments are carried out to control the balance of a biped robot prototype in its standing posture. The neural controller is trained to compensate for external perturbations by controlling the torso's joint motions. Algorithms are embedded in the real time electronic unit of the robot. On-line learning implementations are presented in detail. The results on learning behavior and control performance demonstrate the strength and the efficiency of the proposed approach.

© 2010 Elsevier Ltd. All rights reserved.

1. Introduction

Technological developments have enabled us to build robots with morphologies that are inspired by animals or humans. Therefore, the most recent humanoid robots are technologically complex systems, with an extremely high level of mechanical and electronic integration. They are equipped with complete perceptive systems that enable them to interact with human beings and to move in an environment built for human life. One of the most important difficulties in controlling humanoid robots is maintaining balance during walking or standing. One solution to this problem is to design controllers based on the zero moment point (ZMP) theory (Vukobratovic, 2004). Another method is to design controllers using bio-inspired approaches, i.e., with some capabilities of adaptation and training, leading to the acquisition of reflexes.

Using biologically inspired architectures such as neural networks that are able to learn the “correct” control of the robot's equilibrium is a promising approach. For this purpose, several neural controller-based approaches have been proposed in the past. Cerebellar model articulation controllers (CMAC) proposed in 1975 by Albus (1975) are still studied in the control of legged robots. Recent studies deal with their modeling and generalization properties (Horvath & Szabo, 2007) or with their connections to other approaches like fuzzy logic (Su, Lee,

& Wang, 2006) or computed torque control (Lin & Chen, 2007). CMAC have been used to control the balance of biped robots (Kun & Miller, 1996) or for robust dynamic walking in simulation (Lin & Chen, 2007) and for biped robot experiments (Sabourin & Bruneau, 2005).

Recurrent neural networks (i.e., dynamic neural nets) have been extensively studied in the control of complex systems for many years (Marcua, Köppen-Seligerb, & Stücher, 2008; Song & Tahk, 2001). These artificial neural networks have also been used to design stable walking gaits for biped robots (Wu, Song, & Yang, 2007). Several approaches were based on evolutionary synthesis (Fukuda, Komata, & Arakawa, 1997), neural oscillators (Taga, Yamaguchi, & Shimizu, 1991; Geng, Porr, & Wörgötter, 2006), and central pattern generators (Nakanishi et al., 2004; Righetti & Ijspeert, 2006). Recently, researchers have used RNN as predictive compensator (Mizunoa, Kurodaa, Okazakib, & Ohtsu, 2007) or tracking controllers with self-constructing properties. Self-constructing algorithms are very interesting approaches because they allow for optimizing on-line the neuronal controller architecture in order to insure the best control performance. In Gao and Er (2003), a self-constructing fuzzy neural controller was proposed for the tracking control of a simulated planar robot manipulator with two degrees of freedom. In Hsu (2009) a simple growing and pruning algorithm applied to a recurrent neural network has been tested in experimentation to control one degree of freedom of a moving table with a linear ceramic motor system.

Many studies on dynamic neural controllers of robots have focused on continuous-time recurrent neural networks (CTRNN) due to their ability to be universal approximators (Beer, 2006). CTRNNs have been used for bio-inspired control because of their abilities to reproduce the full qualitative range of nerve cell

* Corresponding author. Tel.: +33 1 34 25 28 51; fax : +33 1 34 25 28 39.

E-mail addresses: patrick.henaff@u-cergy.fr (P. Hénaff),

fethi.benouezdou@uvsq.fr (F. Ben Oueddou), bruneau@lisv.uvsq.fr (O. Bruneau).

phenomenology (Beer, 2006). They make it possible to show adaptivity properties based on homeostatic plastic mechanisms (Hoinville & Hénaff, 2004; Williams, 2007). Moreover, they may be a fine model for generating adaptive behavior because they can learn with the back-propagation through time algorithm (BPTT) (Pineda, 1987; Pearlmutter, 1995; Rumelhart, Hinton, & Williams, 1986; Robinson & Fallside, 1987; Werbos, 1990). Unfortunately, these algorithms are complex to implement in real time application, especially for the on-line training of real robots because of the shrinking gradient problem of recurrent neural networks (this gradient instability problem was first studied in Scesa, Hénaff, Ouedzou, & Namoun, 2006).

There are still only a few recent in-depth studies on real time implementation, especially with the concept of learning the equilibrium reflex for a biped robot on-line. The scientific objective of the work presented in this paper is to perform an in-depth analysis of the real time performance of continuous time recurrent neural network controllers for learning balance reflexes for a biped robot. In particular, robot abilities to learn on-line with real time constraints are investigated.

Several contributions in this area have already been published (see Scesa, Mohamed, Henaff, & Ouedzou, 2005). This paper focuses on describing the real time implementation of the learning algorithms embedded into the robot control unit. Some promising new experimental results are also reported.

This paper is organized as follows. The second section deals with the learning models based on the CTRNN and BPTT algorithms. The third section presents the fundamental principles of CTRNN and BPTT. In the fourth section, the experimental biped platform called ROBIAN is described. The real time implementation of the learning algorithm based on CTRNNs and the back-propagation through time algorithm is detailed in Section 5. The sixth section, describes the experimental results in two subparts: a feasibility test for the embedded learning algorithm on controlling the ROBIAN biped torso and a new approach of on-line learning of the equilibrium reflex. Finally, the last section presents the conclusions and potential further developments stemming from this work.

2. Learning equilibrium with CTRNN

It is well known that the human torso attempts to stabilize the whole body when walking (Hyon, 2009; Kubica, Wang, & Winter, 2001; Setiawan, Hyon, Yamaguchi, & Takanishi, 1999). It is also known that biped robot balance can be considered to be a global behavior because the upper part (the torso) and the lower part (the legs) of the robot interact with and disturb each other (see Fig. 1) during walking. This internal interaction can be taken into account in the synthesis of the biped robot controller. For example, Morimoto et al. (2006) showed that a biped robot can stop and walk using simple sinusoidal desired trajectories with their phases adjusted by a coupled oscillator model.

During the walking and halting phases, the robot's balance can be affected by perturbations like external forces applied to its body, including the upper part. One way to control the balance of the robot is to drive the movements of the upper part (i.e., the torso) in order to minimize the perturbations (forces and moments) exerted on the lower part of the robot.

To measure this perturbation, the robot is equipped with a six-force sensor fixed between the torso and the lower part (see Fig. 1). Then, if the forces and moments measured by the sensor are close to zero, the equilibrium of the robot is assumed to be controlled (Mohammed, Gravez, & Ouedzou, 2004).

To achieve this goal, the purpose of this experiment is to learn on-line, using a CTRNN, how to control the robot torso in order to

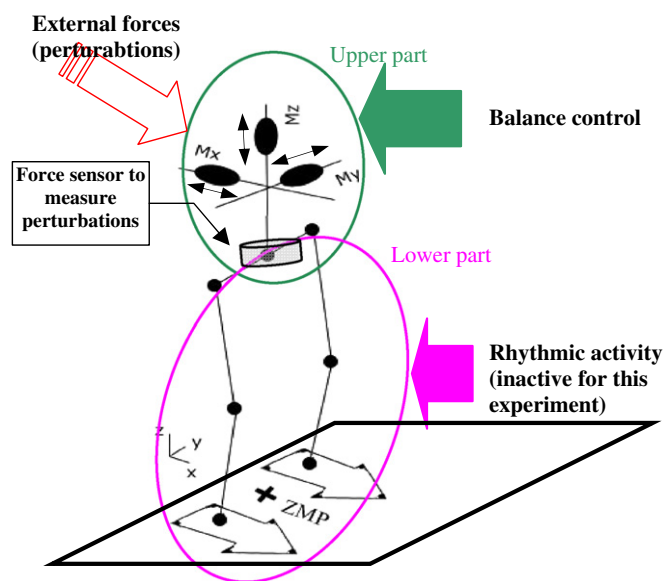


Fig. 1. Control of the biped robot's balance: perturbations caused by unknown external forces can be measured with a force sensor placed between upper and lower part.

reduce the external perturbations measured by the sensor. This learning is carried out with the biped robot in the standing posture when unknown external forces are applied to its torso.

The control of robot balance must take into account the real features of the mechanism and the external phenomena that are not modeled in the simulation (friction, motor properties, attrition, ground slope and passive prosthesis use...). Information on these phenomena is often only available through their effects on the total energy of the system. Hence, taking into account these phenomena into the equilibrium control is a difficult task. To avoid this obstacle, the controller should be able to adapt its behavior in real time, following a cost function that incorporates information on these phenomena. Moreover, in an optimal control approach, time variations must be included in the adaptation process.

To meet these conditions, dynamic recurrent neural networks with back-propagation through a time learning process was chosen as the most appropriate network. The proposed learning control architecture is shown in Fig. 2. The outputs of the network are positions (X,Y,Z) that the torso mechanism then has to reach. Thus, at each time step, the net modifies the actuator speed and accelerations are generated. Consequently, forces between the upper and lower parts of the robot will be produced to compensate the external perturbations.

The inputs should show the network the current state of the system. They must be enough representatives to compute a correct control response. They consist of the measured components provided by the 6-forces sensor and the positions of the current motors given by incremental encoders.

To carry out the on-line learning of an optimal control, the parameters of the net must be adapted in real time while the system is running. Back-propagation through time (BPTT), with its ability to integrate the error in the network at each instant is an appropriate solution to solve the parameter adaptation problem. In BPTT, the network is first unrolled in time creating a layer per time step. Then, this algorithm back-propagates the output error on these virtual temporal layers as the classical back-propagation algorithm does on existing ones. The result is the computation of the error gradient in the network and its integration in time. Then, the parameters are modified by the gradient descent algorithm, and the response of the net approaches the

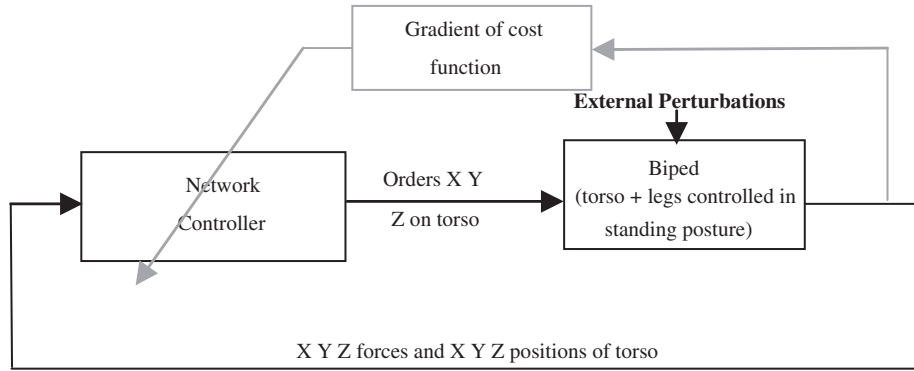


Fig. 2. Principle of the on-line learning balance control scheme. Grey lines represent the learning process with cost computation and modification of the net. Each joint position is measured with an incremental electronic sensor. Forces between the upper and lower parts are measured with the six-force sensor. All signals are fed back to the inputs of the control boards of the electronic controller unit (see Section 5).

desired behavior. Instead of the output error, a cost function that represents the aim of the control can be used for the learning stage. The aim is to bring the forces measured by the sensor closer to zero so that the cost is define by the sum of the squared force components measured by the six axes force sensor in the X, Y and Z directions

$$Cost(t) = F_x^2(t) + F_y^2(t) + F_z^2(t) \quad (1)$$

This cost function is given for the learning algorithm through a gradient computation for each net output

$$e_j(t) = \frac{\partial Cost(t)}{\partial output_j} \quad (2)$$

This equation expresses the influence of each command on the cost function at time t . BPTT will use this gradient to minimize the integral of the cost function (Eq. (1)) over the learning time window. Thus, the cost function computed to optimize the control has the following expression:

$$E = \int_{t_0}^t [F_x^2(t) + F_y^2(t) + F_z^2(t)] dt \quad (3)$$

Instead of computing the exact expression of (3), which would require a strict knowledge of the system, it is possible to use a signal that carries the same information. For the torso experiment, it is defined by the following equations:

$$e_z(t) = F_z(t) \quad e_x(t) = F_x(t) \quad e_y(t) = F_y(t) \quad (4)$$

3. Principles of CTRNN and BPTT

Neurons of a dynamic recurrent neural network are driven by the following equation, derived from (Beer, 2006; Pineda, 1987; Pearlmutter, 1995):

$$T_i \frac{\partial y_i}{\partial t} = -y_i + f(x_i) \quad (5)$$

where y_i is the output of neuron i , x_i the weighted sum of the j th neuron inputs, f the activation function (tanh or sigmoid) and T_i the time constant of the i th neuron

3.1. Propagation algorithm

The input data provided to the network are propagated to generate output responses. The propagation depends on the

intrinsic parameters of the network and the neurons, which are given as follows for the j th neuron:

- ω_{ij} = weight of the connection from neuron i to neuron j
- b_j = bias of the j th neuron
- T_j = time constant of the j th neuron.

By the progressive (i.e., explicit) Euler approach, for the Δt time step, and the discrete propagation equation depends on the following relations (Nguyen & Cottrell, 1997):

- Input sum for the j th neuron:

$$x_j(t) = \sum_i [\omega_{ij} \cdot y_i(t - \Delta t)] + b_j(t) \quad (6)$$

- Output of the j th neuron

$$y_j(t + \Delta t) = \frac{\Delta t}{T_j} f(x_j(t)) + \left(1 - \frac{\Delta t}{T_j}\right) y_j(t) \quad (7)$$

In this equation, the $\Delta t/T_j$ term is a scale parameter. Its value, which cannot be equal to zero, lies in the interval]0;1] and expresses the response speed of the j th neuron.

By defining a new variable $S_j = \Delta t/T_j$, Eq. (7) can be rewritten as follows:

$$y_j(t + \Delta t) = S_j f(x_j(t)) + (1 - S_j) y_j(t) \quad (8)$$

Fig. 3 graphically represents the principles of the propagation through time algorithm. At time t , each neuron (except input neurons) receives the $(t - 1)$ outputs of each other neuron.

3.2. Truncated dynamic BPTT learning algorithm

The time constant parameter of each unit reflects the dynamic aspect of the net. However, the learning process must be able to teach it. For this purpose, Pearlmutter (1995) proposed, a dynamical BPTT version for which the time constant, the weights and the bias parameters can be adjusted. The only remaining problem is the memory needed for BPTT algorithms. To compute the exact error gradient, the algorithm has to store all of the network states from the beginning to the current time step. To avoid this excessive memory use, a truncated version, in which the states of the net are only stored during a time window that follows the current instant, can be used (Williams & Zipser, 1989; Williams & Peng, 1990). Therefore, it only computes an approximation that approaches the exact gradient as the length

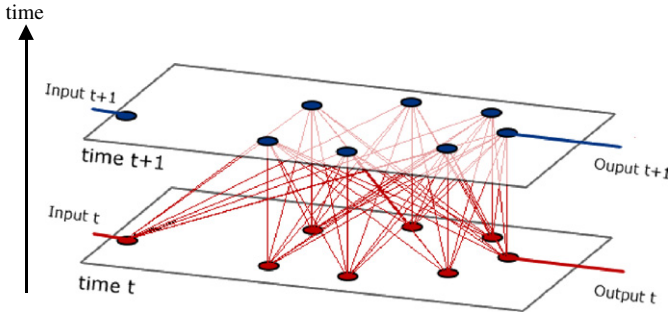


Fig. 3. Equivalent scheme of propagation in time. Example of a 1-6-1 network (1 input, 6 hidden neurons, 1 output) on two time steps.

of the time window increases. Therefore, the learning algorithm used in the experiments will be a truncated dynamic version of the back-propagation through time algorithm, called TDBPPT.

This learning process begins with an error computation at the output of the net. It represents the remaining gap between the current response and the aim of the learning. This error could simply be a squared difference between the desired and computed response. It could also be the result of a more complex computation, e.g., the equilibrium of a biped robot. The objective of the learning process consists of modifying the network parameters (weights, biases and time scale) in order to minimize a desired criterion. In a control application, this criterion would be the gap between the desired state of the system and the actual state. In an identification process, the criterion would be the error between the neural model and the system being taught.

At first, the algorithm computes an error function that corresponds to the criterion to be minimized. This function, E , is the error integral between t_0 and the current time step, t , expressed as follows:

$$E = \int_{t_0}^t \sum_{j=1}^{n_s} e_j(\tau) d\tau \quad (9)$$

where n_s is the number of output neurons, $e_j(\tau)$ the output error of the j th neuron stored at time τ and t_0 the beginning of the integration window. The BPTT algorithm carries out gradient based learning. Thus, the parameters are modified by the negation of the error gradient of each parameter

$$\Delta\omega_{jk} = -\eta \frac{\partial E}{\partial\omega_{jk}} \quad \Delta b_j = -\eta \frac{\partial E}{\partial b_j} \quad \Delta T_j = -\eta \frac{\partial E}{\partial T_j} \quad (10)$$

where η is the learning step. The parameters of the learning process are the learning step and the time window width ($t - t_0$).

3.2.1. Calculation of the gradient components

The back-propagated cost attached to the j th neuron is defined by

$$Z_j(\tau) = \frac{\partial E}{\partial y_j(\tau)} \quad (11)$$

and Z_k is the back-propagated cost attached to the downstream neuron k for each components of the gradient.

The calculation of the gradient components for each parameter is carried out on the network “unrolled in time”. Each component is estimated for the completely unrolled network, but it results in the values of instantaneous components of the gradients, called the back-propagated gradient components. The total value is the integral, over the unrolled network, of the back-propagated values given by Eq. (14), for the j th neuron connected to the k th neuron by the weight ω_{jk} .

By the formula of derivative of the composed functions, the back-propagated components of the gradient can be expressed as follows (see Appendix A for details):

- Gradients on weights

$$\frac{\partial E}{\partial\omega_{jk}} = \int_{t_0}^t \left[\frac{\partial E}{\partial\omega_{jk}}(\tau) \right] d\tau = S_k \int_{t_0}^t [Z_k(\tau) f'(x_k(\tau - \Delta t)) y_j(\tau - \Delta t)] d\tau \quad (12)$$

where f' is the derivative of the sigmoidal activation function.

- Gradients on bias

$$\frac{\partial E}{\partial b_j} = \int_{t_0}^t \left[\frac{\partial E}{\partial b_j}(\tau) \right] d\tau = S_j \int_{t_0}^t [Z_j(\tau) f'(x_j(\tau - \Delta t))] d\tau \quad (13)$$

- Gradients on scale parameters

$$\frac{\partial E}{\partial T_j} = \int_{t_0}^t \left[\frac{\partial E}{\partial T_j}(\tau) \right] d\tau = \frac{-S_j^2}{\Delta t} \int_{t_0}^t [Z_j(\tau) \cdot (f(x_j(\tau - \Delta t)) - y_j(\tau - \Delta t))] d\tau \quad (14)$$

3.2.2. Calculation of back-propagated costs

The value of the back-propagated cost Z_j expresses the influence of the output of the j th neuron, at a given time on the current cost. Its computation uses the instantaneous error values $e_j(\tau)$, calculated outside the network by the comparison of the current state of the network and the objective to be reached.

For each neuron at one instant, τ of the past, the back-propagation cost comes from three sources. Indeed, the back-propagation “goes up” step-by-step through the neurons from the current activities of the network to the oldest states. The back-propagated costs $Z_j^1(\tau)$ and $Z_j^2(\tau)$ are present if the neuron is static or dynamic, and $Z_j^3(\tau)$ appears only for dynamic neurons. $Z_j^1(\tau)$ corresponds to the error coming at the following instants from the synapses connecting the neurons. $Z_j^2(\tau)$ represents the error coming directly from the output neurons (it does not exist if the neuron is hidden). $Z_j^3(\tau)$ corresponds to the error coming at the following instants by means of the internal dynamics of the neuron itself. Hence, the total backpropagation cost $Z_j(\tau)$ is

$$Z_j(\tau) = Z_j^1(\tau) + Z_j^2(\tau) + Z_j^3(\tau) \quad (15)$$

Fig. 4 represents the calculation of these three elements of the back-propagated cost through the j th neuron.

The calculation of these elements is detailed in Appendix B, and it finally yields the total back-propagated cost

$$Z_j(\tau) = \sum_{k=1}^n [Z_k(\tau + \Delta t) S_k f'(x_k(\tau)) \omega_{jk}(\tau)] + \frac{\partial e_j(\tau)}{\partial y_j(\tau)} \Delta t + Z_j(\tau + \Delta t) (1 - S_j) \quad (16)$$

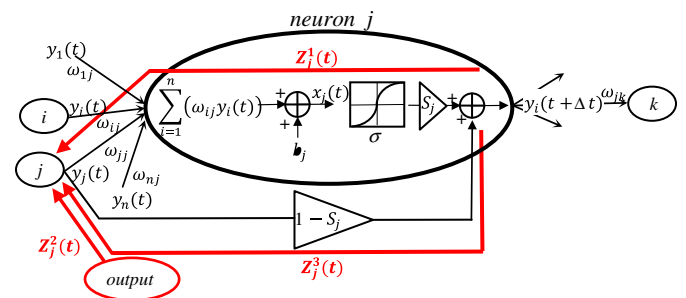


Fig. 4. Ways followed by the three elements of the backpropagated cost through a neuron.

The integral, present in the total cost expressed by Eq. (9), which made it possible to generate an optimal controller, is consequently distributed over time and over the output neurons. Nevertheless, the calculation of the components of the gradient, reintroduces this integrating aspect, thus guaranteeing the convergence towards an optimal solution over time. This explains the presence of the multiplying term, Δt , in the expression $Z_j(\tau)$ that takes into account the temporal integral of the errors coming from outside.

4. ROBIAN biped

The learning approach is validated on the ROBIAN biped prototype (Konno, Sellaouti, Amar, & Oueddou, 2002). ROBIAN consists of two different parts: a locomotion system (lower limbs, i.e., legs) and a special torso mechanism (upper part) that is described in the next section. Each leg has a total of seven dofs (Sellaouti & Oueddou, 2005): three actuated dofs for the hip, one actuated for the knee, two actuated for the ankle and one passive for the foot, providing a flexible foot system (see Fig. 5).

4.1. A simplified torso for balance control

The human upper body can be modeled with 13 degrees of freedom, which is not an easy to mechanism control. In order to identify a minimal mechanism of the human upper body that can mimic it when walking, a previous analysis of the six wrench components exerted by the upper part of a virtual manikin on the locomotion apparatus was conducted. It led to the identification of two coupling relations between the upper and lower body. Then, the dynamic equivalence between mechanisms allowed us to identify a trunk mechanism with four degrees of freedom that is able to reproduce the dynamic effects of the upper limbs during the walking gait (to understand in detail the advantage of using a simplified torso for balance control, see Mohammed et al. 2004).

ROBIAN's trunk (see Figs. 6 and 7) is an original mechanism with 4 dofs: one rotational (R) and three translational (P) movements. It is important to note that the three elements, C_2 , C_3 and C_4 , are directly connected to element C_1 by prismatic joints (P) and the whole system is connected by a rotational joint (R) to element C_0 .

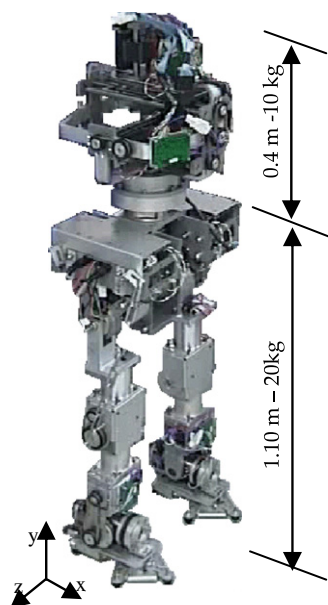


Fig. 5. The ROBIAN biped: total height is 1.50 m and weight is 30 kg.

The sagittal plane of the trunk R3P is (O_0, X_0, Y_0) , and the frontal plane is (O_0, Y_0, Z_0) . The properties of the R3P mechanism are presented in Table 1. The C_i ($i=1, \dots, 4$) elements represent the mobile masses that are used to compensate for the external disturbances.

The simulation results obtained in Mohamed, Gravez, Bruneau, and Oueddou (2001) demonstrate the existence of coupling relations between torques and forces exerted by the legs on the trunk during walking

$$M_x = k_1 F_z \text{ and } M_z = k_2 F_x \quad (17)$$

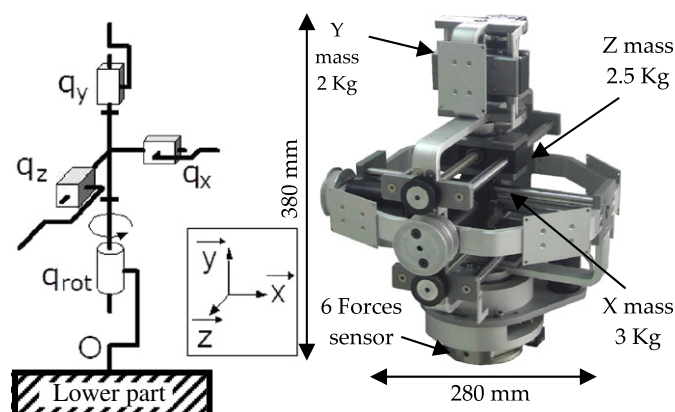


Fig. 6. The ROBIAN trunk is an RPPP mechanism. On the photo, one can see Y and Z masses that follow their prismatic joint (the X mass is hidden). Total weight is 13 kg.

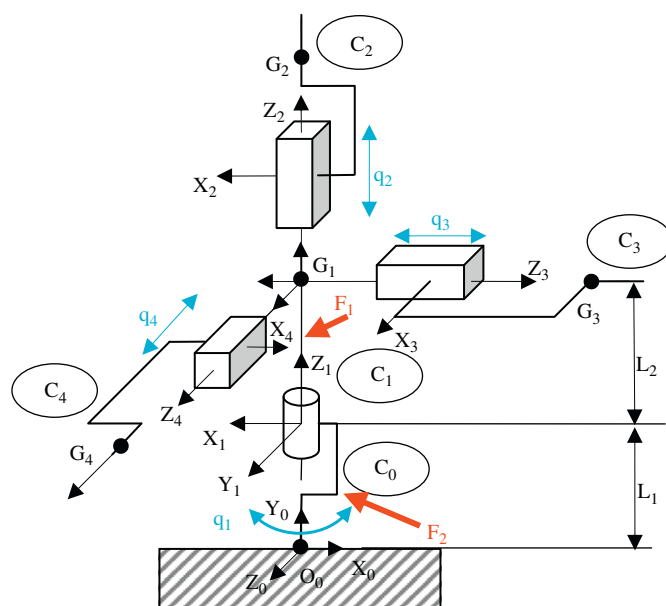


Fig. 7. Modeling of trunk R3P subjected to external forces, F1 and F2.

Table 1
Mass and inertia properties of the robian torso.

C_0	m_0 (kg)	0.5
C_1	m_1 (kg)	5
C_2	m_2 (kg)	2
C_3	m_3 (kg)	3
C_4	m_4 (kg)	2.5
Total inertia	I_y (kg m²)	0.28
L_1	(m)	0.1
L_2	(m)	0.2

where F_x , and F_z are the components of the measured efforts exerted by the support to the trunk at point O_0 .

The efforts measured to the trunk by the sensor on point O_0 are given by (see Zaoui, Bruneau, Oueddou, & Maalej, accepted for publication for more details)

$$\begin{cases} \vec{F}_0 = \sum_{i=0}^4 m_i(\vec{\gamma}_{C_i} - \vec{g}) - \sum_{i=1}^2 \vec{F}_i \\ \vec{M}_0 = \sum_{i=0}^4 (\vec{h}(i, O_0) - \vec{M}(m_i \vec{g}, O_0)) - \sum_{i=1}^2 \vec{M}(\vec{F}_i, O_0) \end{cases} \quad (18)$$

where

- $\vec{h}(i, O_0)$: Derivation of element C_i angular momentum calculated at point O_0
- $\{\vec{F}_0/\vec{M}_0\}$: Efforts exerted by the frame on C_0 at point O_0
- \vec{F}_1 and \vec{F}_2 : External forces applied, respectively, to C_1 and C_0 (C_1 and C_0 are the elements in contact with the external environment of the robot) with $F_1 = [F_{1X}, F_{1Y}, F_{1Z}]^T$ and $F_2 = [F_{2X}, F_{2Y}, F_{2Z}]^T$
- $\vec{\gamma}_{C_i}$: Acceleration of C_i elements.

Note that if body C_0 is fixed, then $\vec{\gamma}_{C_0} = \vec{0}$ and $\vec{h}(0/O_0) = \vec{0}$.

Furthermore, because $G_0 = O_0$: $\vec{M}(m_0 \vec{g}, O_0) = \vec{0}$.

These dynamic equations show that the movements of the three masses can compensate for unknown external forces applied to the trunk. Thus, in using BPTT to learn this compensation, the cost function and the gradient components for the learning stage will use these equations.

5. Real time implementation

The real time implementation of the neural controller is a necessary objective to conduct an efficient learning control. The general architecture of the robotic application is based on the exchange of information between different systems: a PC user, an industrial electronic control unit and the ROBIAN robot (Fig. 8).

The electronic unit is built with one server board (SH3 RISC, 96 MHz, 16 Mb of RAM, 128 kb of flash memory, 32 kb of Shared memory) and 8 client boards (SH3 RISC, 96 MHz, 8 Mb for RAM, 128 kb for the flash memory). These 8 boards are dedicated to the control of the 16 DC motors and to the measurement acquisition (2 DC motors per board). The user computer is dedicated to supervising the robot. It communicates with the server board via an Ethernet protocol, which exchanges data with the eight client boards via a special local area network. These client boards exchange with the robot through point-to-point wiring. Programs

are run on the server board and are built around the use of objects representing the actuators, the sensors of the robot and their parameters. These objects are written in the shared memory of the electronic unit. The control of each motor is regulated locally via the control boards.

5.1. Multitask organization

The neural networks algorithms, written in the C language, are implemented in the server board. A user interface, written in the Visual Basic graphic language, gives access to all the parameters of the algorithms, to the measurements taken on the robot and to the neural network parameters. The entire application is divided into five specific tasks, as shown in Fig. 8: the “Main” task, which focuses on most of the information exchanged and manages the objects of the shared memory, the “Measure” task, the “Transfer” task, the “Propagation” task and the “Back-propagation” task.

5.2. Scheduling of learning tasks

5.2.1. Temporal evaluation

Other experiments have shown that the fastest oscillation period that can be applied to the biped robot in its frontal plan is 500 ms. Thus the computing time required by the ROBIAN control must be limited to 10 ms in order to maintain sufficient control.

However, the mathematical complexities (number of implied operations) associated with the propagation functions and the training of the neural controller are $O(n^2)$ and $O(W_T n^2)$, respectively, with n being the number of neurons and W_T , the width of the time window used. Consequently, if the two functions are computed sequentially, then the number of operations will evolve out of $O(n^2 + W_T n^2)$, and the associated computing time will be able to exceed the limit required. For this reason, the propagation and the backpropagation are separated into two dependent and synchronized tasks. Then, according to the ratio of the computing time (related to the width of the time window) of each task for one training, several propagations in the network are carried out.

Table 2 compares the time performance of propagation and backpropagation. The sampling of the propagation task depends on the computing time necessary for the propagation in the network and, consequently, on the number of parameters defined with the network structure. For example, the propagation time is 2 ms through a completely recurring network consisting of 10 neurons (and 3 ms for 14 neurons and time window of 10). The backpropagation, which is computed along the time window, has an execution time that depends on the width of the time window. For a 10 neuron network with a time window memorizing the 20 preceding states, the back-propagation task will require 50 ms (60 ms for 14 neurons and a time window of 10). Table 3.

The algorithmic complexity of the learning tasks is calculated involved in the training, where n is the number of neurons in the net, W_T the Time Window width (number of net states stored). Fourteen neurons and $W_T = 10$ are setup values for the experiments described in Section 5.

Thus, even taking into account the commutation time between the tasks, these two sampling periods remain approximately equal to 2 and 50 ms (see next subsection). Fig. 9 is a snapshot taken by the

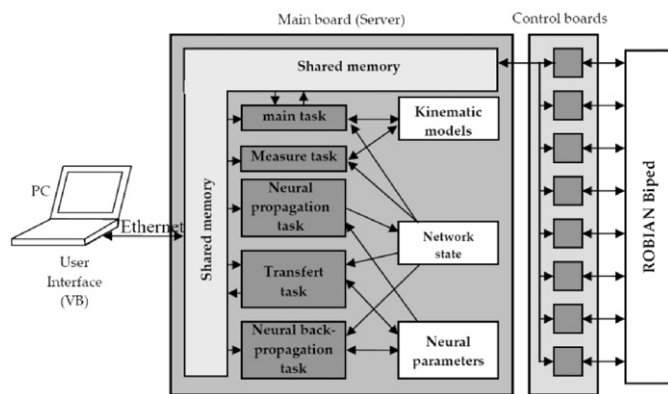


Fig. 8. Control architecture of the ROBIAN platform.

Table 2
Time performances of propagation and backpropagation.

	Propagation	Backpropagation
Complexity	$O(n^2)$	$O(TW \cdot n^2)$
Computational time ($n=10, W_T=20$)	2 ms	50 ms
Computational time ($n=14, W_T=10$)	3 ms	60 ms

numerical oscilloscope of the user interface, which is connected to the electronic control unit. It shows the dependence and synchronization between the propagation and back-propagation tasks.

The learning procedure based on the BPTT algorithm uses neuron outputs to calculate backpropagated costs and gradients. However, these same values are modified by the propagation procedure carried out in parallel. It is then not possible to use the same set of variables for the two tasks. Thus, two different tables have been set up in the neuronal algorithms to memorize the states of the network.

5.2.2. Synchronization

For each new iteration of learning, it is necessary to transfer the states from the network generated by the propagation task to the learning task. This procedure, which consists of copying one table into another, is placed upstream of the back-propagation. It is called “temporal photograph” because it memorizes the current instantaneous state of the network. Thus, this table contains the previous states of W_T that are present in the temporal window.

Table 3
Numerical values of learning algorithm parameters.

Parameter	Name	Initial value
Number of neurons	n	14
Initial weights	ω	Randomly chosen in range of $[-5;5]$
Width of time window	W_T	10 steps
Time constant values	T	Randomly chosen in range $[0 \text{ s}; 1 \text{ s}]$
Learning step	η	0.001 s

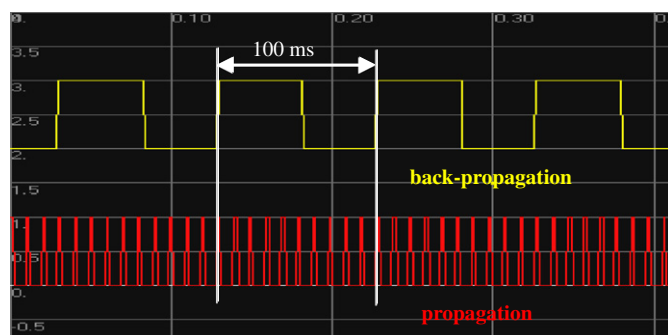


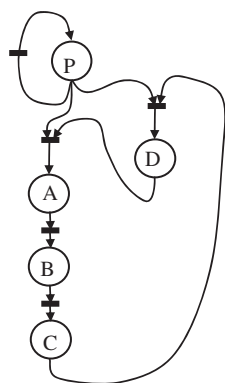
Fig. 9. Visualization of activations of propagation and backpropagation tasks computed in real time by the 10 neurons and there is one backpropagation for 10 propagation step states.

The learning procedure gathers four principal stages: temporal photograph (A), backpropagated costs calculation (B), gradient components computation (C), and parameter modification (D). Stages (B) and (D) are based on the values memorized by the temporal photograph and do not interact with those used by the propagation. On the other hand, stages (A) and (D) copy the network states and modify the parameters. Therefore, their executions are directly related to that of the propagation, as shown in Fig. 9. The stage (A) must start only when the propagation (P) is finished. Indeed, it is not necessary for the values that make up a state of the network to be modified when copying this state. To avoid this problem, stage (A) initially checks that the propagation is not running. If the propagation is running, then stage (A) waits until it ends before carrying out the copy. The end of the propagation is announced by a binary semaphore, with a value of 0 (propagation stopped) or 1 (propagation in progress).

In the same way, stage (D) must be carried out when the propagation is stopped. Indeed, this stage modifies the parameters of the network. Thus, it is essential that these parameters are not used when their values are updated. Therefore, this stage (D) waits until the next propagation is finished by observing the semaphore value. If the propagation is not running when launching task (D), it will wait until the propagation launches (propagation in progress) and then finishes (propagation stopped). This will ensure that the propagation will not start during the modification of the parameters. The computation times for this example are given by the table in Fig. 10. The duration does not include the waiting periods due to the synchronization process.

Fig. 11 details the execution times of the propagation task, and the other tasks necessary for the learning process. The duration of the induced synchronization waits are also included. In this example, the network consists of 10 neurons and the temporal window has a width of 20 memorized states. The propagation period task is 10 ms, whilst that of the learning task is 100 ms, and the sampling rate of the measurement is 1 ms. Stage (A') corresponds to the 3 ms wait required before the beginning of the temporal photograph (A). The stage (D') represents waiting for 5 ms before modifying the network parameters. Note that this stage waits until the propagation starts and finishes before allowing the modification of the parameters (D).

These details clearly show the loop-time for the control. The time necessary to refresh only the commands of the DC motors, depending on the propagation, is about 2 ms. Taking into account the learning aspects, the loop-time for the control (i.e., the time required to refresh the commands of the DC motors and the network controller) is 53 ms. Hence, it is sufficiently compared to the whole dynamics of the robot, which has an equivalent time constant of 1.5 s. Thus, the loop time does not degrade the robot system and the performance of the control laws.



Task	Duration
Propagation (P)	2 ms
temporal Photograph (A)	4 ms
Z_j (τ)	19 ms
gradient Computation (C)	21 ms
Parameters Modification (D)	1 ms

Fig. 10. On the left, description of synchronization between the propagation task and the backpropagation procedures. On the right, computation times of the different neural procedures for a 10-neuron network.

6. Experimental results

6.1. Settings for experiments

For this experiment, a 6–5–3 network is used: 6 inputs, 5 hidden neurons and 3 outputs (14 neurons). The number of hidden neurons was selected after some testing. At the beginning of the learning, the weight values of the net are randomly chosen. The time constants are selected from a wide enough range that the net behavior is stable, i.e., that there is no oscillation of the control.

6.2. Learning the balance control with dynamic BPTT

For this approach to the robot's equilibrium control, the experiment is only based on the measured wrench between the upper part and the locomotion system given by the 6-component force sensor. The neural torso controller aims to keep these forces and moments close to zero for the X and Z axes and to weight the value for the Y-axis, even when an external perturbation is applied to the torso. The controller only tries to compensate for external perturbations generated manually on the torso to emulate the

coupling effects in a walking gait. The experiment consists of following stages:

- ROBIAN stays in the standing posture with all its joints controlled.
- Two successive stages are carried out: learning for 60 s and then steering.
- During the learning stage (successive short periods), unknown external forces are manually applied to the torso in X and Z directions (for more comprehension, see Figs. 12 and 13).
- The rotational motion of the torso is locked for this experiment.
- The X and Z moment components are not required for the net computation because their effects can be deduced from the component forces due to the coupling relations (Eq. (16)). Moreover, as the rotational motion is locked, M_y data are also unnecessary.

The learning stage consists of applying forces to each axis, successively. Indeed, each mass is able to compensate only for the forces that appear on its axis. The amplitudes of the applied perturbation and their periods are voluntarily variable in order to show the network different kinds of situations. The values of these external forces are in the maximal range corresponding to the limit of the biped toppling.

6.3. On-line learning of disturbance compensation

The aim of the learning stage is to learn to compensate for unknown external disturbances. The time evolutions of responses during the learning stage are depicted in Figs. 14, 15, and 16. The total duration of the learning stage is 120 s. This is the time required by the neural controller to learn the decoupling between the axes when external forces are applied to the ROBIAN torso. The learning algorithm starts at $t=4.5$ s. Until $t=11.4$ s, no external forces are applied, so the measured forces are close to zero (F_y close to the ROBIAN torso weight value), and the net does not modify its parameters. Next, during the X-axis perturbation period, the net modifies its X output command to generate a stronger and faster response to the external forces until it becomes insensitive to the disturbance (around $t=20$ s). Then (from $t=18$ to 22 s), the Z-axis becomes sensitive to the X-perturbation, and the neural controller explores its control space and learns the decoupling between the two axes. Next (after $t=34$ s), on each axis, the mass has to move in the same direction as the force.

Next, from $t=42$ (Fig. 14) to $t=63$ s (Fig. 15), it progressively modifies this coupling by decreasing the influence of the X force on its Z control law. From $t=62$ s, it is clear from Fig. 15 that the network can compensate for each pulse of perturbation due to lateral (Z-axis) or frontal (X-axis) pushes against the robot. After this convergence

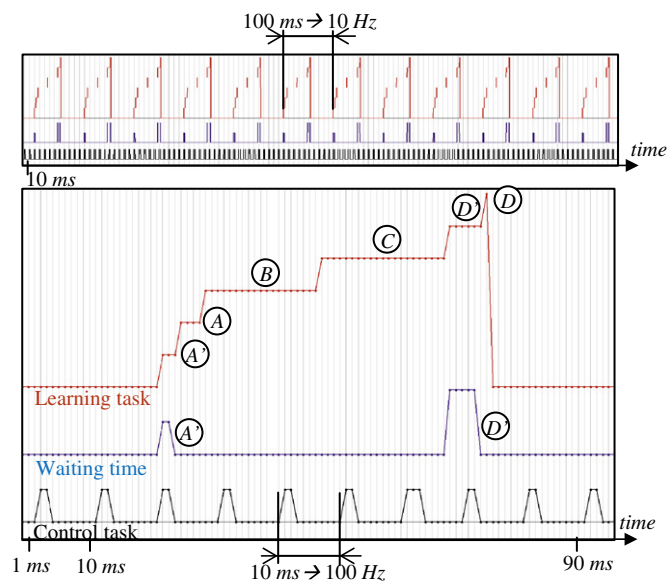


Fig. 11. Synchronization of propagation and learning extracted from the user interface: general view (up) and zoom (bottom) for the 10-neuron network. The duration of the propagation task is 3 ms and its period is 10 ms. The total duration of the backpropagation task is 53 ms and its period is 100 ms.

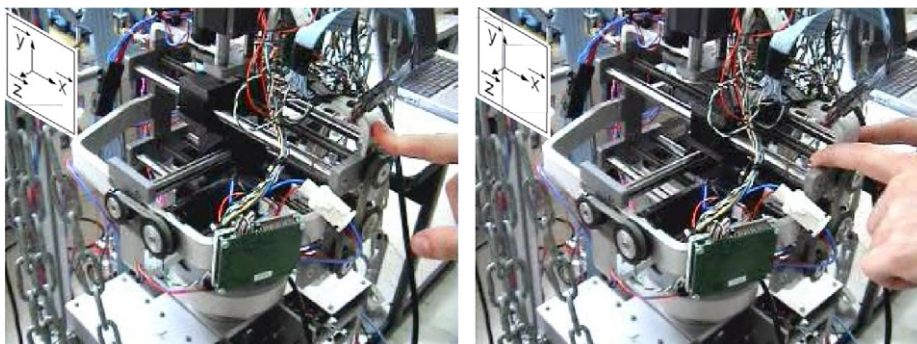


Fig. 12. External forces are applied manually during successive short moments following the X direction here (left: pushing; right pulling).

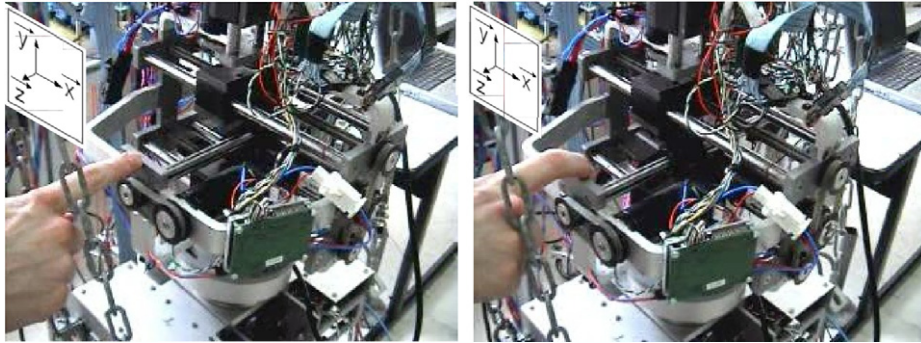


Fig. 13. External forces are applied manually during successive short moments following the Z direction here (left: pushing, right: pulling).

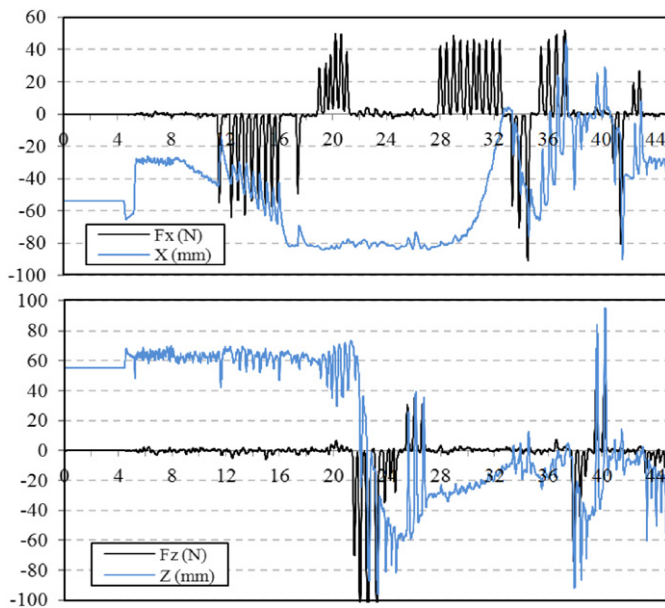


Fig. 14. Beginning of on-line learning for the X-axis and Z-axis during time (s). The learning algorithm starts at $t=4.5$ s. The neural controller tries to learn the decoupling between the two axes.

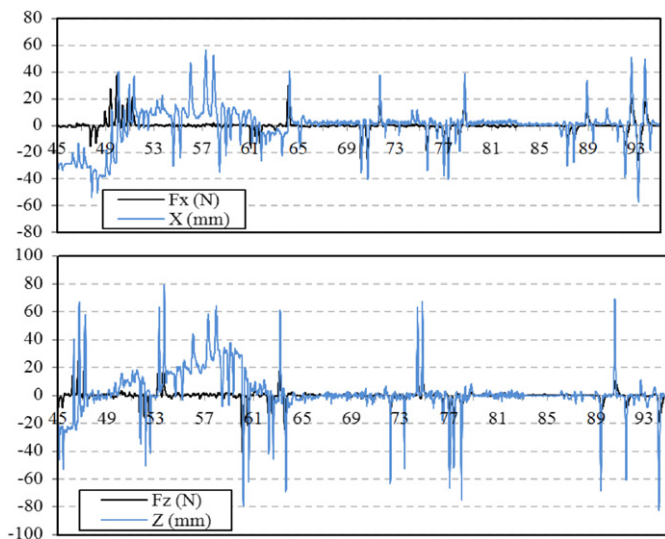


Fig. 15. Middle of on-line learning for the X-axis and Z-axis during time (s). The neural controller learns the decoupling between the two axes, and then compensates each perturbation.

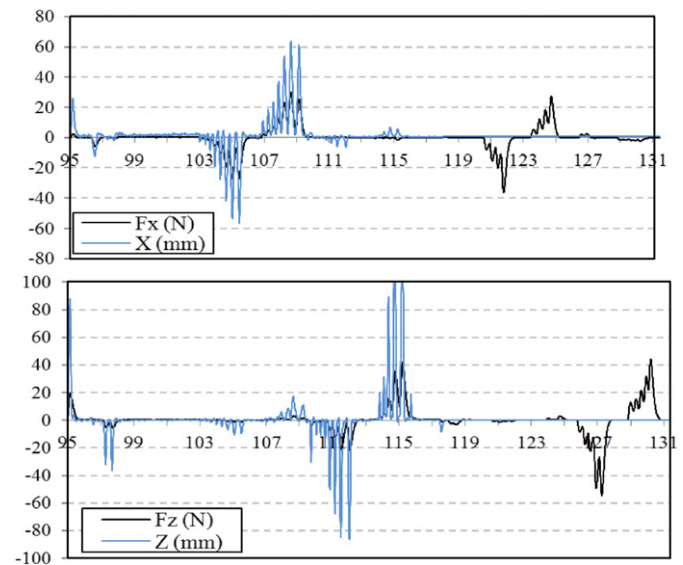


Fig. 16. End of on-line learning for the X-axis and Z-axis during time (s). Until $t=120$ s, the neural controller learned the decoupling between the two axis. After, the learning algorithm stops at $t=120$ s, when a perturbation occurs on an axis, it is not compensate.

period, the learning continued for about 40 s. During that time, the previous changes were carried out (with forces in the opposite direction and more switches between X and Z external perturbation forces). At the end, the network emulated a controller that was able to compensate for a portion of the external forces applied (see Fig. 16). The learning algorithm stops at $t=120$ s; when a perturbation occurs on an axis, it does not compensate. As far as the Y-axis is concerned, the learning process manages to find motions to compensate for external Y forces in the same way as it does for the two other axes. There was no coupling between this axis and the others.

6.4. External disturbance compensation after learning

Once the learning stage is finished, the net's parameters (biases, weights and time constants) are fixed and stored. The steering consists on using them without adding any new modification.

During the neural control period, the same forces are applied, but the 6-force sensor also measures the influence of the motion of the mass. The difference between the force shapes in the two periods represents the compensation brought about by the mass motion. In a previous study (Scesa, Mohamed, Henaiff, & Oueddou, 2005), it was shown that neural control brings a decrease of about 50% in the measured forces on each axis.

7. Conclusion

This paper addresses the real time implementation of on-line learning control with recurrent neural networks. The aim of the control is to compensate for external perturbations due to lateral or frontal pushes against the robot. An experiment is carried out to control the balance of a biped robot in its standing posture. CTRNN neural nets and BPTT learning have been used to optimize a cost function which explains the rejection of perturbations. The algorithm is embedded in the real time electronic unit of the robot, and the on-line learning implementation is very detailed. Preliminary results on the learning behavior and the control performances are presented. The implementation enables to focus on the real performance of the neural network controller algorithms and shows that CTRNN are able to learn the balance reflex on-line, when controlling the robot. These experimental results show the strength and efficiency of the proposed approach.

Acknowledgment

The authors would like to thank the French BIA Company, which provides the real time electronic unit that controls ROBIAN and funded the Ph.D. thesis of V. Scesa via a CIFRE contract.

Appendix A. Details of gradient components calculation

The backpropagated cost attached to the j th neuron is defined by

$$Z_j(\tau) = \frac{\partial E}{\partial y_j(\tau)} \quad (A1)$$

and Z_k the back-propagated cost attached to the neuron downstream k , then each components of the gradient can be written as

• Gradients on weights

$$\begin{aligned} \frac{\partial E}{\partial \omega_{jk}} &= \int_{t_0}^t \left[\frac{\partial E}{\partial \omega_{jk}}(\tau) \right] d\tau = \int_{t_0}^t \left[\frac{\partial E}{\partial y_k(\tau)} \frac{\partial y_k(\tau)}{\partial \omega_{jk}(\tau)} \right] d\tau \\ &= \int_{t_0}^t \left[Z_k(\tau) \frac{\partial y_k(\tau)}{\partial \omega_{jk}(\tau)} \right] d\tau = \int_{t_0}^t \left[Z_k(\tau) \frac{\partial y_k(\tau)}{\partial x_k(\tau)} \frac{\partial x_k(\tau)}{\partial \omega_{jk}(\tau)} \right] d\tau \\ &= S_k \int_{t_0}^t [Z_k(\tau) f'(x_k(\tau - \Delta t)) y_j(\tau - \Delta t)] d\tau \end{aligned} \quad (A2)$$

f' is derivative from the sigmoid activation function

• Gradients on bias

$$\begin{aligned} \frac{\partial E}{\partial b_j} &= \int_{t_0}^t \left[\frac{\partial E}{\partial b_j}(\tau) \right] d\tau = \int_{t_0}^t \left[\frac{\partial E}{\partial y_j(\tau)} \frac{\partial y_j(\tau)}{\partial b_j(\tau)} \right] d\tau = \int_{t_0}^t [Z_j(\tau) \frac{\partial y_j(\tau)}{\partial b_j(\tau)}] d\tau \\ &= \int_{t_0}^t \left[Z_j(\tau) \frac{\partial y_j(\tau)}{\partial x_j(\tau)} \frac{\partial x_j(\tau)}{\partial b_j(\tau)} \right] d\tau = S_j \int_{t_0}^t [Z_j(\tau) f'(x_j(\tau - \Delta t))] d\tau \end{aligned} \quad (A3)$$

• Gradients on scale parameters

$$\begin{aligned} \frac{\partial E}{\partial T_j} &= \int_{t_0}^t \left[\frac{\partial E}{\partial T_j}(\tau) \right] d\tau = \int_{t_0}^t \left[\frac{\partial E}{\partial y_j(\tau)} \frac{\partial y_j(\tau)}{\partial T_j(\tau)} \right] d\tau = \int_{t_0}^t [Z_j(\tau) \frac{\partial y_j(\tau)}{\partial T_j(\tau)}] d\tau \\ &= \frac{-S_j^2}{\Delta t} \int_{t_0}^t [Z_j(\tau) (f(x_j(\tau - \Delta t)) - y_j(\tau - \Delta t))] d\tau \end{aligned} \quad (A4)$$

Appendix B. Backpropagation in a dynamic neuron.

• Calculation of $Z_j^1(\tau)$:

$Z_j^1(\tau)$ corresponds to the error coming, at the following instants, from the synapses connecting the neurons

$$Z_j^1(\tau) = \sum_{k=1}^n \frac{\partial C}{\partial y_k(\tau + \Delta t)} \frac{\partial y_k(\tau + \Delta t)}{\partial x_k(\tau)} \frac{\partial x_k(\tau)}{\partial y_j(\tau)}$$

$$= \sum_{k=1}^n [Z_k(\tau + \Delta t) S_k f'(x_k(\tau)) \omega_{jk}(\tau)] \quad (B1)$$

• Calculation of $Z_j^2(\tau)$:

$Z_j^2(\tau)$ represents the error coming directly from the output neurons. It does not exist if the neuron is hidden. Hence, for the output neurons only, there is $e_j(\tau) \neq 0$:

- for a hidden neuron: $Z_j^2(\tau) = 0$
- for an output neuron $Z_j^2(\tau) \neq 0$:

The calculation of the instantaneous errors $e_j(\tau)$ constitutes the coupling between the controlled system and the learning algorithm. This calculation depends on the criterion to minimize. Therefore, it is not possible to write a general formula to obtain these values because they are related to the controlled system and to the cost function

$$Z_j^2(\tau) = \frac{\partial E}{\partial y_j(\tau)} = \frac{\partial [\sum_{k=1}^n e_k(\tau)]}{\partial y_j(\tau)} \Delta t = \frac{\partial e_j(\tau)}{\partial y_j(\tau)} \Delta t \quad (B2)$$

• Calculation of $Z_j^3(\tau)$:

$Z_j^3(\tau)$ corresponds to the error coming at the following instants by means of the internal dynamics of the neuron itself

$$Z_j^3(\tau) = \frac{\partial C}{\partial y_j(\tau + \Delta t)} \frac{\partial y_j(\tau + \Delta t)}{\partial y_j(\tau)} = Z_j(\tau + \Delta t)(1 - S_j) \quad (B3)$$

The difference between $Z_j^1(\tau)$ and $Z_j^3(\tau)$, in the error expression coming from the j th neuron, comes from the terms considered in the Eq. (4) because $Z_j^3(\tau)$ takes into account the dynamic part expressed by the term, $(1 - S_j)y_j(\tau)$.

• Calculation of $Z_j(\tau)$:

Finally, by gathering the Eqs. (B1), (B2) and (B3), the following equation of the backpropagated cost is obtained

$$\begin{aligned} Z_j(\tau) &= Z_j^1(\tau) + Z_j^2(\tau) + Z_j^3(\tau) = \sum_{k=1}^n [Z_k(\tau + \Delta t) S_k f'(x_k(\tau)) \omega_{jk}(\tau)] \\ &\quad + \frac{\partial e_j(\tau)}{\partial y_j(\tau)} \Delta t + Z_j(\tau + \Delta t)(1 - S_j) \end{aligned} \quad (B4)$$

References

- Albus, J. S. (1975). A new approach to manipulator control: the cerebellar model articulation controller (CMAC). *Transactions of the ASME Series G:Journal of Dynamic Systems, Measurement and Control*, 97, 220–233.
- Beer, R. D. (2006). Parameter space structure of continuous-time recurrent neural networks. *Neural Computation*, 18, 3009–3051.
- Fukuda, T., Komata, Y., Arakawa, T. (1997). Stabilization control of biped locomotion robot based learning with GAs having self-adaptive mutation and recurrent neural networks. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pp. 217–222.
- Gao, Y., & Er, M. J. (2003). On-line adaptive fuzzy neural identification and control of a class of MIMO nonlinear systems. *IEEE Transactions on Fuzzy Systems*, 11(4), 462–477.
- Geng, T., Porr, B., & Wörgötter, F. (2006). A reflexive neural network for dynamic biped walking control. *Neural Computation*, 18(5), 1156–1196. MIT press.
- Hoinville, T., Hénaff, P. (2004). Comparative study of two homeostatic mechanisms in evolved neural controllers for legged locomotion. In *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems*, Japan (pp. 2624–2629).
- Horvath, G., & Szabo, T. (2007). Kernel CMAC with improved capability. *IEEE Transactions on Systems, Man, and Cybernetics, Part B*, 37(1), 124–138.
- Hsu, Chun-Fei (2009). Intelligent position tracking control for LCM drive using stable on-line self-constructing recurrent neural network controller with bound architecture. *Control Engineering Practice*, 17(6), 714–722.
- Hyon, S.-H. (2009). Compliant terrain adaptation for biped humanoids without measuring ground surface and contact forces. *IEEE Transactions on Robotics*, 25(1).
- Konno, A., Sellaoui, R., Amar, F.B., Ouezdou, F.B. (2002). Design and development of the biped prototype ROBIAN. In *Proceedings of IEEE International Conference on Robotics and Automation*, vol. 2, issue 11, pp.1384–1389.
- Kubica, E., Wang, D., Winter, D. (2001). Feedforward and deterministic fuzzy control of balance and posture during human gait. In *Proceedings of IEEE International Conference on Robotics & Automation (ICRA)*, Seoul, Korea (pp. 2293–2298).

- Kun, A.L., Miller, W.T. (1996). Adaptive dynamic balance of a biped robot using neural networks. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, vol. 1, pp. 240–245, Minnesota.
- Lin, Chih-Min, & Chen, Chiu-Hsiung (2007). Robust fault-tolerant control for a biped robot using a recurrent cerebellar model articulation controller. *IEEE Transactions on Systems, Man, and Cybernetics, Part B*, 37(1), 110–123.
- Marcua, T., Köppen-Seliger, B., & Stücher, R. (2008). Design of fault detection for a hydraulic looper using dynamic neural networks. *Control Engineering Practice*, 16, 192–213.
- Mizunoo, N., Kuroda, M., Okazaki, T., & Ohtsu, K. (2007). Minimum time ship maneuvering method using neural network and nonlinear model predictive compensator. *Control Engineering Practice*, 15, 757–765.
- Mohamed, B., Gravez, F., Bruneau, O., Ouezdou, F.B. (2001). Four Dof Torso dynamic effects on biped walking gait. In *Proceedings of 14th CISM-IFToMM Symposium On Theory and Practice of Robots and Manipulators, Romansy* (pp. 453–462).
- Mohammed, B., Gravez, F., & Ouezdou, F. B. (2004). Emulation of the dynamic effects of human torso during a walking gait. *Journal of mechanical design, Transactions of the ASME*, 126(5), 830–841.
- Morimoto, J., Endo, G., Nakanishi, J., Hyon, S.H., Cheng, G., Bentevegna, D., Atkeson, C.G. (2006). Modulation of simple sinusoidal patterns by a coupled oscillator model for biped walking. In *Proceedings of IEEE International Conference on Robotics and Automation, ICRA 2006*, issue 15–19, pp. 1579–1584.
- Nakanishi, J., Morimoto, J., Endo, G., Cheng, G., Schaal, S., & Kawato, M. (2004). Robot learning from demonstration. *Robotics and Autonomous Systems*, 47(2–3), 79–91.
- Nguyen, M. H., & Cottrell, G. W. (1997). Tau net: a neural network for modeling temporal variability. *Neurocomputing*, 15, 249–271.
- Pearlmutter, B. A. (1995). Gradient calculation for dynamic recurrent neural networks: a survey. *IEEE Transactions on Neural Networks*, 6(5), 1212–1228.
- Pineda, F. J. (1987). Generalization of back-propagation to recurrent neural networks. *Physical Review Letters*, 59(19), 2229–2232.
- Righetti, L., Ijspeert, A.J. (2006). Programmable central pattern generators: an application to biped locomotion control. In *Proceedings of the IEEE International Conference on Robotics and Automation, Orlando* (pp. 1585–1590).
- Robinson, A. J., & Fallside, F. (1987). Static and dynamic error propagation networks with application to speech coding. *Proceedings of Neural Information Processing Systems*, 632–641 American Institute of Physics.
- Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986). Learning internal representations by error propagation. In D. E. Rumelhart, J. L. McClelland, & the PDP Research Group (Eds.), *Parallel Distributed Processing: Explorations in the Microstructure of Cognition* (pp. 318–362). Cambridge MA: MIT Press.
- Sabourin, C., & Bruneau, O. (2005). Robustness of the dynamic walk of a biped robot subjected to disturbing external forces by using CMAC neural networks. *Journal of Robotics and Autonomous Systems*, 51, 81–99. Elsevier Science.
- Scesa, V., Hénaff, P., Ouezdou, F. B., & Namoun, F. (2006). Time window width influence on dynamic BPTT(h) learning algorithm performances: experimental study. In S. Kollias (Ed.), *Proceedings of IEEE International Conference on Artificial Neural Networks* (pp. 93–102). Berlin Heidelberg: Springer-Verlag. Part I, LNCS 4131.
- Scesa, V., Mohamed, B., Hénaff, P., Ouezdou, F.B. (2005). Dynamic recurrent neural network for biped robot equilibrium control: preliminary results. In *Proceedings of IEEE International Conference on Robotics and Automation* (pp. 4125–4130).
- Sellaouti, R., & Ouezdou, F. B. (2005). Design and control of a 3DOFs parallel actuated mechanism for biped application. *Mechanism and Machine Theory*, 40(12), 1367–1393.
- Setiawan, S. A., Hyon, S. H., Yamaguchi, J., Takanishi, A. (1999). Physical interaction between human and a bipedal humanoid robot: Realization of human-follow walking. In *Proceedings of IEEE International Conference on Robotics & Automation (ICRA)*, Detroit, U.S.A. (pp. 361–367).
- Song, E-J, & Tahk, M-J. (2001). Real-time neural-network midcourse guidance. *Control Engineering Practice*, 9, 1145–1154.
- Su, Shun-Feng, Lee, Zne-Jung, & Wang, Yan-Ping (2006). Robust and fast learning for fuzzy cerebellar model articulation controllers. *IEEE Transactions on Systems, Man, and Cybernetics, Part B*, 36(1), 203–208.
- Taga, G., Yamaguchi, Y., & Shimizu, H. (1991). Self-organized control of bipedal locomotion by neural oscillators in unpredictable environment. *Biological Cybernetics*, 65(3), 147–159.
- Vukobratovic, M. (2004). Zero-moment-point—Thirty Five years of its life. *International Journal of Humanoid Robotics*, 1(1), 157–173.
- Werbos, P. J. (1990). Backpropagation through time: what it does and how to do it. *Proceedings of the IEEE*, 78(10), 1550–1560. October.
- Williams, H. J. (2007). Homeostatic plasticity improves signal propagation in continuous-time recurrent neural networks. *Biosystems*, 87(2–3), 252–259.
- Williams, R. J., & Peng, J. (1990). An efficient gradient-based algorithm for on-line training of recurrent network trajectories. *Neural Computation*, 4, 491–501.
- Williams, R. J., & Zipser, D. (1989). A learning algorithm for continually running fully recurrent neural networks. *Neural Computation*, 1270–1280.
- Wu, Y., Song, Q., & Yang, X. (2007). Robust recurrent Neural Network Control of Biped Robot. *Journal of Intelligent Robotic Systems*, 49, 151–169.
- Zaoui, C., Bruneau, O., Ouezdou, F.B., Maalej, A. (2009). Simulations of the dynamic behavior of a bipedal robot with trunk and arms subjected to 3D external disturbances in a vertical posture, during walking and during object handling, *Journal of Multibody System Dynamics*, 21 (3), 261–280.

9.2 On the analysis of sigmoidal time parameters for dynamic truncated BPTT algorithm. IEEE International Joint Conference on Neural Networks, IJCNN 2006

On The Analysis Of Sigmoid Time Parameters For Dynamic Truncated BPTT Algorithm

V.Scesa, P.Henaff, F.B.Ouezdou, and F.Namoun

Abstract—The purpose of the research addressed in this paper concerns a comparative study of two expressions of the time scale parameter for Continuous Time Recurrent Neural Network (CTRNN): a classical time constant expression, and a sigmoid one. Their influence on the stability, the convergence speed and the generalization ability of a BackPropagation Through Time (BPTT) learning algorithm, will be discussed. Firstly, three mathematical conclusions related to the propagation and learning equations are deduced. Then these conclusions are validated on experiments carried out on a real biped robot. Through the identification of the balancing behavior under different robot torso motions, the sigmoid expression will be shown to get the best learning results.

I. INTRODUCTION

THE supporting project of this study was born from the collaboration between a robotic research laboratory (LIRIS) and an industrial company (BIA). This project aims at shaping a smart architecture able to learn how to control non linear multi actuators system including real time constraints. Through the design and the implementation of this controller, we want to evaluate the ability of neural architectures and learning algorithms to achieve non linear control needs in real time. The project also aims at analyzing the capability of such algorithms to be adapted for a particular system, by rising optimal control, but also to be adaptive to changes in its environment, by generalizing the knowledge learned. The experiments will be carried out on two different structures: the ROBIAN biped robot from the LIRIS [1], and the BIA 6 hydraulic axis road simulator. In this paper, the experiments are limited to the ROBIAN robot.

The architecture of the developed controller is based on recurrent neural network with leaky integrator neurons. This allows the focusing of the net dynamics on time constant parameters. Recurrent connections are used for merging the neurons responses in a function able to contain oscillating motions. In order to shape this architecture, a BPTT based algorithm was chosen for its ability to integrate the learning error [2], [3], [4], [5]. The gradient algorithms are mainly used for system modeling, optimization applications, speech

recognition [6], or meta learning [7].

The using of recurrent networks with gradient based learning algorithm is not obvious and may be unstable. As time constants modifications have a strong influence over network dynamics [8], it can intensify or eliminate this problem, and the way they are modified needs a specific attention. However, among all the studies based on CTRNNs, only a few are focused on the dynamic analysis of the net parameters [9], [10].

The aim of this paper is to study the influence of the expression of the time parameter over the algorithm stability, on the learning performances and on the evolution of the net dynamics. In a first part, the neural model and the learning algorithm will be detailed and the two time scale parameters expressions will be given. Then, their comparison will be performed, in a mathematic way by focusing on the propagation and learning equations, and on experimental results. An experiment carried out on the identification of the ROBIAN robot balancing will show the performance of the two expressions.

II. NEURAL MODEL AND LEARNING ALGORITHM

A. Neurons and propagation

Following the leaky integrator equation (1), the input data is propagated in the network to generate the neurons outputs. The result belongs to the intrinsic parameters of the neurons and the network: weights, biases and time parameters. Classically, the propagation is written as follows:

$$T_j \cdot \frac{\partial y_j}{\partial t} = -y_j + f\left(\sum_i [w_{ij} \cdot y_i] + b_j\right) \quad (1)$$

The corresponding discrete expressions are :

$$x_j(t) = \sum_i [w_{ij} \cdot y_i(t - \Delta t)] + b_j \quad (2)$$

$$y_j(t) = \frac{\Delta t}{T_j} \cdot f(x_j(t)) + \left(1 - \frac{\Delta t}{T_j}\right) \cdot y_j(t - \Delta t) \quad (3)$$

Where y_j corresponds to the j neuron activity, f is the activation function (\tanh) and Δt is the time step. The net parameters w_{ij} , b_j and T_j are respectively the weights, biases and time constants.

In (3), the $\Delta t/T_j$ term is a scale parameter. Its value, given within $]0;1[$, expresses the response speed of the j neuron. This expression is not the only one possible to define

V.Scesa is with the LIRIS laboratory, UVSQ (Versailles university) CNRS, 10,12 av. de l'Europe 78140 Velizy France (phone: (33)139254957, fax: (33)139254985, e-mail: vincent.scesa@liris.uvsq.fr)

P.Henaff is with the LIRIS-UVSQ-CNRS laboratory (e-mail: patrick.henaff@liris.uvsq.fr)

F.B.Ouezdou is with the LIRIS-UVSQ-CNRS laboratory (e-mail: ouezdou@liris.uvsq.fr)

Faycal Namoun is with the BIA company, 8 rue de l'Hautil 78700 Conflans France (e-mail: f.namoun@bia.fr)

the neuron dynamics. In [6] and [11], a sigmoid version of the time scale parameter is proposed. Through this change, the aim is to modify the way the neuron speed varies.

By defining a variable S_j that will be called **time scale parameter**, (3) can be written:

$$y_j(t) = S_j \cdot f(x_j(t)) + (1 - S_j) \cdot y_j(t - \Delta t) \quad (4)$$

In this paper, we will compare the two expressions of the time scale parameter:

$$S_j^l = \frac{\Delta t}{T_j} \quad \text{vs} \quad S_j^s = \frac{1}{1 + e^{-\lambda_j}} \quad (5)$$

The first one uses the classical **time constant** T_j . The second one uses another parameter called the **sigmoid time parameter** λ_j .

The λ_j parameter has no physical meaning, and it has nothing to do with the neuron sigmoid activation function. It expresses the neuron speed in another way, allowing a smoother logarithmic modification of S_j rather than the classical linear one.

B. Learning process

The objective consists in modifying the parameters of the network (weights, biases and time scale parameters) in order to minimize a desired criterion. In a control application, this criterion would be the gap between the desired state of the system and the actual one. In an identification process, the criterion would be the error between the neural model and the taught system. In order to carry out this net parameters adaptation while the system is running, BackPropagation Through Time algorithm, detailed in [3], [4] and [5], is used.

At first, the algorithm computes an error function that corresponds to the criterion to be minimized. This function is the integral of the error expressed as follows:

$$E = \int_{t_0}^t [e_j(\tau) \cdot d\tau] \quad (6)$$

Where E represents the criterion and is equal to the error integral between t_0 and the current time step t. $e_j(\tau)$ is the output error of the neuron j stored at time τ , and t_0 is the beginning of the integration window.

BPTT algorithm carries out a gradient based learning. Thus, the modification of the parameters is following the opposite value of the error gradient of each parameter:

$$\Delta w_{jk} = -\eta_1 \cdot \frac{\partial E}{\partial w_{jk}} \quad (7)$$

$$\Delta b_j = -\eta_2 \cdot \frac{\partial E}{\partial b_j} \quad (8)$$

$$\text{and: } \Delta T_j = -\eta_3 \cdot \frac{\partial E}{\partial T_j} \quad \text{or} \quad \Delta \lambda_j = -\eta_3 \cdot \frac{\partial E}{\partial \lambda_j} \quad (9)$$

The learning speed is defined by the three different

learning rates η_1 for the weights, η_2 for the biases and η_3 for the time parameters.

The use of continuous time neurons implies a dynamic BPTT learning [2]. Hence the gradient values giving the influences of each parameter on the error are derived from gradient equations (11) to (14) written with the backpropagated error (10) :

$$z_j(\tau) = \frac{\partial E}{\partial y_j(\tau)} \quad (10)$$

$$\frac{\partial E}{\partial w_{jk}} = \int_{t_0}^t [S_k \cdot z_k(\tau) \cdot f'(x_k(\tau - \Delta t)) \cdot y_j(\tau - \Delta t)] \cdot d\tau \quad (11)$$

$$\frac{\partial E}{\partial b_j} = \int_{t_0}^t [S_j \cdot z_j(\tau) \cdot f'(x_k(\tau - \Delta t))] \cdot d\tau \quad (12)$$

$$\frac{\partial E}{\partial S_j} = \int_{t_0}^t [z_j(\tau) \cdot (f(x_j(\tau - \Delta t)) - y_j(\tau - \Delta t))] \cdot d\tau \quad (13)$$

and :

$$\frac{\partial E}{\partial T_j} = \frac{\partial E}{\partial S_j} \cdot \frac{\partial S_j}{\partial T_j} \quad \text{or} \quad \frac{\partial E}{\partial \lambda_j} = \frac{\partial E}{\partial S_j} \cdot \frac{\partial S_j}{\partial \lambda_j} \quad (14)$$

Where z_j is the backpropagated error of the j neuron, f' is the derivative of the activation function (tanh), and τ is the current time step in the time window.

The backpropagated errors are computed as the classical backpropagation by considering the network states at each time step as successive layers. Thus, the algorithm backpropagates the output errors in the network and in time. Equation (15) details the z_j calculus:

$$\begin{aligned} z_j(\tau) &= z_j^l(\tau) + z_j^2(\tau) + z_j^3(\tau) \quad (15) \\ z_j^l(\tau) &= \sum [z_k(\tau + \Delta t) \cdot S_k \cdot f'(x_k(\tau)) \cdot w_{jk}] \\ z_j^2(\tau) &= z_j^k(\tau + \Delta t) \cdot (1 - S_j) \\ z_j^3(\tau) &= e_j(\tau) \cdot \Delta t \end{aligned}$$

In (15), $z_j^l(\tau)$ represents the error coming from the other neurons, while the error due to the dynamic of the neuron j itself is given by $z_j^2(\tau)$, and $z_j^3(\tau)$ is the output error.

In classical BPTT, the z_j value is computed for each neuron and for each time step since the beginning of the learning. Nevertheless, to prevent the memory explosion induced by the storage of every network states ($t_0 = 0$), a truncated BPTT algorithm [12] (BPTT(h)) is used. Thus, the algorithm only keeps in memory the past states included in a time window that follows the current instant ($t_0 = t - h$ with h the time window width). The backpropagated error values are only computed along this sliding time window ($\tau \in [t - h; t]$). Hence only a gradient approximation is computed. This approximation is as close to the exact gradient as the time window is large.

III. MATHEMATICAL COMPARISON

A. Domain comparison

In gradient algorithms, each parameter is modified incrementally. For the time scale parameter, the two different expressions S_j^1 and S_j^2 , lead to quite different modifications during the learning.

$$S_j^1 \text{ new} = \frac{\Delta t}{T_j \text{ old} + \Delta T_j} \quad (16)$$

$$S_j^2 \text{ new} = \frac{1}{1 + e^{-\lambda_j \text{ old} + \Delta \lambda_j}} \quad (17)$$

It is obvious that for the S_j^1 expression, the modification may lead to value out of the allowed domain]0:1]. In order to avoid this problem, the variations of T_j must be restricted in a non continuous way (a strong limit on the minimum value). On the other hand, the S_j^2 values are naturally kept in this interval with asymptotic limitations.

B. Temporal responses comparison

The way the neuron dynamic varies depends on the S_j^1 or S_j^2 expressions. Starting from a slow neuron and going to a faster one the speed evolution will be different for a linear evolution on T_j or λ_j .

On Fig.1 the responses evolutions to an Heaviside input for a linear variation of T_j and λ_j are depicted. In the first graph, T_j starts from a slow response ($T_j=200s$, $S_j^1=5.10^{-4}$) and varies linearly to a fast one ($T_j=0.0101s$, $S_j^1=0.99$). In the second graph, starting from the same slow response ($\lambda_j=-9.9$, $S_j^2=5.10^{-4}$), λ_j varies linearly to the same faster one ($\lambda_j=4.59$, $S_j^2=0.99$).

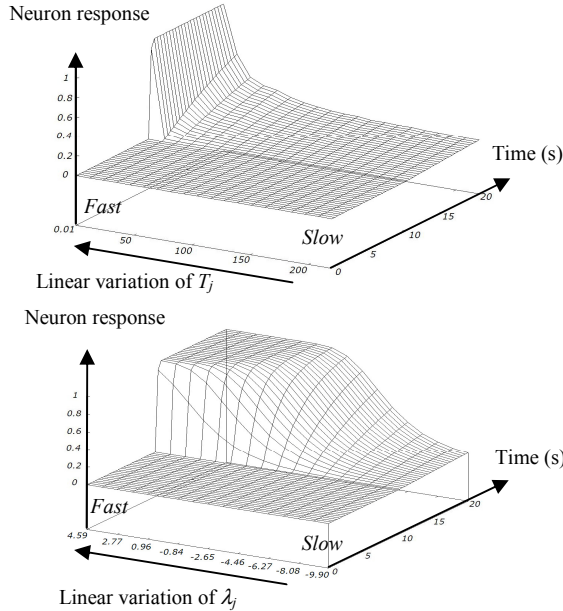


Fig 1 : Neuron responses to an Heaviside input for different speed values varying linearly according to T_j (up) and λ_j (down).

While the modification of the neuron behavior is quite rough for S_j^1 , it is smoother for S_j^2 . Actually, the second one modifies the time scale parameter in a logarithmic plan. In a

gradient based algorithm, since the modifications of the parameters are incremental, the use of the sigmoid time parameter allows to expend the S_j evolution on a larger range. Thus, the temporal behavior of each neuron can vary in a slower way.

A smoother and extended evolution is to be preferred because it makes the variation more accurate. The same range of the neuron speed corresponds to a small T_j range with S_j^1 , whereas it is contained in a larger λ_j range with S_j^2 .

C. Amplifying factor comparison in the learning

In the learning equations (14), the time parameters gradients can be developed as follows :

$$\frac{\partial E}{\partial T_j} = -\frac{\Delta t}{T_j^2} \cdot \frac{\partial E}{\partial S_j} \quad (18)$$

$$\frac{\partial E}{\partial \lambda_j} = \frac{1}{1 + e^{-\lambda_j}} \cdot \left(1 - \frac{1}{1 + e^{-\lambda_j}}\right) \cdot \frac{\partial E}{\partial S_j} \quad (19)$$

For the same value of $\partial E/\partial S_j$ a different multiplicative coefficient amplifies the value of the gradient. This amplification is much more important and bumpy with the S_j^1 expression (maximum= $1/\Delta t$), than with the S_j^2 (maximum= $1/4$). This coefficient increases the rough change of dynamics observed previously on Fig.1. Hence the difference between the smooth evolution with the sigmoid time parameter and the rough one with the time constant is amplified.

D. Comparison of the neuron dynamic modification

The natural way for changing the dynamic of a system governed by a first order equation is to modify its time constant value. The physical meaning of a time constant change, i.e. a change in the response time of the system, makes the interpretation easier of this parameter variation. A higher time constant value leads to a slower system.

Thus, for the neuron, considering a modification of the X_j parameter used, whatever it is ($X_j^1=T_j$ or $X_j^2=\lambda_j$), the interpretation will be easier by bringing ΔX_j to the corresponding ΔT_j value.

This transformation is straight for X_j^1 :

$$\Delta T_j = \Delta X_j \quad (19)$$

For X_j^2 , the transformation can be identified by equalizing the S_j new variable in (16) and (17):

$$\Delta T_j = (T_j - \Delta t) \cdot (e^{\Delta X_j} - 1) \quad (20)$$

The set of the two equations (19) and (20) expresses how

the time constant and so the neuron dynamic are modified in both cases. The differences between the two exhort the contribution of X_j^2 (λ_j) use compared to the X_j^1 (T_j) use.

The contribution can be divided into five kinds according to the ΔX_j values. These different contributions can be located by the A, B, C and D areas and the line E depicted on the following graph:

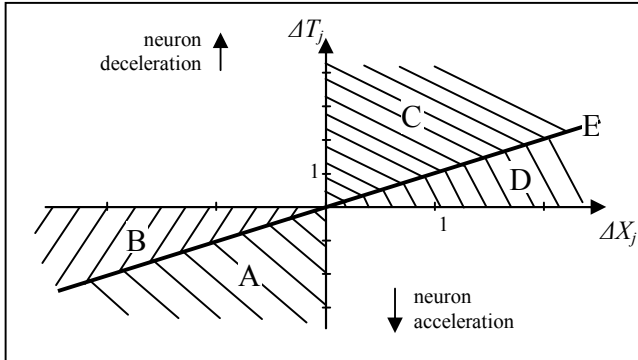


Fig 2 : Localizations of the contributions of X_j^2 use compared to X_j^1 use, in the plan defined by the ΔX_j values and the corresponding ΔT_j .

The five contributions corresponds respectively to :

- A - an increase of the asked neuron acceleration
- B - a decrease of the asked neuron acceleration
- C - an increase of the asked neuron deceleration
- D - a decrease of the asked neuron deceleration
- E - a null contribution (the use of X_j^2 brings the same modifications than X_j^1)

In the graph Fig.3, (19) and (20) are plotted in the same plan as the previous Fig.2.

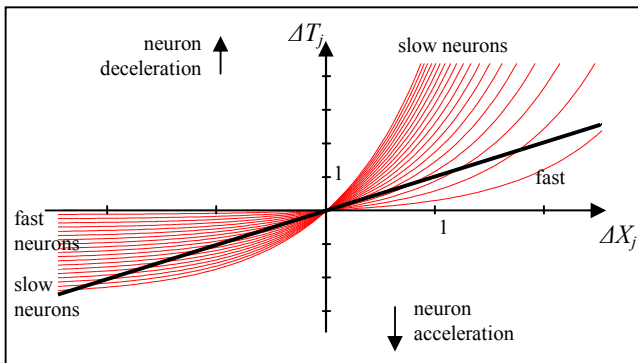


Fig 3 : Time constant variation corresponding to a variation X_j^2 (thin curves) or X_j^1 (bold line). Each curve corresponds to a different initial time constant value.

First, we can notice that, only with X_j^2 , the resulting time constant variation is linked with the current T_j value. There are several thin curves but only one bold line. Thus, according to the value of T_j (slow or fast neuron) the modification of the neuron dynamic will be different according to the neuron speed.

Then, with X_j^2 and according to the T_j values, if the neuron is slow, the acceleration or deceleration caused by the learning will be increased (thin curves are in the A and C

areas). On the other hand, if the neuron is fast, the variations will be stifled (thin curves are in areas B or D).

Nevertheless, for a fast neuron, if a strong deceleration is asked (great positive ΔX_j value), the use of X_j^2 will accentuate this deceleration (the curve goes from area D to area C).

Finally, with X_j^2 and whatever the neuron speed is, if a strong acceleration is asked (great negative ΔX_j value), ΔT_j will be asymptotically limited. The limit is $\Delta T_j = \Delta t \cdot T_j$. It leads to an instant response of the neuron ($T_j = \Delta t$).

To sum up, the use of a sigmoid time parameter rather than a classical time constant one brings the 3 following advantages:

- an asymptotic limit to the fastest behavior,
- a smoother and extended evolution of the neuron dynamic,
- a mobility improvement of the time parameter for slow neurons and a reduced mobility for faster ones.

IV. EXPERIMENTAL COMPARISON

A. Experiment description

The main objective of this experiment is to validate our mathematical deductions and to analyze the sigmoid use influence on the learning convergence.

To focus on this point, a model identification rather than a controller shaping will be used. Fig.4 shows how the identification of ROBIAN's torso influence on the ZMP (Zero Moment Point) will be performed.

The ZMP is a key notion in the balance control of a walking robot. It corresponds to the position of the center of pressure on the ground. The ZMP control, introduced thirty five years ago [13], consists in controlling the equilibrium of the biped robot by keeping the ZMP inside the polygon defined by the contact point with the ground.

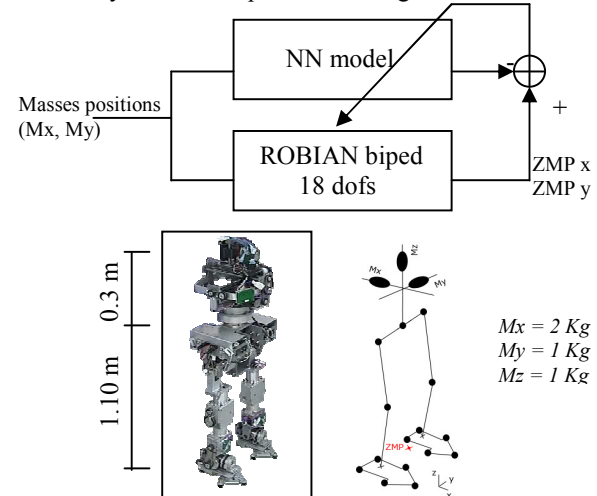


Fig 4: Learning architecture for the identification (up) of ROBIAN robot (down), with 2 legs and a 4 degrees of freedom torso.

The motion of the X and Y masses of the torso (M_x and

My) disturbs the robot equilibrium, leading to variations of the X and Y ZMP positions. The neural network (NN) tries to reproduce these influences by computing the gap between the desired positions and its own outputs. During the learning process, it will try to modify its parameters to vanish this error, by minimizing a cost function. This function is the normalized squared errors sum on the two axis, The output errors can be derived from it as follows:

$$Cost = \left(\frac{ZMP_x - y_x}{maxZMP_x} \right)^2 + \left(\frac{ZMP_y - y_y}{maxZMP_y} \right)^2 \quad (20)$$

$$e_j(t) = \frac{\partial Cost}{\partial y_j(t)} = - \frac{2}{maxZMP_j} \cdot (ZMP_j - y_j(t)) \quad (21)$$

Where y_x, y_y are the output activities on the axis and $maxZMP_j$ is the maximum amplitude on the j axis (for normalization).

Two different criteria will be analyzed to quantify the ability of the algorithm to learn the ZMP positions: convergence speed and generalization performance. The observation of the Cost evolution during the learning will inform us about the time needed for convergence and how it occurs. After the learning, a test on another pattern will reveal the learned model ability to generalize its knowledge.

Representative situations should be presented during the learning. They must fully characterize the behavior of the studied system. This implies to give the network patterns that express the dynamic of the ZMP evolution with its oscillations, damping and resonance for the 2 axis.

The learning pattern contains a succession of four different excitations applied to the robot: squared commands with varying amplitudes (0.01m to 0.20m) and with varying frequencies (1Hz-3Hz, containing the resonance of the robot), for both X and Y axis.

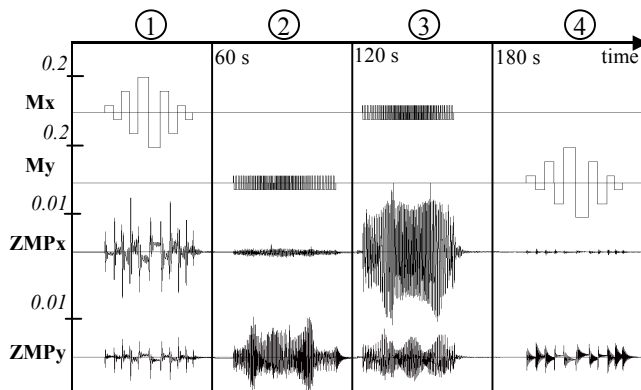


Fig. 5: Masses positions (m) and ZMP positions (m) on the two axis during the learning pattern (60x4 seconds with a sampling rate = 45Hz). In periods 1 and 4, the X mass and then the Y mass, are submitted to successive steps with various amplitude. In the 2 and 3 periods, steps with varying frequency are applied.

In the testing pattern a sine command with a varying frequency that produces the resonance is applied on both X

and Y axis.

B. Learning without time parameters modification

The Cost evolution is depicted on the Fig.6. In this experiment, the network (composed of 2 inputs 15 hidden neurons and 2 outputs) was taught with one hundred loops on the learning pattern ($\eta_1=\eta_2=0.025, \eta_3=0$ to fix the S_j values) and a time window width $h=40$.

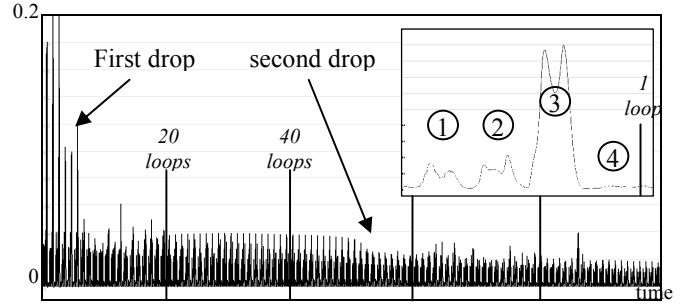


Fig. 6: Cost evolution during the learning without modification of the time scale parameters.

At the beginning of the learning, the network parameters are randomly chosen to generate stable outputs. The time scale parameters (S_j) are taken in the interval $[0.5;1]$, to obtain neurons with speeds close to the dynamic of the system.

Fig.6 shows the evolution of the cost during the learning. In the zoom depicted on the top right corner, the four cost peaks are due to the different kinds of command (amplitude variations: peaks 1 and 4, or frequency variations: peaks 2 and 3) of the learning pattern.

During the experiment, two main drops happened. They are respectively linked to a quick decrease of peak 3 (learning of the X behavior) and peak 2 (learning on Y axis). The instant when the second drop occurred (SDI : Second Drop Instant) is a good factor to quantify the convergence speed. It occurs when the network found a correct answer on the both two axis. Here it takes about 55 loops to arise.

$$SDI=55$$

After the learning, a relevant value for estimating the generalization capacity of the learned network is the costs sum during the test pattern (TES : Test Errors Sum). It expresses the difference between an ideal model of the robot and the learned one. With a perfect model, the TES value would be zero. The remaining error is here 59.

$$TES=59$$

C. Learning with time constants variation S_j^l

The learning is started again with the same initial values for the network parameters and the same learning rate η_1 and η_2 . Two different values of learning rate η_3 are compared for the modification of the time constant. The first one will be $\eta_3=0.0001$, chosen for generating a slow learning on the time constant. Then, a rate $\eta_3=0.001$, will be used for faster learning.

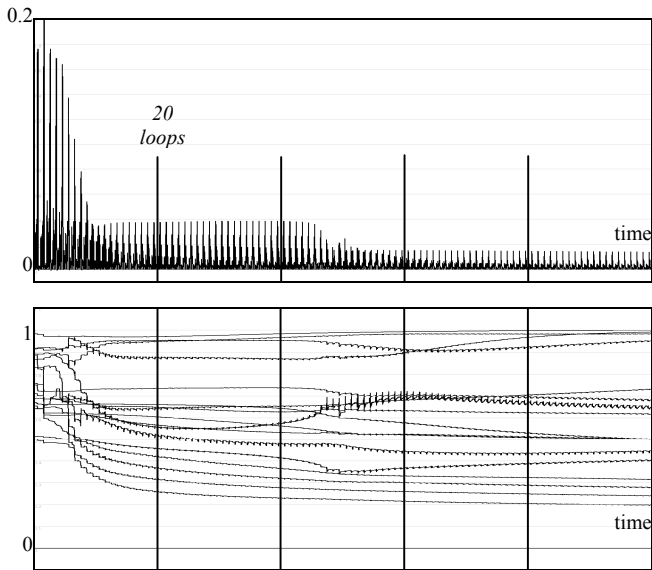


Fig. 7: Cost evolution during the learning (up) and the corresponding variations of the S_j^l values (down) through slow modification of the time constants : $\eta_3=0.0001$.

In this first learning, the modification of the time constant is quite slow. The convergence speed obtained is approximately the same as the previous learning, but the generalization capacity at the end is better:

$$\text{SDI} = 50 \quad \text{TES} = 35.1$$

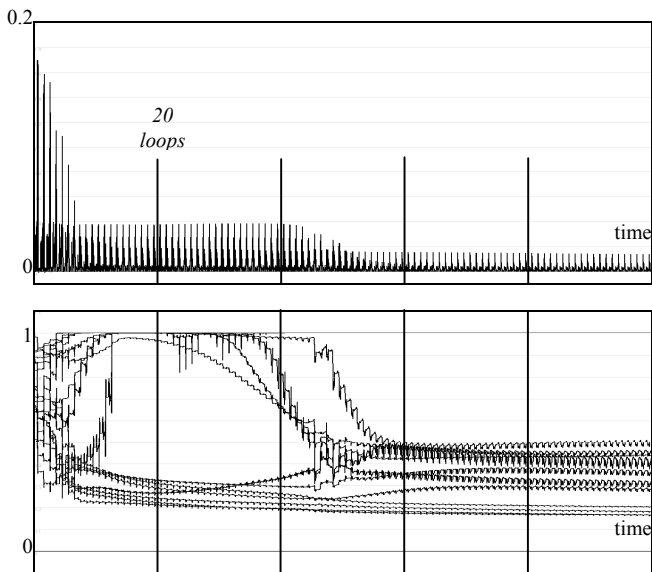


Fig. 8: Cost evolution (up) and S_j^2 variations (down) through fast modification of the time constants : $\eta_3=0.001$.

In a second learning, the modification of the time constant is faster. The convergence speed obtained is, again, approximately the same, but the learned network shows a better generalization capacity:

$$\text{SDI} = 51 \quad \text{TES} = 32.9$$

D. Learning with sigmoid time parameter variation S_j^2

For this experiment, the learning is modifying the time scale values through the modification of the sigmoid time parameters instead of the time constants. It starts with the

initial sigmoid time parameters that correspond to the previous initial time constants. The same weights and biases initial values are also used. The previous learning rates η_1 and η_2 are kept. Two η_3 values are used: $\eta_3=1.25$ and $\eta_3=0.5$. They were experimentally chosen to bring the same time scale variation speed than in the previous section.

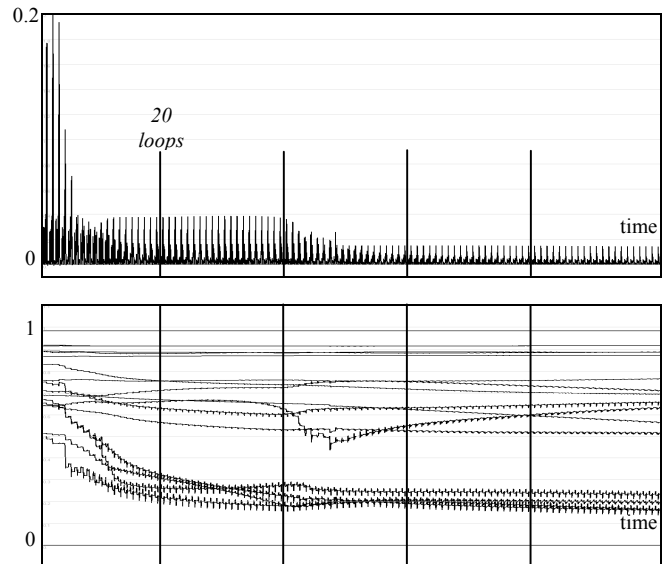


Fig. 9: Cost evolution (up) and S_j^l variations (down) through slow modification of the time constants : $\eta_3=0.5$.

During this slow learning, the convergence speed seems to be a little faster, and the learned network answers correctly on the test pattern:

$$\text{SDI} = 49 \quad \text{TES} = 37.6$$

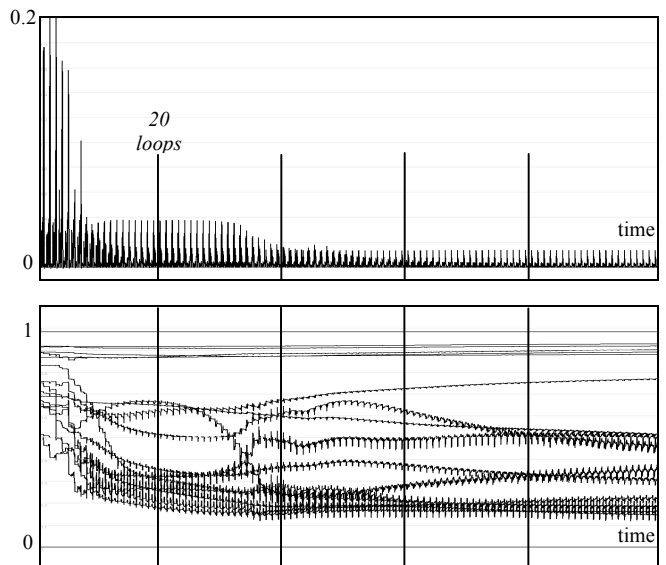


Fig. 10: Cost evolution (up) and S_j^2 variations (down) through fast modification of the sigmoid time parameters : $\eta_3=1.25$.

In this experiment, the convergence speed reaches its fastest value and the generalization test shows the best result.

$$\text{SDI} = 32 \quad \text{TES} = 31.7$$

V. DISCUSSION

A. Validation of the 3 mathematic deductions

By observing the time scale modifications graphs, it is possible to validate the mathematical deductions given in section III.

1) Asymptotic limit to fastest behaviors

A neuron reaches its fastest dynamic when its time scale value is 1. During the S_j^1 learning (Fig.7 and Fig.8), the algorithm drives several S_j^1 parameters to this maximum value. One neuron answers instantaneously for slow learning, and 6 neurons do so for fast learning (superposed on the graph). Actually, the algorithm hardly blocks the modification of the T_j parameters to this limit. The consequence can easily be seen on graph Fig.8. The time scale values, and, as a consequence, the neurons speed, are not slowed down while approaching the instant limit. When they reached it, they get stuck on it (after 20 loops, 9 neurons are stuck and only 3 of them managed to slow down).

This phenomenon is troublesome because the range of possible network dynamic behaviors decreases with the number of neurons stuck. The problem is not the instant responses (that could be useful) but the attracting power of this limit that confines the network's dynamic evolution.

This problem does not occur in the S_j^2 (Fig.9 and Fig.10). Some time scale parameters approach the instant limit but it does not behave as an attractor. The first mathematical deduction is thus experimentally confirmed.

2) Smoother and extended evolution of the dynamic

The network dynamic is strongly linked with the time scale values. If all neurons are fast (time scale parameters close to 1), the network will generate a fast response to its input excitation. In the same way, if they all are slow, the global response will be slow. But, if the speeds are homogeneously distributed, the network could arise slow or fast responses. The possible behaviors contained are more varied. Thus, during the learning, it is important to allow the S_j to evolve in the same way for full possible range]0:1]. If not, some ranges should be kept aside and some dynamic behavior would be difficult to arise.

However it clearly appears in Fig.8 that S_j^1 learning brutally jumps between a slow and a fast range (where the parameters are getting stuck). The algorithm avoids the interval [0.5:0.9[, because in this range, the speed variation are very sensitive to little modifications of T_j .

With the S_j^2 modification, this not occurs (Fig.10). Although this experiment presents approximately the same modification speed, the time scale values are varying in an homogeneous way along the whole range.

Thus, this remark meets the second mathematical

deduction. The time constant variation induces a brutal modification of the time scale parameters whereas the sigmoid time parameter modification is smoother and extended on the whole range.

3) Mobility improvement or reduction

In the backpropagated error equation (15), the second term behaves as a first order along the time window [2]. So, the faster the neuron, the more the backpropagated error (z_j) variation on this neuron will be important. Moreover, if the S_j value of this neuron also varies, this will induce disturbing variations of the weights and biases parameters (in (11) and (12), z_j and S_j are multiplied). The convergence stability and speed could be affected by this phenomenon.

The evolution of the time scale values of Fig.8 and Fig.10 shows that the compared methods have opposite behaviors. On the one hand, the S_j^1 use implies oscillating values for fast neurons and slow variation for slow neurons. On the other hand, the S_j^2 expression induces a varying behavior for slow neuron and smooth variations for fast neurons. This observation validates to the third mathematical deduction.

B. Consequences

The study of the cost evolution graphs and the SDI and TES values provides the experimental consequences of these deductions.

Firstly, comparing the cost evolution graphs of Fig.7, 8, 9 and 10 to the one obtained during the no modification experiment Fig.6, it is obvious to say that the time scale modification brings a stabilization of the convergence. Without modification of the neuron dynamic, the cost meets a lot of difficulties to clearly converge to a minimum. It takes more loops for the second drop: SDI = 55, and the convergence is less stable. More than 90 loops are needed to reach the cost range obtained just after the SDI point in the other experiments.

The convergence speed, expressed by the SDI values, is approximately the same for all experiments except for the last one where it's better. There, the homogeneous modifications help the weights modification to reach the convergence in a better way. Whereas, in the other experiments, it does not help enough or perturbs it.

Finally, the quality of the created nets can be evaluated through the TES values. Although the TES values obtained are not strongly different, it is possible to extract some remarks from these results. The worst is the one obtained with no modification of neurons dynamics. The best result is achieved with strong modifications of the sigmoid time parameters. This is probably due to the way the time scale values vary. They explore a wide range of time scale values

smoothly and homogeneously. As far as the two low rates learning are concerned, the modification only seems able to help the convergence stabilization but doesn't manage to find a good set of time scale value for decreasing the cost.

The experiments carried out validated the mathematical deductions. It also highlights their influence on the learning behavior. Actually, the use of a sigmoid time parameter rather than a classical time constant one shows a faster learning convergence, and more accurate neural networks.

VI. CONCLUSIONS

In this study, a comparison between two different ways of expressing the time scale parameter in a leaky integrator neural network (classical time constant and sigmoid time parameter) was carried out. The approach focused on identifying the one that fits best to the adopted learning method based on truncated dynamic backpropagation though time algorithm.

At first, a mathematical comparison of the influence on the neuron behavior when time scale changes occur was made. Next, an experimental study based on a real system identification (a biped robot) was carried out.

It appears that the sigmoid expression brought the following advantages:

- an asymptotic limit to an instantaneous response of the neurons, which is important to avoid stuck behaviors,
- a smoother and extended evolution of the neuron dynamic, that provides a more homogeneous path in the whole time scale range $]0;1]$,
- a mobility improvement of the time parameter for slow neuron and a stiffer mobility for faster one, which brings a less oscillating convergence.

These advantages lead a faster convergence of the learning algorithm and more accurate neural networks.

As a further development, the adopted learning algorithm will be tried for controlling the equilibrium of the ROBIAN biped robot and for generating force trajectories on the BIA road simulator. Thus, the stability and speed of our learning method will be tested on two different systems (electric and hydraulic). Thus its limits will be shown, as far as real time control, robustness and generalization are concerned.

REFERENCES

[1] B. Mohamed, F. Gravez, F.B. Ouezdou, "Emulation of the dynamic effects of human torso during walking gait", in *Journal of Mechanical Design*, vol. 126, Issue 5, pp 830-841, Sept 2004.

[2] B.A. Pearlmutter, "Gradient calculation for dynamic recurrent neural networks: a survey", in *Transactions on Neural Networks*, 6(5):1212-1228, 1995.

[3] P.J. Werbos, "Backpropagation through time: what it does and how to do it", *Proceedings of the IEEE*, vol. 78, no. 10, pp. 1550-1560, 1990.

[4] A.J. Robinson and F. Fallside, "Static and dynamic error propagation networks with application to speech coding", In Anderson pp. 632 – 641, 1987.

[5] D.E. Rumelhart, G.E. Hinton, R.J. Williams, "Learning internal representations by error propagation Parallel distributed processing: explorations in the microstructure of cognition", eds D E Rumelhart, J L Mc-Clelland and the PDP Research Group (MIT Press, Cambridge MA) pp 318–362, 1986.

[6] Nguyen, M.H. and G.W. Cottrell, "Tau Net: A neural network for modeling temporal variability", in *Neurocomputing* 15 pp. 249-271, 1997.

[7] S. Hochreiter, A.S. Younger, and P.R. Conwell, "Learning to learn using gradient descent", in lecture notes on Comp. Sci. 2130, proc. Intl. Conf. on Artificial Neural Networks (ICANN-2001), pages 87-94. Springer: Berlin, Heidelberg, 2001.

[8] K. Doya. "Bifurcations of recurrent neural networks in gradient descent learning". Submitted to *IEEE Transactions on Neural Networks*, 1993.

[9] Draye, J., Pavisic, D., Libert, G, "Dynamic recurrent neural networks: a dynamical analysis", *IEEE Trans. on Systems Man and Cybernetics, Part B*, vol 26, n 5, pp. 692-706. 1996

[10] R.D. Beer, "Parameter space structure of continuous-time recurrent neural networks", submitted, the supplementary *Mathematica notebook*, 2005.

[11] F-S. Tsung. "Modeling Dynamical Systems with Recurrent Neural Networks". PhD thesis, Department of Computer Science. University of California, San Diego, 1994.

[12] R. J. Williams and D. Zipser, "Gradient-based learning algorithms for recurrent connectionist networks", In Y. Chauvin and D. E. Rumelhart, editors, *Backpropagation: Theory, Architectures, and Applications*, Erlbaum, Hillsdale, NJ, 1990.

[13] Vukobratovic, M. and Borovac, B.. "Zero-moment point – thirty five years of its life" in *International Journal of Humanoid Robotics*. 1(1): 157-173. 2004.

- 9.3** Muscle emulation with DC motor and neural networks for biped robots. *International Journal of Neural Systems*, 2010

MUSCLE EMULATION WITH DC MOTOR AND NEURAL NETWORKS FOR BIPED ROBOTS

HAYSSAM SERHAN

*University of Versailles S Quentin,
LISV Laboratory, France
hserhan@hotmail.com*

CHAIBAN G. NASR

*Lebanese University — Faculty of Engineering I, Lebanon
chnasr@ieee.org*

PATRICK HENAFF

*ETIS UMR 8051
University of Cergy-Pontoise-ENSEA-CNRS, F-9500, France
patrick.henaff@u-cergy.fr*

This paper shows how to use a DC motor and its PID controller, to behave analogously to a muscle. A model of the muscle that has been learned by a NNARX (Neural Network Auto Regressive eXogenous) structure is used. The PID parameters are tuned by an MLP Network with a special indirect online learning algorithm. The calculation of the learning algorithm is performed based on a mathematical equation of the DC motor or with a Neural Network identification of the motor. For each of the two algorithms, the output of the muscle model is used as a reference for the DC motor control loop. The results show that we succeeded in forcing the physical system to behave in the same way as the muscle model with acceptable margin of error. An implementation in the knees of a simulated biped robot is realized. Simulation compares articular trajectories with and without the muscle emulator and shows that with muscle emulator, articular trajectories become closer to the human being ones and that total power consumption is reduced.

Keywords: Neural Networks; NNARX; muscle model; tuning PID; biped walking.

1. Introduction

Biorobotics seeks to develop new robotic technologies modeled on the performance of human and animal neuromuscular systems. The excellent ability of animals to negotiate rough terrain has caused engineers to focus on properties of muscle and other unique features in order to design improved robots for exploration. Creating an artificial system that can mimic mammalian muscle behavior is an important task that has endless applications in two domains. The first one is in the field of artificial prostheses for handicap applications. The second is in the field of bio-inspired robots in which we try to get a robot to simulate natural human walking. Skeletal Muscle is a remarkable biomechanical system which converts

chemical energy into mechanical energy sufficient to move the skeleton.

Creating such artificial system implementable in a real robot would represent significant progress.

Since many years, many studies have been reported on how to model various aspects of muscles using artificial actuators. One way to achieve this feat (see Ref. 1 for a review) consists of using special actuation systems like pneumatic actuators (McKibben muscle) or hydraulic, shape memory alloys, electroactive polymers... But, for the moment, these solutions based on new mechanisms, are not as efficient as DC motors, because they are not linear and powerful enough and then require designing new control laws.

Another way is to use DC motor with special muscle-like mechanical system (based on springs and cables) that provides mechanical compliance in the articulation in order to reproduce the properties of mammalian muscle, especially to store and restore energy during movement.²⁻⁴ In the domain of legged robotics, when the DC motor becomes unnecessary to produce the movement, this approach is called passive walking.⁵ This approach is interesting but it performs to make new articulations and new limbs.

The first contribution of this paper is to demonstrate that it is possible to emulate a muscle in a robot articulation without mechanical transformation. In this approach the compliance is controlled by tuning the PID controller with a neural controller that learns on line the tuning so that the DC-motor output behaves like a muscle or like a model of a muscle. The second contribution is in the validation of proposed scheme on the legs of a simulated biped robot in order to obtain an equivalent walking gait to the one of human.

PID Tuning is a research area and a detailed review is outside the scope of this article. However, some recent studies show that the problem of tuning PID for non-linear applications is not yet resolved. Therefore in Ref. 6, authors propose a new tuning method based on the Nyquist curve analysis for a class of unknown plants. In their method, no plant model is assumed and the control is sufficiently robust to gain variations. Fuzzy approaches were utilized to tune PID controllers. In Refs. 7 and 8 a two-level tuning method for fuzzy PID controllers is described. The method can be used for multi-input/multi-output processes and ensures the stability of the closed-loop.

It is well-know that Neural Networks are very efficient methods in the identification of non-linear systems and in the learning of complex control laws. Then, coupling Fuzzy logic and Neural Network seems efficient in the control of complex systems. In Ref. 9 authors use a new optimization algorithm to simultaneous design the FNNs and tune a PID controller. In Ref. 10, a control scheme based on neuro-fuzzy approximators is tested in the cart-pole balancing problem and in a DC-motor model.

Several models of NN exist and their complexity depends on problem to solve and on learning algorithm used. RBF networks have been used in the identification of non-linear systems like muscle¹¹

or other complex models.^{12,13} Gaussian neural network is used in the control of piezo-electric system for vibration suppression.¹⁴ Recurrent neural networks¹⁵ was used to identify or to control non linear systems.¹⁶⁻¹⁸

All these methods are interesting but they consume too much time computation to be implemented in a humanoid robot that has at least seven motors on each leg. Then, in the approach presented here, simpler neural models that are able to learn on line in real time are used. The muscle model is identified by a NNARX structure and an MLP neural network learns on-line how to change the parameters of the PID-Controller.

In a precedent paper,¹⁹ the use of NNARX has been justified, detailed and results were presented. NNARX models are similar to their ARX linear counterpart. This predictor is always stable (even if the system is unstable) because there is a pure algebraic relationship between prediction, past measurements and inputs. This is a very important feature of the NNARX structure in the nonlinear case since the stability issue is more complex than in linear systems.

This paper is organized as follows. Section 2 presents two configurations using the NNARX muscle model and a MLP Neural Network with a special learning algorithm to control the parameters of a PID controller. Section 3 is dedicated to results. First, the two configurations of the emulator are compared and a test of robustness is provided on the best one. Then, an implementation in the knees of a simulated biped robot is realized and simulation compares articular trajectories with and without the muscle emulator. A discussion is conducted in Sec. 4, and the paper concludes with perspectives for future work.

2. Physical Emulation of a Muscle

This paper looks into the idea of enabling the output of the DC Motor, combined with the PID Controller in a closed loop, to behave like a real muscle by a simple variation of the K_c , T_i and T_d parameters of the PID Controller.

In this approach, a Multi-Layer Perceptron (ANN) is proposed to tune the parameters of the PID Controller as shown in Fig. 1. The goal of this ANN is to learn to change on-line the parameters

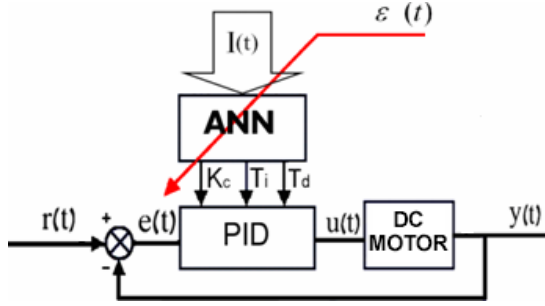


Fig. 1. Principle of the tuning approach. $y(t)$ is the torque output of the DC motor, $r(t)$ is the signal command (like a neuromuscular signal), $l(t)$ is the vector of ANN input signals, $\varepsilon(t)$ is the backpropagated learning error issued from a comparison between motor response and muscle model response.

of the PID to allow the output of the DC Motor to follow that of the NNARX Muscle Model.

To get the ANN tuner to learn, two different structures can be used. In the first one, called Model Matching the Muscle Model is connected in parallel to the closed loop formed by the Motor and the PID Controller [DC-Motor & PID controller], and the output of the muscle is used as desired output. In the second structure called Model Following, the Muscle Model is connected in series to the [DC-Motor & PID controller] system and the output of the Muscle Model is connected like a reference to this closed loop.

In the preliminary version of this work previously published,¹⁹ these two structures have been compared deeply. It has been show that the Model Following presents best performances compared to the Model- Matching. The present paper focuses on the Model-Following structure. First, this study consists of comparing control results obtained when the learning is led with the calculation through the actuator equation (Fig. 2.) and when it is led by a neural identification of the actuator, the DC motor in our case (Fig. 3.) Secondly, effects of this structure on the walking gaits of a simulated biped robot are carried on.

2.1. Learning with actuator model or with actuator identification

The usual algorithm for back-propagation is based on direct error in which each output is compared with its desired value

$$\varepsilon(t) = (y_d(t) - y(t)) \quad (1)$$

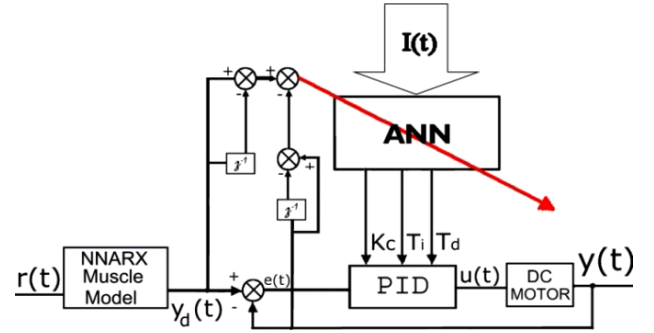


Fig. 2. Block diagram of serial structures called Model Following. $y(t)$ is the torque output of the DC motor, $y_d(t)$ is the output of the Muscle Model, $r(t)$ is the signal command, and $l(t)$ is the vector of input signals.

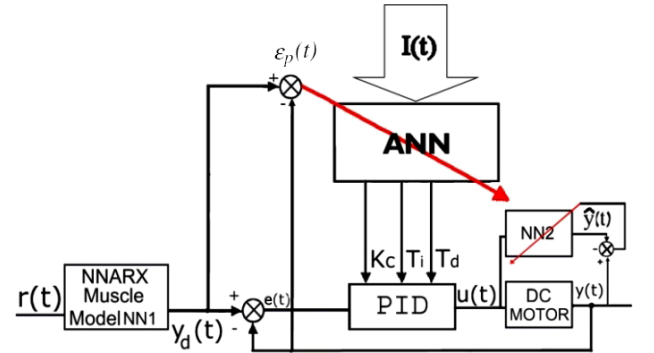


Fig. 3. Structure of the serial configuration with identification of the actuator.

and the corresponding training gradient is back-propagated through the network.

In the Model-Following structure, we want the output $y(t)$ of the PID Controller & DC Motor loop to follow that of the Muscle Model $y_d(t)$. In order to update the weights of the ANN which update the PID parameters, we designed a special learning back-propagation algorithm since using an actuator model and the conventional error (9) will lead to mathematical equations that are impossible to solve. Moreover, the training of the network must be implemented in an indirect manner because the desired value of parameters K_c , T_i and T_d are unknown.

Here, two solutions are possible for the algorithm: learning with a model of the actuator and then, the learning error is different from Eq. (1); or learning with an identification of the actuator and then learning error is Eq. (1).

In the first solution, which corresponds to structure of Fig. 2 we use a dynamic model of the actuator. Then, we can calculate the analytical expression of the training gradient based on the following error (also suggested by S. Takagi²⁰):

$$\varepsilon_v(t) = (\Delta y_d(t) - \Delta y(t)) \quad (2)$$

The second solution is based on the structure of Fig. 3 in which an identification of the actuator is carried out with a Neural Network NN2 (this is especially needed in cases where the calculation of the plant equation is too complicated) and the learning is based on the following error:

$$\varepsilon_p(t) = (y_d(t) - y(t)) \quad (3)$$

Subsection 2.3 details calculation of each solution.

2.2. Architecture of the ANN tuner

The architecture of the ANN tuner used to tune the PID is represented in Fig. 4.

The parameters of this ANN were determined after a few previous experiments (see Ref. 19). The value of the sampling time period T_s used to calculate the outputs and to update the weights of the network is set at 100 ms. The parameters of the two nonlinear functions are $a = b = 1$. It should be noticed that the value of parameter c has a direct influence on system stability because this parameter can be assimilated as a gain in the output layer. We set this value at $c = 0.1$ in all experiments presented in this paper. In the same way, the number of neurons in the hidden layer (NH) will be set at $NH = 12$.

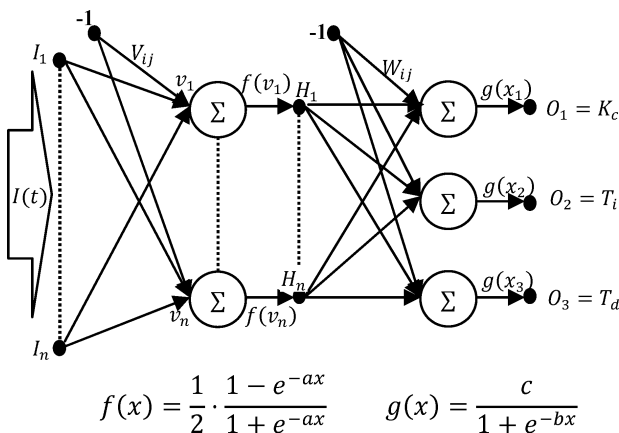


Fig. 4. Artificial Neural network which tune the PID controller

The only thing that remains is the choice of input signals that may give the neural network more information about the dynamics of the nonlinear system. It is evident from the literature that a muscle could be approximately modeled as a linear system with a transfer function of the second order.²⁰ So introducing past samples of the signal command should enable the output of the muscle model and the error to contribute to an improved learning profile for the neural network. As a result, considering the second order of past input, the input vector is:

$$I(t) = [e(t), e(t - 1), e(t - 2), \rho(t)] \quad (4)$$

where $\rho(t)$ is a signal that gives the neural network directional information when the command signal $r(t)$ rises up or falls down, as mentioned by Takagi *et al.* Indeed, these authors have demonstrated that it will be useful to use this signal $\rho(t)$ as input for the neural network. They propose the following expression:

$$\rho(t) = \begin{cases} \eta & \Delta r(t) = r(t) - r(t - T_s) > 0 \\ \rho(t - 1) & \Delta r(t) = r(t) - r(t - T_s) = 0 \\ -\eta & \Delta r(t) = r(t) - r(t - T_s) < 0 \end{cases} \quad (5)$$

where $\Delta = 1 - z^{-1}$ is the derivative operator for a sampling period T_s .

2.3. Indirect learning with Back-Propagation Algorithm

Regarding the Back-Propagation Algorithm without identification of the actuator, the output quadratic error which must be minimized has the following expression:

$$\varepsilon_v^2(t) = (\Delta y_d(t) - \Delta y(t))^2 \quad (6)$$

where $\Delta = 1 - z^{-1}$, y is the system output and y_d is the corresponding desired value. So, in this case, the error is the difference between the variations of the outputs of the two systems.

Regarding the Back-Propagation Algorithm with identification of the actuator we should minimize the following classical output quadratic error:

$$\varepsilon_p^2(t) = (y_d(t) - y(t))^2 \quad (7)$$

This error is the difference between the output of the muscle model and that of the DC motor.

Because the output of the ANN tuner is composed of the 3 PID parameters, we have to recalculate the entire back-propagation algorithm for both

the output and the hidden layer of the neural network. The idea is to update the value of the neural network weights in order to minimize the global error:

$$\xi(t) = \frac{1}{2} \varepsilon_{p,v}^2(t) \quad (8)$$

This minimization is based on the gradient descent method in which we have to calculate, for the 2 configurations, the partial derivative of the global error with respect to the weight to be equal to zero:

- Configuration with a model of the actuator:

$$\frac{\partial \xi(t)}{\partial W_{jk}} = \frac{\partial \xi}{\partial \varepsilon_v} \cdot \frac{\partial \varepsilon_v}{\partial \Delta y} \cdot \frac{\partial \Delta y}{\partial \Delta u} \cdot \frac{\partial \Delta u}{\partial O_k} \cdot \frac{\partial O_k}{\partial W_{jk}} \quad (9)$$

- Configuration with identification of the actuator:

$$\frac{\partial \xi(t)}{\partial W_{jk}} = \frac{\partial \xi}{\partial \varepsilon_p} \cdot \frac{\partial \varepsilon_p}{\partial y} \cdot \frac{\partial y}{\partial u} \cdot \frac{\partial u}{\partial O_k} \cdot \frac{\partial O_k}{\partial W_{jk}} \quad (10)$$

In the Eqs. ((9) and (10)) some terms are easy to calculate by referring to Eqs. (2), (3), (6) and (7):

$$\begin{aligned} \frac{\partial \xi}{\partial \varepsilon_{p,v}} &= \varepsilon_{p,v}(t), & \frac{\partial \varepsilon_v}{\partial \Delta y} &= -1 \quad \text{and} \\ \frac{\partial \varepsilon_p}{\partial y} &= -1 \end{aligned}$$

The other terms depend on:

- the PID regulation: $\frac{\partial u}{\partial O_k}$ and $\frac{\partial \Delta u}{\partial O_k}$
- the ANN tuner: $\frac{\partial O_k}{\partial W_{jk}}$
- the actuator: $\frac{\partial y}{\partial u}$ and $\frac{\partial \Delta y}{\partial \Delta u}$

Calculation with respect to the PID regulation

Due to the PID control, we know that:

$$u(t) - u(t-1) = k_c \left(\Delta + \frac{T_s}{T_i} + \Delta^2 \frac{T_d}{T_s} \right) e(t) \quad (11)$$

Because $O_1 = K_c, O_2 = T_i, O_3 = T_d$, we are able to calculate for the configuration without identification:

$$\frac{\partial \Delta u}{\partial O_k} = \begin{cases} \left(\Delta + \frac{T_s}{T_i} + \Delta^2 \frac{T_d}{T_s} \right) e(t) \\ -\frac{K_c T_s}{T_i^2} e(t) \\ \Delta^2 \frac{K_c}{T_s} e(t) \end{cases} \quad (12)$$

And for the one with identification:

$$\frac{\partial u}{\partial O_k} = \begin{cases} e(t) + \frac{T_s}{2T_i} (e(t) + e(t - T_s)) + \frac{T_d}{T_s} \Delta e(t) \\ -\frac{K_c T_s}{4T_i^2} (e(t) + e(t - T_s)) \\ \frac{K_c}{T_s} \Delta e(t) \end{cases} \quad (13)$$

Calculation with respect to the ANN tuner

V_{ij} and W_{jk} are the weights referring to Fig. 4, we introduce the following parameters:

- of the two successive Neural Network layers.
- H_j , is the output of the hidden layer:

$$H_j = f \left(\sum_{i=0}^p V_{ij} \cdot I_i \right) \quad (14)$$

- O_k is the output of the output layer:

$$O_k = g \left(\sum_{j=0}^q W_{jk} \cdot H_j \right) \quad (15)$$

Then, for the output layer of the two configurations, we have:

$$\frac{\partial O_k}{\partial W_{jk}} = \frac{b}{c} O_k (c - O_k) H_j \quad (16)$$

The same derivatives could be determined for updating the weights of the hidden layer:

$$\begin{aligned} \frac{\partial O_k}{\partial V_{ij}} &= \frac{\partial O_k}{\partial H_j} \cdot \frac{\partial H_j}{\partial V_{ij}} \\ &= \frac{b}{c} O_k (c - O_k) W_{jk} \cdot a \left(\frac{1}{2} + H_j \right) \left(\frac{1}{2} - H_j \right) \end{aligned} \quad (17)$$

Calculation with derivative through an actuator model

We calculate the partial derivative of Δy (variation of the output between two consecutive samples) with respect to Δu (variation of the input between two consecutive samples). This calculation is of great complexity as we have to calculate the derivative of the output with respect to the control signal which leads to calculating the derivative through the plant. But since we calculate the partial derivative of Δy with respect to Δu and not y with respect to u , the problem is simplified. Consider the DC motor as a

second order system, thus (p is the Laplace variable):

$$H(p) = \frac{A}{(1 + \tau p)(1 + \tau' p)} = \frac{Y(p)}{U(p)} \quad (18)$$

This Laplace equation corresponds in the time space continuum to a differential with the sample period T_s :

$$\tau\tau'\ddot{y}(t) + (\tau + \tau')\dot{y}(t) + 1 = Au(t)$$

where:

$$\dot{y}(t) = \frac{y(t) - y(t - T_s)}{T_s}, \quad \ddot{y}(t) = \frac{\dot{y}(t) - \dot{y}(t - T_s)}{T_s} \quad (19)$$

Hence:

$$Au(t) = y(t) + \frac{\tau + \tau'}{T_s} y(t - T_s) + \frac{\tau\tau'}{T_s^2} \times (y(t) - 2y(t - T_s) + y(t - 2T_s)) \quad (20)$$

And:

$$\begin{aligned} Au(t - T_s) &= y(t - T_s) \\ &+ \frac{\tau + \tau'}{T_s} (y(t - T_s) - y(t - 2T_s)) \\ &+ \frac{\tau\tau'}{T_s^2} (y(t - T_s) - 2y(t - 2T_s) \\ &+ y(t - 3T_s)) \end{aligned} \quad (21)$$

We obtain:

$$\begin{aligned} A\Delta u(t) &= \Delta y(t) + \frac{\tau + \tau'}{T_s} (\Delta y(t) - \Delta y(t - T_s)) \\ &+ \frac{\tau\tau'}{T_s^2} (\Delta y(t) - 2\Delta y(t - T_s) + \Delta y(t - 2T_s)) \\ &= \Delta y(t) + \frac{\tau + \tau'}{T_s} \Delta^2 y(t) + \frac{\tau\tau'}{T_s^2} \Delta^3 y(t) \end{aligned} \quad (22)$$

And then:

$$\begin{aligned} A\partial\Delta u(t) &= \partial\Delta y(t) + 2 \left(\frac{\tau + \tau'}{T_s} \right) \Delta y(t) \partial\Delta y(t) \\ &+ \frac{\tau\tau'}{T_s^2} 3\Delta^2 y(t) \partial\Delta y(t) \end{aligned} \quad (23)$$

Finally, we have:

$$\frac{\partial\Delta y(t)}{\partial\Delta u(t)} = \frac{A}{1 + 2 \left(\frac{\tau + \tau'}{T_s} \right) \Delta y(t) + 3 \frac{\tau\tau'}{T_s^2} \Delta^2 y(t)} \quad (24)$$

This equation expresses the influence of variation of the motor control $u(t)$ on changes in its speed.

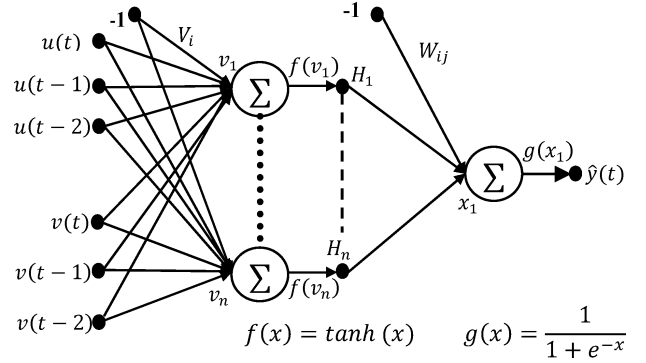


Fig. 5. MLP Neural Network NN2 used for the actuator identification.

Calculation through a neural identification of the actuator

The Neural Network structure used in this identification is an MLP presented on Fig 5. It has 6 neurons in the input layer, 8 in the hidden one, and 1 for the output layer.

The output of the NN2 is called $\hat{y}(t)$ with a sigmoid as nonlinear neuron function. The input layer is as follows:

$$\mathcal{O}(t) = [u(t), u(t - 1), y(t), y(t - 1), y(t - 2)]^T \quad (25)$$

The choice of this structure was based in the theory of system identification in the context of neural network. In this theory it is suggested that past Inputs and past Outputs could be used as input to the neural network.

The number of past measurements is related to system order and could be determined experimentally. The same holds for the number of neurons in the hidden layer that should also be determined experimentally.

Consequently, we obtained the final Network structure by taking into consideration the combination of best past measurements number and best neuron number in the hidden layer. Experiments yielded three past measurements (of input and output) and eight hidden neurons. More than three past measurements and more than eight hidden neurons did not contribute to better results. The training algorithm used is the classical gradient descent back-propagation algorithm (Fig. 6). The final learning error -i.e. difference between the output of the real plant and that of the MLP network- is on the order of 2.10^{-5} (see Fig. 7).

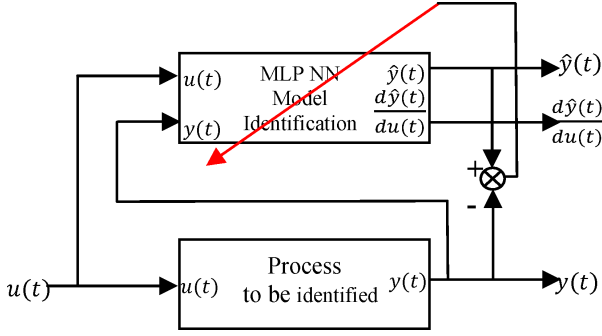


Fig. 6. Block diagram of the identification process.

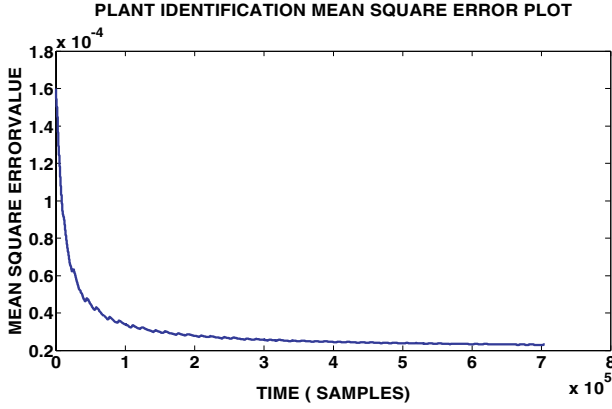


Fig. 7. Result of superposition between the output of the real plant and that of the MLP network.

Using this neural identification, we can calculate the derivative of the output with respect to the input $u(t)$. This derivative being given by the following equation:

$$\frac{\partial \hat{y}_k(t)}{\partial \mathcal{O}_l(t)} = g'(n_k^2) \sum_{i=1}^{m_h} W_{kt}^2 f'(n_i^1) W_{il}^1 \quad (26)$$

where $\mathcal{O}_l(t)$ represents input $[u(t), y(t)]$. In our case only $\frac{\partial \hat{y}_k(t)}{\partial u(t)}$ interests us. n_k^p is the input of the k th node in the p th layer, W_{ki}^p is the weight matrix of the p th layer. $g'(n_k^p)$ is the derivative of the output non linear function, $f'(n_i^p)$ is the derivative of the hidden non linear function, k is the output number, in this particular case we have only one so $k = 1$, l is the input number, here $l = 1$ because is the first input in the regression vector.

According to this identification, in Eq. (8) of the learning algorithm we can replace $\frac{\partial y(t)}{\partial u(t)}$ by $\frac{\partial \hat{y}(t)}{\partial u(t)}$.

2.4. Weights updating rules

All the aforementioned calculation will contribute to the following updating rule for the weights of the ANN PID tuner:

- for the output layer W_{jk} as:

$$\Delta W_{jk}(t) = -\alpha \frac{\partial \xi(t)}{\partial W_{jk}} + \mu \Delta W_{jk}(t-1) \quad (27)$$

- for the hidden layer V_{ij} as:

$$\Delta V_{ij}(t) = -\alpha \frac{\partial \xi(t)}{\partial V_{ij}} + \mu \Delta V_{ij}(t-1) \quad (28)$$

where α is the learning rate, and μ is the momentum term.

3. Results

We have implemented the models under Simulink in order to test the performance of the two serial structures of the Model Following. Each performance is based on the computation of the mean square error given by the expression:

$$E = \frac{1}{N} \sum_{n=0}^N (y(n) - y_d(n))^2 \quad (29)$$

For each structure we present the best results obtained. The PID controller used in all our simulations is designed so that its parameters K_c , T_i , and T_d are inputs, and the error, and its integral and derivative $e(t)$ are outputs. The Neural Network is designed as an S-Function simulator. The Motor block is a transfer function of the DC Motor, and the input command $r(t)$ is a specially designed signal under Matlab that emulates a neuromuscular command.

To enable the system to work properly, many tests were carried out to enable us to:

- Tune the different parameters.
- Choose the appropriate inputs for the neural network, the best number of neurons in the hidden layer, and the best sampling frequency.
- And lastly, pick the best combination of all the above items.

In the following paragraphs we apply our algorithms to the Model Following structure and we test the two configurations: with an actuator model and with an identification of the actuator (which in

this case is a DC-Motor). For each configuration we present the best results obtained.

The number of neurons in the hidden layer of the ANN tuner is based on experience from our previous work.¹⁹ The choice of the sampling frequency is based on the response time of the system; while taking into consideration the learning time of the algorithm. This is done in order for the neural network to be able to calculate the corresponding outputs and make the necessary weight modifications within at least one sample time. The motor transfer function is given by Eq. (13). The motor response time is about 50 ms.

Good results were obtained from the first attempt and we were not obliged to go through all the cases. Consequently, with only a sampling period of 100ms and 12 hidden neurons for the ANN tuner, the output of the physical system was able to follow that of the Muscle Model. Parameters a, b, c that define the form of the nonlinear functions of the neural network are defined in Sec. 2.2.

3.1. Learn to follow the muscle model

The mean square error plot during learning shows good convergence of the learning process (Fig. 8). After about four hundred thousand samples, this error is about $2 * 10^{-5}$.

After learning, *i.e.* during the control, we calculated the error value E Eq. (29), in two zones. The first one corresponds to the fast variations of the muscle model output, and the second corresponds to the slow variations.

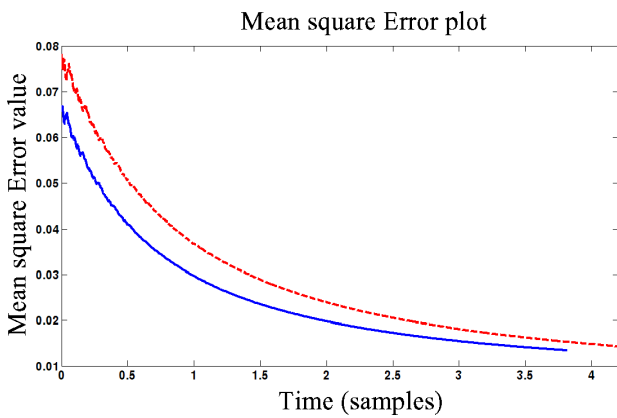


Fig. 8. Comparison when learning with actuator model (dot lines) and with NN identification of the actuator (solid line).

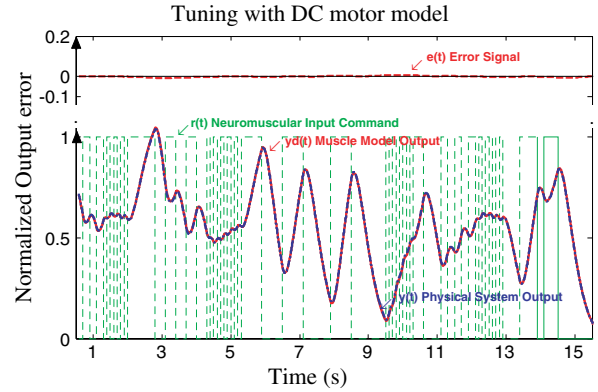


Fig. 9. Result obtained (Model Following) using derivative calculation of the DC motor. Muscle Model output $y_d(t)$ is represented by a solid line, physical system output $y(t)$ is represented by a dotted line, and $r(t)$ is represented by dot-to-dot line.

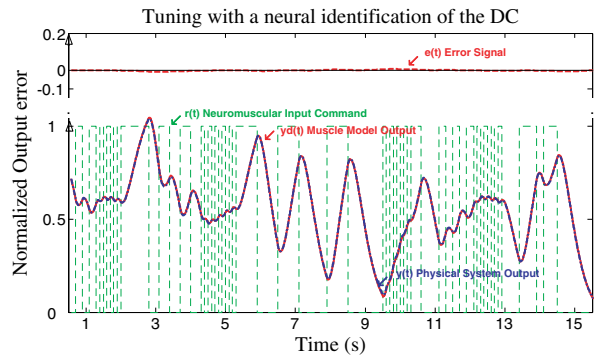


Fig. 10. Result obtained (Model Following) using NN Identification of the DC motor.

The graph of the results obtained when tuning the PID with a model of actuator is represented in Fig. 9. Improvement in the two zones of fast and slow variations of the muscle model output are very significant with only an error E of 1% in both zones. The results obtained when tuning the PID with neural identification of actuator are given in Fig. 10. Result is very similar to the one of Fig. 9.

Other simulations that further increase the number of neurons in the hidden layer and/or decreases of the sampling period did not contribute any enhancements (but here they were obtained without even knowing the transfer function of the actuator).

3.2. Test of robustness

In order to evaluate the robustness of our controller, we implemented three transfer functions for the DC

Motor in the simulator. We implemented the transfer functions representing different operating conditions in order to see how the system would behave when the DC Motor or its parameters change. The nominal plant is given below (taken from Ching-Hung Lee *et al.*²¹):

$$H_1(p) = \frac{15}{0.005p^2 + 0.1p + 1.305} \quad (30)$$

The plant with loading variant:

$$H_2(p) = \frac{18}{0.032p^2 + 0.072p + 1.28} \quad (31)$$

The third function is an identification of the DC-Motor:

$$H_3(p) = \frac{18}{0.031p^2 + 0.077p + 1.25} \quad (32)$$

To achieve our aim, in the simulator we implemented a random function which selects one of the three transfer functions randomly and continuously during each simulation.

Figure 11 shows an enlarged scale of the graph of the behavior of the learned control system when the transfer function of the DC Motor changes (test of robustness). We can see that an error will be present when the transfer function changes but for less than 2 seconds.

The neural network adapts the PID controller with the necessary parameters and the system regains very good superposition. Even during those two seconds the system remains stable and no oscillations or overpass appear. The same results were obtained for both configurations (with and without identification).

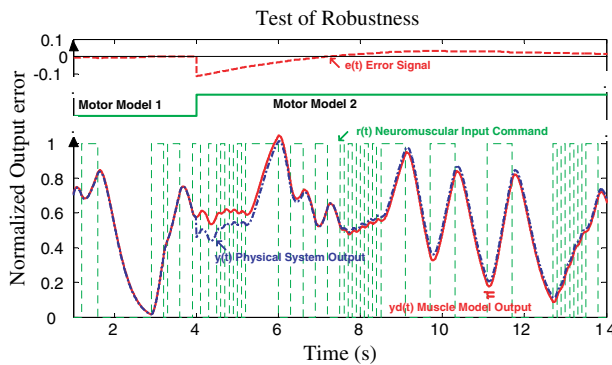


Fig. 11. Test of Robustness (Model Following) using NN Identification of the DC motor.

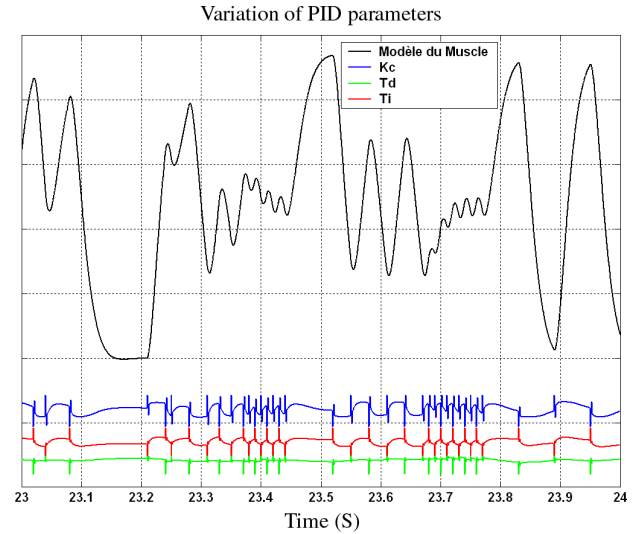


Fig. 12. Variations of K_c , T_i , T_d parameters after learning, observed on a short time duration (values are normalized).

Figure 12 shows the variations of the PID parameters during a time instance. We can see that gains modification in real time is therefore effective and remains stable. K_c and T_i vary most while T_d remains almost constant. But in general we can say that K_c , T_i and T_d undergo small changes during the slow variations of the model of muscle and undergo major changes while shocking the model. This particularly happens when the model signal is choppy. This in turn allows the PID controller to respond quickly to changes in the muscle model and the electric motor to follow it. The “peaks” which can be seen in the variations, occurring with each signal interruption of the model do not destabilize the PID adaptation. It should be noted here that with fixed PID parameters the system cannot be stabilized.

It is clear that identifying the actuator using the NN without knowing anything about its dynamic characteristics or its transfer function yielded the same result that calculating the derivative through the actuator model. But in the case of plant identification we do not have to take into account the parameter variation of the plant because the identification neural network calculates the derivative of the motor output with respect to the control signal for each sample based on its network weights. Then this approach must be used if the actuator model is difficult to obtain.

3.3. Implementation in the knees of a simulated biped robot

In this section we integrate the muscle emulator into the control laws of the knee joints of a simulated biped robot doted of six active degrees of freedom on each leg.²⁴ Each Knee is drive with a 90W DC motor (Maxon RE-35) with 1/90 gear. Here the Following structure with the model of the knee DC motor is used. The biped robot is simulated under OpenHRP simulator. A walking algorithm based on a state machine provides a stable 3D dynamic walk of the biped (see all details in Ref. 22). Walking algorithm and muscle emulator are coding in C++ in OpenHRP.

The state machine is based on the analysis of the full human walking cycle that can be divided into eight phases (Fig. 13).

Simulations are compared with those obtained in Ref. 21 without the muscle emulator. The control scheme proposed for DC motors of both knees of the robot is shown in Fig. 14.

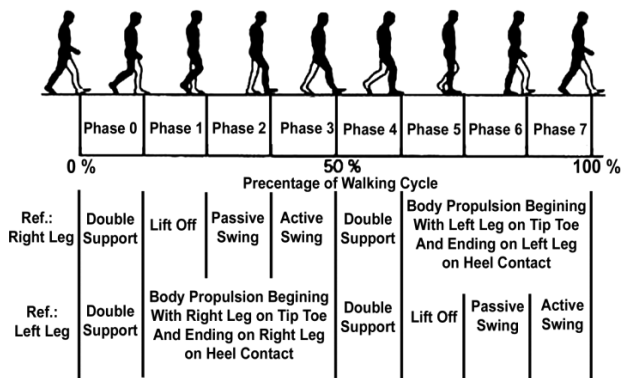


Fig. 13. Full human walking cycle description (Right Leg). Functionality of each phase can be depicted with respect to one leg acting first as swing leg and then as stance leg (see Ref. 22 for more details).

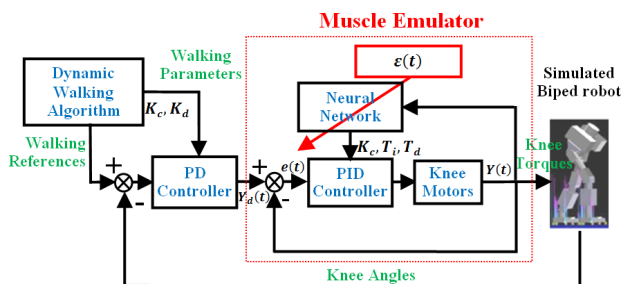


Fig. 14. Control scheme of proposed adaptation of PID knees by the neural emulator of muscle.

The other ten motors use the same structure as that already described in Ref. 22. This schema consists of two levels: a low-level control implemented by neural networks that tunes the parameters of the PID DC motor, and a high level corresponding to the dynamic walking algorithm based on the state machine. The neural network is already programmed in accordance with the previous section, but in any case we can always perform a relearning of the neural network if necessary.

Figure 15 compares articular angle values of the human (data are average values extracted from

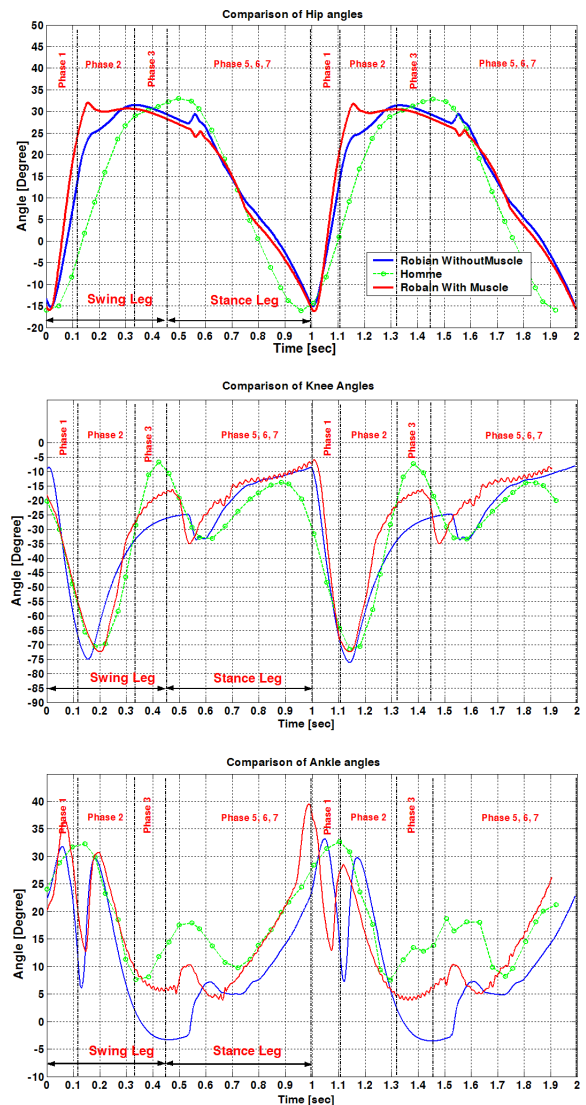


Fig. 15. Comparison of joint angles between an average human and biped robot (right leg) for the angles of the hip, knee and ankle for two walking cycles. The speed is 0.6 m/sec.

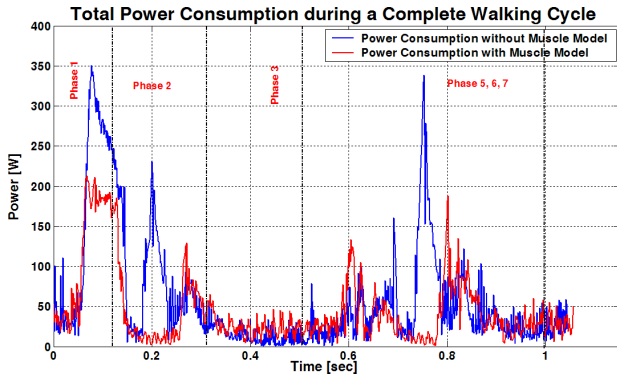


Fig. 16. Graphs of total motor power of hip, knee and ankle for a complete cycle of walking (right leg).

multiple experiments and given in Ref. 23), and those obtained with or without muscle emulator on robot under OpenHRP simulations. Here human data are adjusted with a homothetic transformation to the dimension of the robot. In this way, overlap between robot angular values and those of human robot are acceptable on major part of the walking cycle. In phases 1 and 2, the gap between robot and human is due to the sole of the foot of the robot that is rigid unlike that of human. Phase 4 is too short to be visible.

Because the muscle emulator is implemented in the knee, it has little effect on hip. However, during all the walking phases, for knee and ankle, it tends to be closer the angular variations of the robot from those of human being. The emulator smoothes knee angular trajectory and tends to absorb the rapid changes of the electric motor. During Phase 3 human beings tends to extend the knee up and then to flex after contact with the ground, which is difficult to do with the model without muscle emulator but possible with it.

Figure 16 presents graphs of total power consumption of the six DC motors of one leg during the walking cycle. The total consumption of the robot is significantly reduced through the muscle model that allows the motor to store and restore energy when PID gains produce lower damping as springs.

4. Discussion

In this approach the muscle model is connected as a reference to the closed loop formed by the PID controller and the DC Motor. Also we have evaluated two learning algorithms. This structure is simple

(4 inputs and 12 neurons in the hidden layer), its training time was about 1 minute (on a Pentium IV computer running at 3.0Ghz with 1GB RAM) and the sampling period was not necessarily high (100 ms) enough to make it interesting for a real time implementation on a real robot. This has resulted in a much shorter calculation time with an output error of less than 1%. In an online robustness test, we have changed the transfer function of the DC Motor and demonstrate that the Neural Network responded and programmed the PID controller rapidly.

One important is that the variations of the PID parameters do not converge to a specified value after finishing the learning algorithm, but continued to change based on the variation of the muscle model output. This is because it is not possible to run a physical match between the two systems using the PID controller. What occurs here is that at each sampling period the neural network tries to find the best combination of these parameters in order for the motor output to follow that of the model. This is the reason the sampling period plays a crucial role in our system. This sampling period had to be very short (5 to 10ms) in the parallel structure. However, this sampling period could be higher (100ms) in the model following structure, because in this structure the physical system only has to follow the model output which is a slower signal compared with the neuromuscular signal input. So, we lowered the changing rate of the PID parameters that gave us more favorable constraints for real time implementation.

All these results have encouraged us to implement the muscle emulator in a simulated robot. This robot has DC-Motors acting as actuators with PID controllers (which is the case in almost all robots). This should apply in low level control while keeping the high level control as it is. Simulations show that it is possible to tune articular PID controllers to improve the robot's walk towards that of human being. Moreover, total power consumption is reduced with the muscle emulator.

5. Conclusion and Future Work

This paper has demonstrated that it is possible to physically emulate a nonlinear muscle model based on a DC-Motor whose PID controller is tuned by a Neural Network. When the actuator model is uncertain, we show that using a neural identification of the

actuator is possible. The muscle emulator has been implemented in a simulation of an existing robot that uses DC Motors as joint actuators and a PID as low level controllers. With this approach, the robot is able to walk like a human by keeping the high level control unchanged. The results show that this solution is advantageous especially in terms of power consumption. However, further work must be done in future. First, it will be necessary to study effect of muscle emulator in robot's ankles and an experiment on the real biped robot must be carried on to validate this approach definitely.

Acknowledgement and Dedication

The co-authors Patrick Hénaff and Hayssam Serhan would like to dedicate this paper to the memory of their friend and colleague Dr. Chaiban Nasr who died during the revision of this paper.

References

1. A. O'Halloran and F. O'Malley, Materials and technologies for artificial muscle: A review for the mechanical muscle project. Department of Electronic Engineering, National University of Ireland Galway, Available online at http://www.tcd.ie/bioengineering/documents/ChapterVII_000.pdf
2. C. S. Lee and R. V. Gonzalez, Fuzzy logic versus a PID controller for position control of a muscle-like actuated arm, *Journal of Mechanical Science and Technology* **22** (2008) 1475–1482.
3. Y. Tamura, M. Saito and R. Nagato, A new motor model representing the stretch induced force enhancement and shortening induced force depression in skeletal muscle, *Journal of Biomechanics* **38**(4) (2005) 877–884.
4. F. Garcia-Cordova, A. Guerrero-Gonzalez, J. L. Pedreno-Molina and J. C. Moran, Emulation of the animal muscular actuation system in an experimental platform, in *Proc. of IEEE International Conference on Systems, Man, and Cybernetics*, Vol. 1 (2001), pp. 64–69.
5. McGeer, T. Passive dynamic walking, *International Journal of Robotics Research* **9**(2) (1990) 62–82.
6. K. L. Moore and Y. Q. Chen, Relay, feedback tuning of robust PID controllers with iso-damping property, *IEEE Transactions on Systems, Man, and Cybernetics, Part B* **35**(1) (2005) 23–31.
7. S.-J. Ho, S. Li-Sun and H. Shinn-Ying, Optimizing fuzzy neural networks for tuning PID controllers using an orthogonal simulated annealing algorithm OSA, *IEEE Transactions on Fuzzy Systems* **14**(3) (2006) 421–434.
8. G. K. I. Mann, H. Bao-Gang and R. G. Gosine, Two-level tuning of fuzzy PID controllers, *IEEE Transactions on Systems, Man, and Cybernetics, Part B* **31**(2) (2001) 263–269.
9. E. Harinath and G. K. I. Mann, Design and tuning of standard additive model based fuzzy PID controllers for multivariable process systems, *IEEE Transactions on Systems, Man, and Cybernetics, Part B* **38**(3) (2008) 667–674.
10. G. G. Rigatos, Adaptive fuzzy control with output feedback for H-infinity tracking of SISO nonlinear systems, *International Journal of Neural Systems* **18**(4) (2008) 305–320.
11. N. D. Donaldson, H. Gollee, K. J. Hunt, J. C. Jarvis and M. K. N. Kwende, A radial basis function model of muscle stimulated with irregular inter-pulse intervals, *Med. Eng. Phys.* **17**(6) (1995) 431–441.
12. R. V. Mayorga and J. Carrera, A radial basis function network approach for the computational of inverse continuous time variant functions, *International Journal of Neural Systems* **17**(3) (2007) 149–160.
13. R. Savitha, S. Suresh and N. Sundararajan, A fully complex-valued radial basis function network and its learning algorithm, *International Journal of Neural Systems* **19**(4) (2009) 253–267.
14. S. Suresh, N. Kannan, N. Sundararajan and P. Saratchandran, Neural adaptive control for vibration suppression in composite Fin-Tip of aircraft, *International Journal of Neural Systems* **18**(3) (2008) 219–231.
15. A. M. Schaefer and H. G. Zimmermann, Recurrent neural networks are universal approximators, *International Journal of Neural Systems* **17**(4) (2007) 253–263.
16. J. Liang, Z. Wang and X. Liu, Global synchronization in an array of discrete-time neural networks with mixed coupling and time-varying delays, *International Journal of Neural Systems* **19** (2009) 57–63.
17. V. Scesa, B. Mohamed, P. Henaff and F. B. Ouezdou, Dynamic recurrent neural network for biped robot equilibrium control: preliminary results, in *Proceedings of IEEE International Conference on Robotics and Automation* (2005), Barcelona, Spain, pp. 4125–4130.
18. G. Puscasu and B. Codres, Nonlinear system identification based on internal recurrent neural networks, *International Journal of Neural Systems* **19**(2) (2009) 115–125.
19. H. Serhan, C. Nasr and P. Henaff, Designing a muscle like system based on PID controller and tuned by neural network, *Proc. of IEEE Int. Joint Conference on Neural Networks*, Vancouver (2006), pp. 10090–10097.
20. S. Takagi, T. Oki, T. Yamamoto and M. Kaneda, A skill-based PID controller using artificial neural networks, Department of Communication Engineering,

- Okayama Prefectural University (1997), pp. 4454–4459.
21. C.-H. Lee and C.-C. Teng, Calculation of PID controller parameters by using a fuzzy neural network, *ISA Transactions* **42** (2003) 391–400.
 22. H. Serhan, C. Nasr, P. Henaff and F. Ouezdou, A new control strategy for ROBIAN biped robot inspired from human walking, *Proceedings of 2008 IEEE/RSJ International Conference on Intelligent Robots and Systems IROS* (2008) 2479–2485.
 23. S. Bouisset and B. Maton, Muscles, posture et mouvement (in french, Hermann, 1995).
 24. A. Konno, R. Sellaouti, F. B. Amar and F. B. Ouezdou, Design and development of the biped prototype ROBIAN, *Proceedings of IEEE International Conference on Robotics and Automation* **2** (2002) 1384–1389.

- 9.4 A new control strategy for ROBIAN biped robot inspired from human walking. Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2008**

A New Control Strategy for ROBIAN Biped Robot Inspired from Human Walking

H. Serhan, C. Nasr (SM IEEE), P. Hénaff, F. Ouezdou (IEEE Member)

Abstract— In this paper, we show that a biped robot can walk dynamically using a simple control technique inspired from human locomotion. We introduce four critical angles that affect robot speed and step length. Our control approach consists in tuning the PID parameters of each joint for introducing active compliance and then to increase stability of the walk. This method could be easily implemented in real time because it needs acceptable calculation time. We validated the control approach to a dynamic simulation of our 14DOF biped called ROBIAN. A comparison with human walking is presented and discussed. We prove that we can maintain robot stability and walk cycle's repetition without referencing a predefined trajectory or detecting the center of pressure. Results show that the walk of the biped is very similar to human one. A power consumption analysis confirms that our approach could be implemented on the real robot ROBIAN.

I. INTRODUCTION

Biorobotics research seeks to develop new robotic technologies modeled after the performance of human and animal neuromuscular systems. These techniques lead to complex control strategies. The implementation of such control approaches in biped walking robot did not enhance the robot locomotion to mimic the human one. In addition, those techniques required many mechanical customization or simplifications. Those customizations contributed to a non-human model. However, observation of human walking let one assumes that this walk is seemingly simple. Walking inspiration from human being could be done by trying to mimic natural human locomotion on many different ways. The more common methods are:

- Biologically inspired control approaches using neural oscillators as a central pattern generator (CPG) and reflexive control [1][2][3].
- Passive dynamic walking approach and its extension to active feedback [4]-[7], [9]-[16].
- Pragmatic rules for real-time control [8].
- Tracking of optimal reference trajectories [17] [18].

Regarding the first method, many researchers have been done in this field. In 1991, Taga demonstrated the effectiveness of this approach in unpredicted environments [1]. Since then, several attempts have been made to explore

more the effectiveness of neural oscillator based controllers on legged locomotion [2]. However, sensory feedback signals also play a crucial role in such control systems. This technique has many advantages but also drawbacks that limit its utilization. The best advantage is its ability to learn through interaction with the environment. However, manual tuning of the oscillators parameters is required. In addition, many parameter optimization techniques were developed, but for the price of increased computational effort along with the increase of the state space dimension. This calculation time is too prohibitive and makes those techniques undesirable for implementation in real-time. Matsubara et al. [3] developed a new technique based on sensory feedback to CPG for adapting the controller to the environment, but also this technique has a high computational cost. So they went into a reduction of the state space dimensionality.

The second approach known as Passive dynamic walking pioneered by McGeer more than a decade ago [7] has been well studied by several researchers [4]-[6]. Passive dynamic walking is attractive for its elegance and simplicity, active feedback control is necessary to achieve walking on level ground and varying slopes, robustness to uncertainties and disturbances, and to regulate walking speed. The first result in active feedback control that exploits passive walking appeared in [5], [9], [10], [11], for planar bipeds. Passive walking in three dimensions was studied in [12] and [13]. Later the results in [11] were extended to the general case of 3-D walking in [14]. An interesting and elegant extension of these ideas appears in [15] where geometric reduction methods are used to generate stable 3-D walking from 2-D gaits. Robustness issues were addressed in [16] using total energy as a storage function in the hybrid passivity framework.

The third approach proposed by Sabourin et al. [8], is based on a control strategy that allows the transition of velocities for the dynamic walk of an under actuated two-dimensional robot without using a reference trajectory and by simple succession of active and passive phases.

The fourth approach [17][18] is based on the analysis of the variations of angles, velocity, and acceleration of human locomotion and trying to reproduce them on robots. This method has many drawbacks. First, to mimic human the robot dimensions should be customized to the model. Secondly this method cannot give the robot the possibility to adapt itself to the surrounding environment (in a non-regular ground for instance.).

In this paper, we propose to mimic human motion based on critical points that we extracted from human locomotion. This work could be considered as an enhancement to the

H. Serhan is with LISV Laboratory-Versailles S^t Quentin University-France and the Lebanese University - Faculty of Engineering I - Lebanon (phone: 961-3-317992; email: hserhan@hotmail.com)

C. Nasr is with the Lebanese University - Faculty of Engineering I - Lebanon (phone: 961-3-369245; email: chnasr@ieee.org)

P. Hénaff is with LISV Laboratory-Versailles S^t Quentin University-France (phone: +331.39.25.49.91; email:patrick.henaff@uvsq.fr)

F. B. Ouezdou is with LISV -Versailles S^t Quentin University-France (phone: +331.39.25.49.50; email:fethi.benouezdou@uvsq.fr)

idea proposed by Sabourin et al., and an extension to the real world of anthropomorphic three-dimensional robots with foot. The low level control is a PID controller. However, the high level control will be an adaptive control algorithm based on the predefined points we summarized from human locomotion and deduced from tests done by different researchers in this field. Because the proposed technique is not based on following a predefined trajectory, this will give the robot the possibility to adapt itself to the surrounding environment.

This paper is organized as follows: In section II, a description of the control technique is discussed in details along with a description of the robot and the simulator used. In section III, Implementation of the control algorithm and the results obtained will be presented in detail. In section IV, a discussion of the results obtained is being developed. This paper concludes with a perspective for further work.

II. CONTROL METHOD DESCRIPTION

A. The ROBIAN project

Our main motivation for bipedal research concerns a significant contribution to the study of human being locomotion system. For this purpose, a multi-degree of freedom biped prototype with flexible feet called ROBIAN (acronym for Robot BIpède ANthropomorphique), has been developed. The major application of ROBIAN (Fig. 1) prototype is the development of a real test bed of active/passive prosthesis devices enhancing research on the human being locomotion apparatus. ROBIAN is composed of two different parts: a locomotion system (lower limbs) and a torso mechanism (upper part). In this paper, we focus only on the locomotion systems. Each leg has a total of seven DoFs [17], [18], three actuated DoFs for the hip, one actuated for the knee, two actuated for the ankle and one passive for the foot providing flexible feet. The total height of robot is 1.50 m, its weight is 30 kg, and modular design was developed. Hence, each concerned kinematical module (hip, knee, ankle or foot) can be easily replaced by a prosthetic device to be tested. Mechanical construction of the modular locomotion system has already been done and static walking gaits were already achieved in 2003.

B. Human motion & inspired walking parameters

The work on human locomotion done by, Viel [21], Bouisset [22], Allard [23], and Winter [24] allows us to identify the major properties in the human walking that influence step length, velocity, and equilibrium in a dynamic walk.

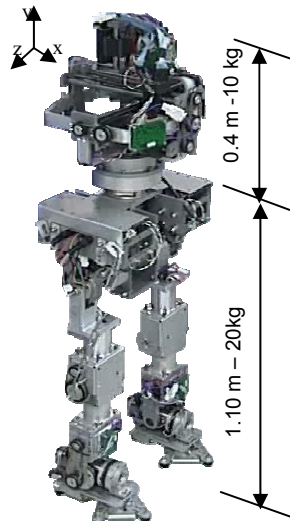


Fig. 1. The ROBIAN biped

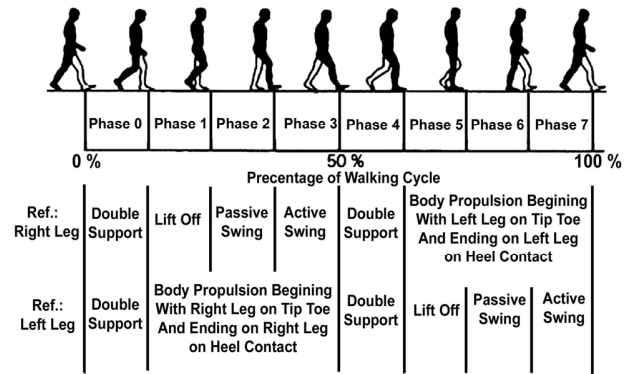


Fig. 2. Full Human Walking Cycle Description (Ref. Right Leg)

The full human walking cycle described in [21], [22], [23], and [24], can be divided into eight phases. The functionality of each phase can be depicted with respect to one leg acting first as swing leg and then as stance leg (Fig. 2). Our description will be more physical in order to inspire the way the robot will be controlled:

- Phase 0 is the double support phase (DS), during which the two legs are touching the ground. Neuromuscular commands impose body weight translation toward the stance leg in order to free the swing leg (SW) for lift-off. Muscles acting on ankle joint of the swing leg try to maintain body in a stable position.
- Phase 1 is the lift-off phase in which hip muscles contraction accelerate forward the movement of the swing leg. Muscles activating the knee joint remain passive, which contribute to knee flexing (similar to a double pendulum). Muscles acting on ankle joint of the swing leg try first to contribute in the lift-off phase (tip toe). When the leg is not touching the ground anymore, they will remain passive.
- Phase 2 begins when the thigh is accelerated enough. This phase is the passive swing. In this phase, neuromuscular commands will keep hip joint passive. It will remain passive until it catches an angle related to the step length. The same for the knee and ankle joints.
- Phase 3 is active swing phase. In this phase muscles acting hip joint and knee joint will be activated for being rigid before the leg touches the ground. The ankle joint will be activated before heel contact.
- Phase 4 is the DS phase in the next half walking cycle. During this phase, the swing leg will begin to take the role of the stance leg and vice-versa. Here also body weight is moved toward the stance leg.
- Phase 5: the swing leg is now taking the role of stance leg (ST). Here the muscles acting on the hip joint will be contracted in a way to keep the upper body in a stable inclination. Muscles acting on the knee joint will extend it, and thus provoking body propulsion (BP). Muscles acting on the ankle joint will be activated to keep body in a stable position.
- Phases 6 and 7 are similar to the fifth phase in terms of functionality related to the stance leg.

We can deduce from the above study that human locomotion is a succession of passive and active phases [8].

In the same manner, the locomotion of our anthropomorphic robot ROBIAN can be characterized by essential angles inspired from the previous description and defined in reference of the movements of the locomotion apparatus of human being in 3 plans (sagittal “S”, frontal “F” and transversal “T”) [25],[26]. These angles could be summarized as follows (Fig. 3).

In Sagittal plan (S plan):

Two angles that have direct effect on step length, and walk stability are:

- Swing leg Hip Anterior Extreme Angle: $\theta_{Max_H_S}^{SW}$
- Stance leg Hip Posterior Extreme Angle: $\theta_{Max_H_S}^{ST}$

Four angles that have direct effect on robot dynamics and propulsion are:

- Swing leg Knee Anterior Extreme Angle : $\theta_{Min_K_S}^{SW}$
- Swing leg Knee Posterior Extreme Angle: $\theta_{Max_K_S}^{SW}$
- Stance leg Knee Anterior Extreme Angle: $\theta_{Max_K_S}^{ST}$
- Swing leg Ankle Anterior Extreme Angle: $\theta_{Max_A_S}^{SW}$
- Upper Body Inclination Angle: $\theta_{Max_S}^{UB}$ has direct effect on robot speed and stability. This is due to its effect on the position of the robot center of mass (section III.B.).

Based on the swing leg Tibia Ground Contact Angle $\theta_{GC_S}^{SW}$, the control unit should regulate the lift off phase of the upcoming swing leg.

In Frontal Plan (F plan):

Two angles have direct effect on robot stability:

- Maximum Hip ABduction Angle: $\theta_{Max_H_F}^{DS}$
- Maximum Hip ADduction Angle: $-\theta_{Max_H_F}^{DS}$

Indeed, the robot will move the projection of its center of mass (CoM) to the stance leg before controlling the swing leg. Moreover, during liftoff phase the weight begins to transfer toward swing leg in order to hit the ground at $\theta_{Max_H_F}^{DS}$. In order to keep the body parallel to ground in all phases, we define frontal plan ankles angle of the stance and swing leg to follow that of the corresponding hip frontal adduction and abduction angles.

In Transversal Plan (T plan):

In this plan, we are not taking into account hip movement because internal rotation movement of the human leg is negligible. We assume that the leg is not doing any rotation.

C. PD Controllers parameters

It should be noticed that all the motors are drove by PD controllers with the predefined angles (detailed in the paragraph B) as desired angles. A general equation of each PD controller of each actuator τ is given below:

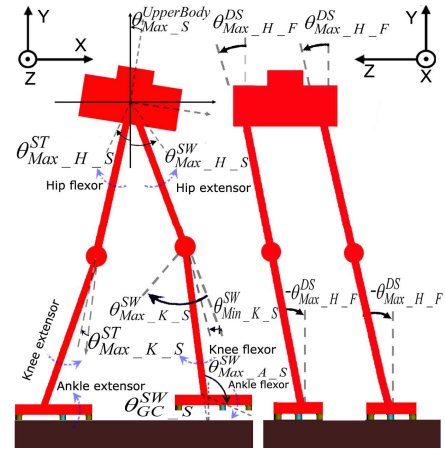


Fig. 3. Extreme angles characterizing ROBIAN Locomotion

$$\tau[Actuator] = Kc[phase]*(\theta_d - \theta) + Kd[phase]*(\dot{\theta}_d - \dot{\theta}) \quad (1)$$

Where: θ_d and $\dot{\theta}_d$ are the desired angle and velocity values.

At each phase, the values of the Kc, Kd parameters are changed. The desired velocity here is not always equal to zero, because for ankle joints a variable angle value is followed in order to maintain foot parallel to ground before contact.

The initial values of Kc and Kd are presented in Table I. They were adjusted according to the rules defined as follows [27]:

- Select a typical operating setting for the desired speed turns off derivative parts, and then increase Kc to maximum or until oscillation occurs.
- If the system oscillates divide Kc by two.
- Increase Kd and observe behavior when increasing /decreasing the desired speed by about 5%. Choose a value of Kd, which gives a damped response.
- Check if overall controller performance is satisfactorily under typical system conditions.

Table I
Initials Kc, Kd values of all PD Controller in all phases

	Phase 0, 4		Phase 1, 5		Phase 2, 6		Phase 3, 7	
	Kc	Kd	Kc	Kd	Kc	Kd	Kc	Kd
Hip_F_R	1000	25	774	47	774	47	1000	25
Hip_S_R	200	14	80	10	80	10	100	12
Knee_S_R	40	3	80	10	80	10	130	11
Ankle_S_R	-	-	90	8	90	8	80	3
Ankle_F_R	28	1.5	28	1.5	28	1.5	45	4
Hip_F_L	1000	25	774	47	774	47	1000	25
Hip_S_L	100	12	100	12	100	12	80	3
Knee_S_L	80	10	130	11	130	11	80	10
Ankle_S_L	-	-	-	-	-	-	80	3
Ankle_F_L	45	4	45	4	45	4	28	1.5

An empirical way was used for adjusting those parameters when the swing leg is near to touch the ground. Indeed, in this phase the parameters should be changed in order to touch the ground as smoothly as possible. We developed an algorithm that it will give us different values for Kc, Kd in all phases.

D. Inspired Control strategy

Our control strategy consists in analyzing ROBIAN dynamic walk through the previously determined parameters

(paragraph II.B.). We have divided our control algorithm into 12 states. States 1,3,4,5,6,10 correspond to the swing leg and states 2,7,8,9,11 correspond to the stance leg. They will actually happen in the same time if right & left leg motion during a half walking cycle is studied. The left leg is the swing leg and the right leg is the stance leg. The initial position is defined when the two legs are in contact with ground and the robot is standing up but one of the legs is behind the other. Applying techniques inspired from human locomotion already described in paragraph B Robian dynamic walk can be modeled with the Petri Net Algorithm (Fig. 4).

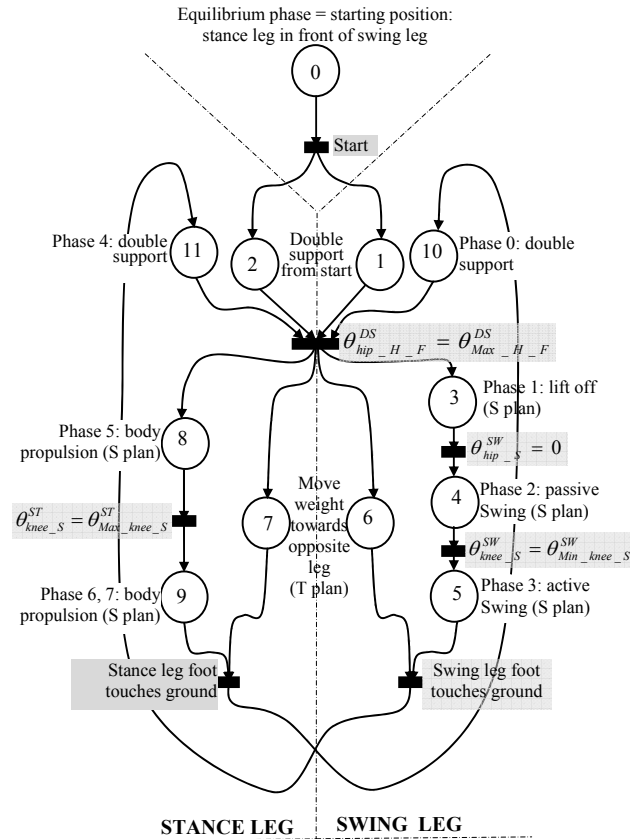


Fig. 4. Petri Net Algorithm for Robian Dynamic Walk (S_Plan or Sagittal Plan, F_Plan or Frontal Plan)

Before describing the control conditions the correspondence between the Petri Net Algorithm states and the human walking phases will be detailed:

- State 0 is the equilibrium Phase where the robot is standing up with Stance Leg in front of Swing Leg. This state is the starting position.
- State 1 & 2 correspond to the DS phase. Those two states occur only at startup. However, State 10 & 11 corresponds also to the double support phase during normal walk (short duration ≈ 25 ms).
- State 3 corresponds to the Swing leg lift-off (Phase 1).
- State 4 corresponds to the Passive Swing phase (Phase 2)
- State 5 corresponds to the Active Swing Phase (Phase 3)

- State 6 & 7 correspond to the Body weight online transfer of the swing & stance leg, respectively. They will happen in parallel to Phases 1, 2, 3 & 5, 6, 7.
- State 8 corresponds to Stance leg Body Propulsion Phase (Phase 5)
- State 9 corresponds to the Stance leg Body Stabilization Phase (Phases 6 & 7)

Control strategies are described below:

1) Phase 0 and Phase 4, “Double Support”:

In these phases the robot weight is moved to the stance leg. Swing and stance leg hips, knees and ankles in S plan have to be controlled for bringing the CoM of the robot down by flexing the knee and keeping its upper part parallel to ground (this is only done when starting from the equilibrium position). In the same way, it is necessary to control stance and swing leg hips in frontal plan for moving body toward the stance leg and approach the desired Hip

Abduction angle $\theta_{Max_H_F}^{DS}$.

Swing and stance leg ankles in frontal plan are controlled in order to follow the opposite of their corresponding hip angles. This action will lead to maintaining body parallel to ground in the frontal plan and the robot in a symmetric equilibrium position.

2) Phase 1, “Lift Off” and Phase 5, “Body Propulsion BP”:

These phases begin when right and left hip angles in frontal plan reach the desired Hip Abduction angle $\theta_{Max_H_F}^{DS}$.

First, hip motor of swing leg in S plan is controlled in order to move leg forward and hence create body inertial effects until thigh is in vertical position. This will cause swing leg knee to bend in opposite direction, upper body and swing leg ankle to extend. This is carried out in order to reach a specific position, which is the vertical thigh by programming a hip desired value that will cause the leg to accelerate forward.

The hip of stance leg is controlled in order to stabilize upper body to the desired Anterior Extreme Angle $\theta_{Max_S}^{UB}$ and to counteract swing leg hip torque.

Swing & stance leg hip motors in the frontal plan are controlled to begin a smooth transition from the desired hip abduction angle $\theta_{Max_H_F}^{DS}$ to the desired hip adduction angle $-\theta_{Max_H_F}^{DS}$, following a linear function:

$$\theta_{H_F}^{SW} = \theta_{H_F}^{ST} = -\theta_{H_S}^{SW} * \frac{\theta_{Max_H_F}^{DS}}{\theta_{Max_H_S}^{SW}} \quad (2)$$

Swing leg knee motor is kept free, which let the tibia to bend in opposite to the moving direction (flexion). The knee of stance leg is controlled in order to approach the desired anterior extreme angle $\theta_{Max_K_S}^{ST}$. This will cause robot body propulsion (BP) induced by the difference between

$\theta_{Max_K_S}^{ST}$ and $\theta_{Min_K_S}^{SW}$ required by the switching between swing and stance leg after ground contact).

Swing leg ankle motor in S plan is droved in order to not overpass the ankle anterior extreme angle $\theta_{Max_A_S}^{SW}$. Stance leg ankle in S plan is left passive for freeing body forward propulsion. This is the most important feature discovered in body propulsion, because applying any torque on the ankle joint will have direct effect of robot stability.

Finally, left and right ankle motors in frontal plan must be controlled so their corresponding angles follow the inverse of hip angles (Eq. 2).

3) Phase 2, "Passive Swing" and Phase 6, "BP":

These phases begin when thigh is in vertical position. After this position the body will move forward under gravity.

The swing leg hip in S plan is left free until its corresponding angle reaches the hip posterior extreme angle $\theta_{Max_H_S}^{SW}$. The hip of stance leg is controlled in S

plan in order to stabilize upper body at $\theta_{Max_S}^{UB}$.

Swing and Stance leg hip motors in frontal plan must be controlled for continuing smooth transition towards swing leg (Eq. 2).

Swing leg knee motor is kept free but it will start extending under inertia effect. Stance leg knee angle is stabilized.

Swing leg ankle motor in S plan is controlled in order to ensure that the foot is parallel to ground. For this end, the value of the ankle angle according to the corresponding knee and hip angles is needed. The velocity will follow the same calculation. Stance leg ankle is kept passive for freeing body forward propulsion (same as in lift off phase). Finally, left and right ankle motors in frontal plan must be controlled so their corresponding angles follow the inverse of hip angles (Eq. 2).

4) Phase 3, "Active Swing" and Phase 7, "BP & Stabilization":

These phases begin when thigh reaches the hip posterior extreme angle $\theta_{Max_H_S}^{SW}$. This is the swing leg landing phase. Swing leg hip motor in S plan is controlled for stabilizing $\theta_{Max_H_S}^{SW}$ till ground contact. The hip motor of stance leg in S plan is controlled in order to stabilize upper body.

Swing and Stance hip motors in frontal plan are controlled to continue smooth transition (Eq. 2).

Swing leg knee motor is controlled in order to stabilize it at $\theta_{Min_K_S}^{SW}$ and stance leg knee angle is stabilized at $\theta_{Max_K_S}^{ST}$.

Swing leg ankle motor in S plan is controlled so that foot touches ground in a parallel position to ground. Stance leg

ankle in S plan is controlled in order to maintain foot parallel to ground.

Finally, left and right ankle motors in frontal plan must be controlled so their corresponding angles follow the opposite of hip angles.

Stability margin for different referenced angles is summarized in the Table II. Those values were obtained after hundreds of tests applied to Robian simulator.

Table II
Stability Limit Margins

$\theta_{Max_H_S}^{SW}$		$\theta_{Max_H_S}^{ST}$		$\theta_{Min_K_S}^{SW}$	
Acceptable margin	Currently allocated	Acceptable margin	Currently allocated	Acceptable margin	Currently allocated
20° to 40°	30°	-20° to -10°	-15°	-30° to -2°	-4°
$\theta_{Max_K_S}^{SW}$		$\theta_{Max_K_S}^{ST}$		$\theta_{Max_A_S}^{SW}$	
Acceptable margin	Currently allocated	Acceptable margin	Currently allocated	Acceptable margin	Currently allocated
-85° to -55°	-75°	0° to 50°	35°	0° to 50°	35°
$\theta_{Max_S}^{UB}$		$\theta_{Max_H_F}^{DS}$			
Acceptable margin	Currently allocated	Acceptable margin	Currently allocated		
0° to 18°	6°	0° to 10°	4°		

III. RESULTS

A. Implementation of control Algorithm on simulated ROBIAN biped

We applied this algorithm on a simulated model of our biped robot ROBIAN developed in the LISV Laboratory University of Versailles in France. Because LISV is involved in the Joint Japanese-French Robotics Laboratory (JRL), the simulator used in this work is OpenHRP (Open Architecture Humanoid Robotics Platform), which is a software platform for dynamic simulation of humanoid robots, developed by AIST, the University of Tokyo and MSTC. The robot model is programmed in VRML language. All the robot specifications were taken into account. Many control algorithms were tested on the real robot and on the simulated model in order to test the level of similarity. Thus, the model used here simulates the real robot. For the simulation, the model of the ground/foot contact is based on a spring-damper system. It has direct effect on robot stability. In our model, the parameters K_p (spring coefficient) and C_p (damper coefficient) are set as follows (those values are related to robot and floor parameters, they were calculated experimentally):

- Spring-damper coefficients parallel to axis $[X, Y, Z]$:
 $K_P = [10^5, 10^5, 10^3]$ in N/m; $C_P = [10^4, 10^4, 10^4]$ in N/(m/sec);
- Spring-damper coefficients rotational around axis $[X, Y, Z]$:
 $K_R = [1.610^4, 10^3, 10^3]$ in Nm/rad;
 $C_R = [1.610^3, 10^2, 10^2]$ in Nm/(rad/sec);

ROBIAN foot contains 5 plots in rubber and the spring-damper coefficients values given before are the same for each one of them. It should be noticed here that those values

are affected also by the floor type. Static and sliding friction factors are given below:

Static friction factor = 0.6, sliding friction factor = 0.5.

The proposed control diagram of all the robot actuators is given in Fig. 5. It consists of two levels: the low level control designated by the PID controller and the high level control designated by the Dynamic Walking Algorithm, which tunes the PD controllers and impose the reference signals based on the walking phase. For the low level, we have calculated the PID parameters of the 12 motors in order to have the same adjustment as the real robot.

The high level control is the previously discussed algorithm based on a torque control mode. This algorithm will take the role of adapting the PD controllers' parameters to each walking phase and to introduce the corresponding reference signal to the low level loop. After implementation of this control technique into OpenHRP and with some minor adjustment in phase 3 of PD's controller parameters (as stated in Section II.D), a stable and precise repetitive walk was obtained.

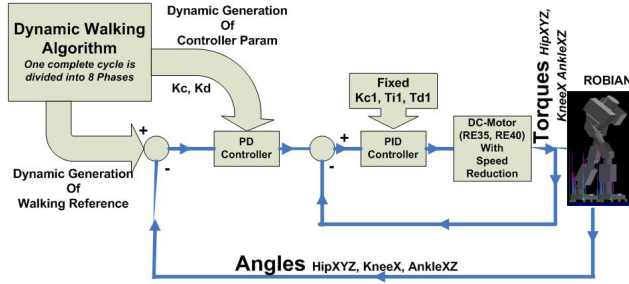


Fig. 5. Proposed dynamic walk control diagram

B. Walking analysis

Walking snapshots from frontal and S plan are represented on Fig. 6. In this simulation we are beginning from an initial position where the two feet are touching the ground. The swing leg is behind the stance leg (double support) and the walking speed is of 0.65 m/sec.

Walking speed and stability is primary influenced by four parameters: $\theta_{Max_S}^{UB}$ affects the robot center of mass

position. $\theta_{Max_H_S}^{SW}$, $\theta_{Max_K_S}^{Stance}$ and $\theta_{Min_knee_xy}^{SW}$ affect the step length and body propulsion velocity.

So with a simple variation of those four reference angle values, the step length and hence the walking speeds of the robot are controlled. The duration of one complete walking cycle is about 1 sec.

Fig. 7 shows the robot means speed variation with respect to upper body inclination angle $\theta_{Max_S}^{UB}$. We can deduce that increasing this latter one will contribute to higher walking speed. A limit for this angle (18 degrees) is determined.

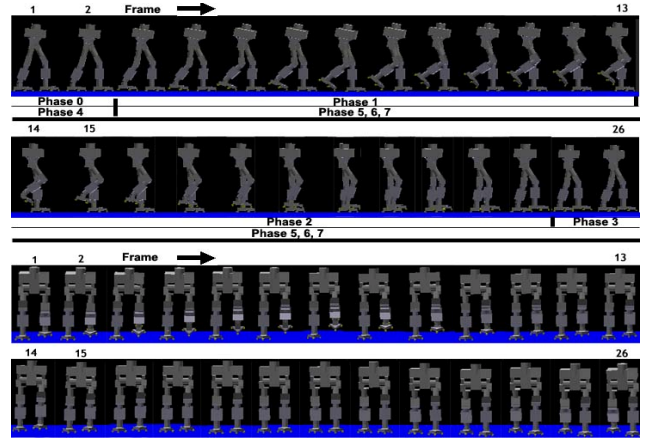


Fig. 6. Walking snapshots from F (up) and S plans (bottom) for a half walking cycle at the speed of 0.65m/s. Time between each snapshot is 19ms (frames captured from left to right).

In Fig. 8, the robot mean speed variation with respect to angle $\theta_{Min_K_S}^{SW}$ is given. Hip anterior extreme angle $\theta_{Max_H_S}^{SW}$ has also some influence on robot step length and hence the walking speed.

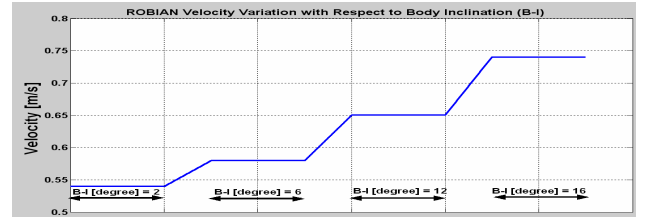


Fig. 7. ROBIAN mean velocity value with respect to Body Inclination (knee anterior extreme angle is fixed to -4 degree)

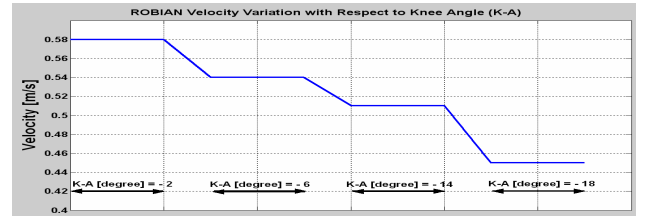


Fig.8. ROBIAN mean velocity value with respect to knee anterior extreme angle (body inclination is fixed to 6 degree)

However, one of our primary targets is inspiring human motion and applying it to ROBIAN. So a comparison between ROBIAN walking angles and human is important. In Fig. 9 a comparison between the human walking angles given by Bouisset et al. [22] and that obtained from applying our algorithm to ROBIAN under OpenHRP is depicted. The values given by Bouisset et al. are an average of values extracted from multiple human walking experiments and adapted with homothetic transform to the dimension of ROBIAN. The superposition between human walking angles and that of ROBIAN are acceptable.

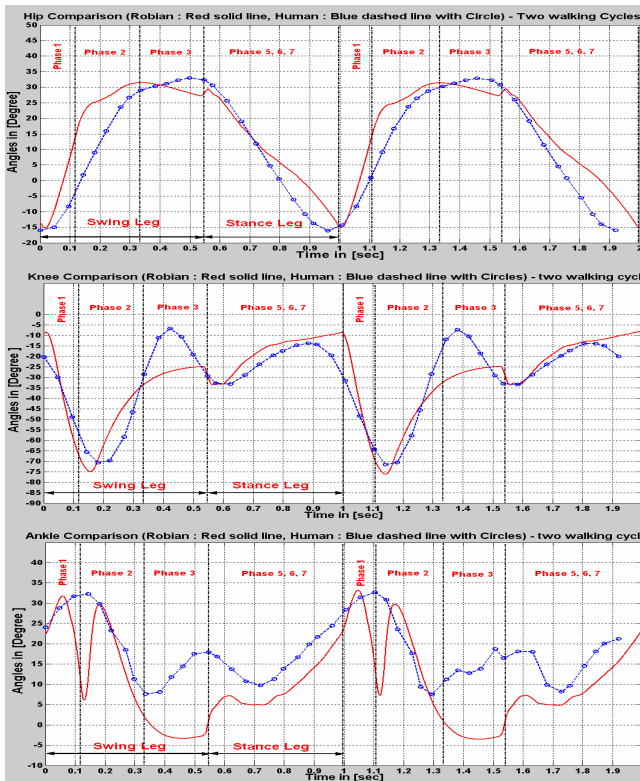


Fig. 9. Comparison graphs for hip, knee, and ankle walking angles between human and ROBIAN for two walking cycles. Phase 0 & Phase 4 are not viewable because they are very short ≈ 25 ms. Walking speed is 0.65 m/sec. Time axes origins are zeroed in order to do the comparison with human angles

Actually small errors exist. It should be noticed here that the difference is in fact caused by brutal variation of PD controllers output that don't exist in human neuromuscular commands, which are very smooth. Finally, the results obtained rely on the fact that ROBIAN was able to walk infinitely into the simulator without falling. This is due to the continuous control of the PD controllers parameters based on the walking phases, thus based on the joint positions. The superposition of successive walking phases presented in Fig 10 shows the stability of a long walk of the robot (ie. ten walking cycles).

C. Power consumption analysis

In order to implement the control algorithm on the real robot ROBIAN, the technological limits of the actuators should be taken into account. The motors used are from Maxon manufacturer (the 90Watts RE35 is installed on the three hip and one knee joints and the 150Watts RE40 is installed on the two ankle ones). First, the power issued from those motors should be within the real limits. Then, if necessary, the value of the motor velocity reduction can be changed in order to ensure that the maximum velocity and maximum torque for each motor is under the values specified into the datasheet.

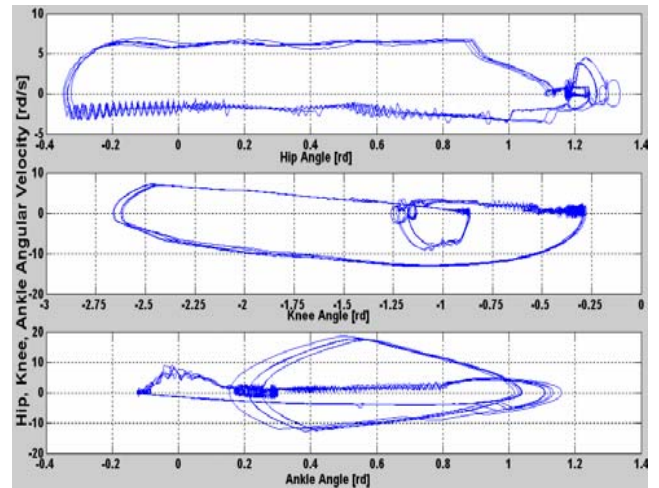


Fig. 10. Phase succession for ten walking cycles (angular velocity of Hip, Knee and ankle function of their corresponding angle)

Fig. 11 plots the graphs of the motors zone of functionality. With the values of velocity reduction (1:50 for the hip, 1:90 for the knee, and 1:50 for the ankle), the motors are working in the space of permanent use. All the motor specifications (velocity, torque and power) are taken into consideration. Those graphs were evaluated at the maximum walking speed of the robot in which case the motors are stressed the most. This simulation was an intermediate stage for validating our control approach and implementing our approach on the real robot is under development.

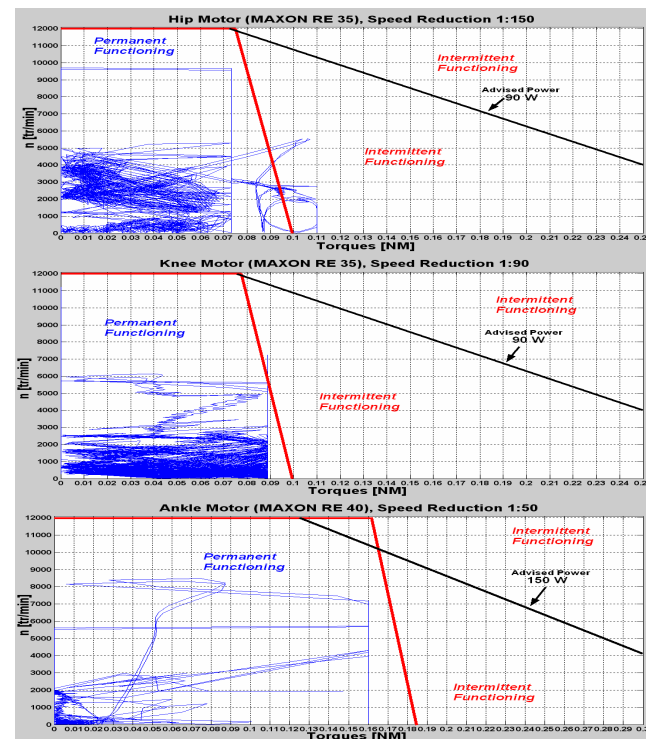


Fig. 11. Graphs of hip, knee, and ankle motors functionality under the motor physical limits for one complete walking cycle.

IV. DISCUSSION AND CONCLUSION

We achieved our goal in making in simulation the ROBIAN robot walk in 3D. We approached the human walking angles without a lot of complexity in the control. We show that hip and knee angles are very close to those of human. Regarding ankle angles, we preserved human ankle motion in the stance phase. However, in the swing phase there is a difference between the ankle angles. This difference is in the landing phase. Actually, we haven't controlled the robot foot as human in the landing phase. This is because a human foot has a lot of flexibility. However, ROBIAN foot has a passive and adjustable flexibility, which highlight the necessity of active and flexible feet. Also the feet land parallel to ground, the swinging leg will have the foot moving as human in the beginning of the swing phase. In passive swing phase, ankle motors were controlled in order to keep the foot parallel to ground. In the active swing phase, the foot was stabilized in this position in order to hit the ground in a way so that the ground reaction forces will be distributed equally on the five foot plots. In order to approach the smoothness of human neuromuscular commands, a muscle model created by Serhan et al. [28] is added to ROBIAN low level control stage. This model is based on a PID controller and a DC motor. Hence, it will be easily integrated into our control diagram without making any changes, neither to the underlying hardware system nor to the high level control algorithm (Dynamic Walk Algorithm section II.C). This integration will enhance this difference between the two graphs (Fig. 9).

Also this technique is not very consuming in processing time (algorithm execution time is about 8ms on a Pentium IV processor running at 3.0 Ghz with 1 GB RAM), so it could be implemented easily in real-time.

In conclusion, the proposed approach, allows us to achieve Robian dynamic walking similar to that of human being.

REFERENCES

- [1] G. Taga, Y. Yamaguchi, H. Shimizu, "Self-organized control of bipedal locomotion by neural oscillators in unpredictable environment," *Biological Cybernetics* 65, 1991, 147-159
- [2] G. Endo, J. Morimoto, J. Nakanishi, G. Cheng, "An empirical exploration of a neural oscillator for biped locomotion control," *IEEE, International Conference on Robotics and Automation*, 2004, pp. 3036-3042
- [3] T. Matsubara, J. Morimoto, J. Nakanishi, M. Sato, K. Doya, "Learning CPG-based biped locomotion with a policy gradient method," *Robotics and Autonomous Systems* 54, 2006, 911-920
- [4] M. Garcia, A. Chatterjee, A. Ruina, and M. Coleman, "The simplest walking model: Stability, complexity, and scaling," *ASME J. Biomechan. Eng.*, vol. 120, no. 2, pp. 281-288, 1998.
- [5] A. Goswami, B. Espiau, and A. Keramane, "Limit cycles in a passive compass gait and passivity-mimicking control laws," *Autonomous Robots*, vol. 4, no. 3, pp. 273-286, 1997.
- [6] A. Goswami, B. Thuliot, and B. Espiau, "A study of the passive gait of a compass-like biped robot: Symmetry and chaos," *I. J. Robot. Res.*, vol. 17, no. 12, pp. 1282-1301, 1998.
- [7] T. McGreer, "Passive dynamic walking," *Int. J. Robot. Res.*, vol. 9, no. 2, pp. 62-82, 1990.
- [8] C. Sabourin, O. Bruneau, J-G. Fontaine. "Pragmatic rules for real-time control of the dynamic walking of an under-actuated biped robot," *Proc. IEEE Conf. on Rob. and Automation*, 2004, 4216-4221
- [9] H. Ohta, M. Yamakita, and K. Furuta, "From Passive to active dynamic walking," in *Proc. IEEE Conf. Decision Control*, Phoenix, AZ, Dec. 1999, pp. 3883-3885.
- [10] M.W. Spong, "Bipedal locomotion, robot gymnastics, and motor air hockey: A rapprochement," in *Proc. ITech COE/Super Mechano-Systems workshop*, Tokyo, Japan, Feb. 1999, pp. 34-41.
- [11] M.W. Spong, "Passivity based control of the compass gait biped," in *Proc. IFAC Triennial World Congr.*, Beijing, China, vol. 3, July 1999, pp. 19-23.
- [12] A.D. Kuo, "Stabilization of lateral motion in passive dynamic walking," *Int. J. Robot. Res.*, vol. 18, no. 9, pp. 917-930, 1999.
- [13] S.H. Collins, M. Wisse, and A. Ruina, "A three-dimensional passive dynamic walking robot with two legs and knees," *Int. J. Robot. Res.*, vol. 20, 4, no. 3, pp. 273-286, 1997.
- [14] M.W. Spong and F. Bullo, "Controlled symmetries and passive walking," *IEEE trans. Automat. Contr.*, vol. 50, no. 7, pp. 1025-1031, 2005.
- [15] A.D. Ames, R.D. Gregg, E.D.B. Wendel, and S. Sastry, "Towards the geometric reduction of controlled three-dimensional robotic bipedal walkers," in *Proc. Workshop Lagrangian Hamiltonian Methods Nonlinear Control*, Nagoya, Japan, July 2006, pp. 117-124.
- [16] M.W. Spong and G. Bhatia, "Further results on control of the compass gait biped," in *Proc. IROS 2003*, Las Vegas, Nevada, 27-30 Oct., 2003, pp. 1933-1938.
- [17] A. Konno, R. Sellaoui, F.B. Amar, and F.B. Ouezdou. "Design and development of the biped prototype ROBIAN". In *IEEE International Conference on Robotics and Automation (ICRA)*. Washington, D.C., U.S.A., 2002, pp. 1384-1389.
- [18] R. Sellaoui, A. Konno, and F.B. Ouezdou. "Design of a 3 dofs parallel actuated mechanism for a biped hip joint". In *IEEE - International Conference on Robotics and Automation (ICRA)*, U.S.A., 2002, pp. 1161-1166
- [19] C Chevallereau and Y Aoustin, Optimal reference trajectories for walking and running of a biped robot, *Robotica* 19 (2001), pp. 557-569.
- [20] Z. Tang, Z. Sun, C. Zhou and L. Hu, "Reference Trajectory Generation for 3-Dimensional Walking of a Humanoid Robot" *Tsinghua Science & Technology* Volume 12, Issue 5, October 2007, Pages 577-584
- [21] E. Viel, "La marche humaine, la course et le saut," Masson, 2000.
- [22] S. Bouisset, B. Maton, "Muscles, posture et mouvement," Hermann, 1995.
- [23] P. Allard, J.-P. Blanchi., "Analyse du mouvement humain par la biomécanique," 2000, Décarie.
- [24] D. Winter, "Biomechanics and Motor Control of Human Movement," Third Edition, Wiley 2005.
- [25] Bouchet A., Cuilleret J., "Anatomie topographique, descriptive et fonctionnelle", T3b le membre inférieur. SIMEP 3rd ed. 1997
- [26] Kapandji I., "Physiologie articulaire, tome Membre inférieur", Maloine, 5th edition, 1999
- [27] C. Williams, "Tuning a PID Temperature Controller," <http://newton.ex.ac.uk/teaching/CDHW/Feedback/Setup-PID.html>, 2003.
- [28] Serhan H., Nasr C., Henaff P., "Designing a Muscle Like System Based on PID Controller and Tuned by Neural Network", *IEEE WCCI Congress*, Vancouver, 2006, pp. 10090-10097

9.5 State Machine-Based Controller for Walk-Halt-Walk Transitions on a Biped Robot. Proceedings of IEEE-RAS International Conference on Humanoid Robots, Humanoids 2008

A Based State Machine Controller to Achieve Dynamic Walk – Standup - Walk Cycle of Biped Robot

H. Serhan, P. Hénaff (IEEE Member), C. Nasr (SM IEEE), F. Ouezdou (IEEE Member)

Abstract— In this paper, we show how it is possible to obtain a walk – Standup - walk cycle with a biped robot using only a state machine controller. This work is a continuation of another in which the same approach was used to control the walking gait of the same robot. We introduce four critical angles that affect robot speed and step length. Our control approach consists of two control levels. The low level is a PID controller and the high level is a Dynamic Walking Algorithm, inspired from human locomotion, which tunes the PD controllers and impose the reference signals based on the walking which increases walk stability. This method could be easily implemented in real time because it needs acceptable calculation time. We validated the control approach to a dynamic simulation of our 14DOF biped called ROBIAN. We prove that we can maintain robot walking and stopping stability and walk cycle's repetition without referencing a predefined trajectory or detecting the center of pressure. Results show that the walk of the biped is very similar to human one.

I. INTRODUCTION

Biorobotics research seeks to develop new robotic technologies modeled after the performance of human and animal neuromuscular systems. These techniques lead to complex control strategies. The implementation of such control approaches in biped walking robot did not enhance the robot locomotion to mimic the human one. In addition, those techniques required many mechanical customization or simplifications. Those customizations contributed to a non-human model. However, observation of human walking let one assumes that this walk is seemingly simple. Walking inspiration from human being could be done by trying to mimic natural human locomotion on many different ways. The more common methods are those based on biologically inspired control approaches using neural oscillators as a central pattern generator (CPG) and reflexive control [1][2][3]. The best advantage of this method is its ability to learn through interaction with the environment. However, manual tuning of the oscillators parameters is required. In addition, many parameter optimization techniques were developed. In addition, many parameter optimization techniques were developed, but for the price of increased computational effort along with the increase of the state

space dimension. Other method like Passive dynamic walking approach and its extension to active feedback [4]-[7], [9]-[16] This approach is attractive for its elegance and simplicity, active feedback control is necessary to achieve walking on level ground and varying slopes, robustness to uncertainties and disturbances, and to regulate walking speed, from which we inspired some notion of passive control. Another approach proposed by Sabourin et al. [8], is based on a control strategy that allows the transition of velocities for the dynamic walk of an under actuated two-dimensional robot without using a reference trajectory and by simple succession of active and passive phases. Tracking of optimal reference trajectories [17] [18] is based on the analysis of the variations of angles, velocity, and acceleration of human locomotion and trying to reproduce them on robots. This method has many drawbacks. First, to mimic human the robot dimensions should be customized to the model. Secondly this method cannot give the robot the possibility to adapt itself to the surrounding environment.

None of those methods had the possibility to be extended for standing-up the biped robot after a walking cycle and to let the robot walk again after this standing-up phase in a stable way. The work presented in this paper comes after a previous one [29] in which we show how to use a simple description technique based on a Petri-Net algorithm to control a biped robot to walk infinitely without falling, in a similar way to humans. A full comparison between human walking angles and that of our robot was presented and discussed. Here, we propose to mimic human motion based on critical points we extracted from human locomotion in order to control the biped robot to stand-up after a stable and dynamic walking cycle and to let it walk again. Because the proposed technique is not based on following a predefined trajectory, this will give the robot the possibility to adapt itself to the surrounding environment.

This paper is organized as follows: In section II, a description of the control technique is discussed in details along with a description of the robot and the simulator used. In section III, implementation of the control algorithm and the results obtained will be presented in detail. In section IV, a discussion of the results obtained is being developed. This paper concludes with a perspective for further work.

II. CONTROL APPROACH FOR WALKING

The proposed control diagram of all the robot actuators is given in Fig.1. It consists of two levels: the low level control designated by the PID controller and the high level control designated by the Dynamic Walking Algorithm, which tunes

H. Serhan is with LISV Laboratory-Versailles S¹ Quentin University-France (phone: 961-3-317992; email: hserhan@hotmail.com)

C. Nasr is with the Lebanese University - Faculty of Engineering I - Lebanon (phone: 961-3-369245; email: chnasr@ieee.org)

P. Hénaff is with LISV Laboratory-Versailles S¹ Quentin University-France (phone: +331.39.25.49.91; email:patrick.henaff@uvsq.fr)

F. B. Ouezdou is with LISV -Versailles S¹ Quentin University-France (phone: +331.39.25.49.50; email:fethi.benouezdou@uvsq.fr)

the PD controllers and impose the reference signals based on the walking phase.

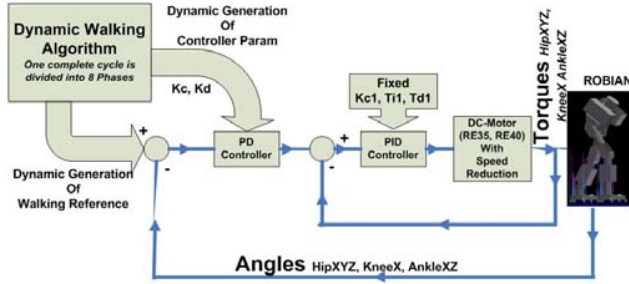


Fig. 1. Proposed dynamic control diagram

Our main motivation for bipedal research concerns a significant contribution to the study of human being locomotion system. For this purpose, a multi-degree of freedom biped prototype with flexible feet called ROBIAN (Fig.2) has been developed. ROBIAN is composed of two different parts: a locomotion system (lower limbs) and a torso mechanism (upper part). In this paper, we focus only on the locomotion systems. Each leg has a total of seven DoFs [17], [18], three

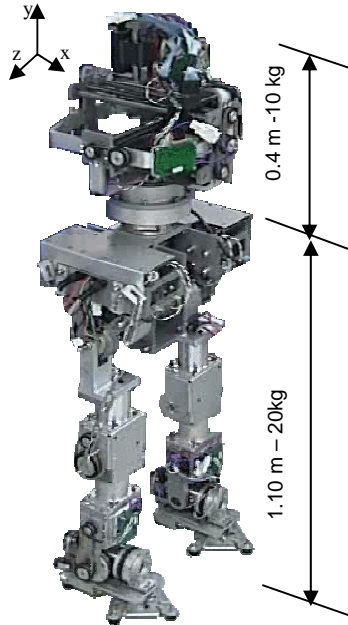


Fig.2. The ROBIAN biped robot

actuated DoFs for the hip, one actuated for the knee, two actuated for the ankle and one passive for the foot providing flexible feet. The total height of robot is 1.50 m, its weight is 30 kg, and modular design was developed.

A. Robot walking parameters

The work on human locomotion done by, Viel [21], Bouisset [22], Allard [23], and Winter [24] allows us to identify the major properties in the human walking that influence step length, velocity, and equilibrium in a dynamic walk. The full human walking cycle described in [21], [22], [23], and [24], can be divided into eight phases. The functionality of each phase can be depicted with respect to one leg acting first as swing leg and then as stance leg (Fig. 2). More description of human walking phase could be found in our previous article [29]. What is important here is that human locomotion is a succession of passive and active phases [8]. In the same manner, the locomotion of our anthropomorphic robot ROBIAN can be characterized by essential angles inspired from the previous description and defined in reference of the movements of the locomotion

apparatus of human being in 3 plans (sagittal “S”, frontal “F” and transversal “T”) [25],[26]. These angles could be summarized as follows (Fig. 3) below:

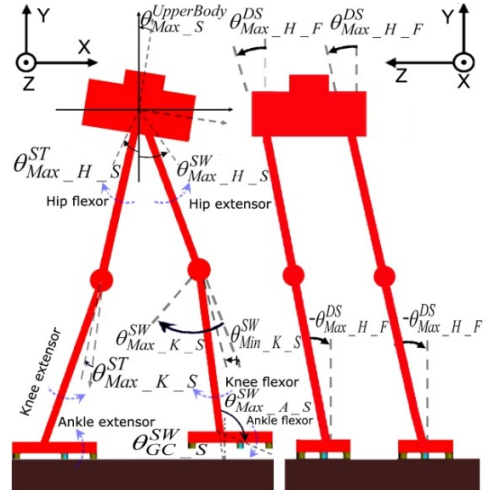


Fig. 3. Extreme angles characterizing ROBIAN Locomotion

In Sagittal plan (S plan), two angles that have direct effect on step length, and walk stability are: Swing leg Hip Anterior Extreme Angle: $\theta_{Max_H_S}^{SW}$ and Stance leg Hip Posterior Extreme Angle: $\theta_{Max_H_S}^{ST}$. Four angles that have direct effect on robot dynamics and propulsion are: Swing leg Knee Anterior Extreme Angle $\theta_{Max_K_S}^{SW}$, Swing leg Knee Posterior Extreme Angle $\theta_{Min_K_S}^{SW}$, Stance leg Knee Anterior Extreme Angle $\theta_{Max_K_S}^{ST}$, Swing leg Ankle Anterior Extreme Angle: $\theta_{Max_A_S}^{SW}$ and Upper Body Inclination Angle: $\theta_{Max_S}^{UB}$ has direct effect on robot speed and stability. This is due to its effect on the position of the robot CoM. Based on the swing leg Tibia Ground Contact Angle $\theta_{G_C_S}^{SW}$, the control unit should regulate the lift off phase of the upcoming swing leg.

In Frontal Plan (F plan), two angles have direct effect on robot stability are: Maximum Hip ABduction Angle: $\theta_{Max_H_F}^{DS}$ and Maximum Hip ADduction Angle: $-\theta_{Max_H_F}^{DS}$.

Notice that, in Transversal Plan (T plan), we are not taking into account hip movement because internal rotation movement of the human leg is negligible. We assume that the leg is not doing any rotation.

B. PD Controllers parameters

It should be noticed that all the motors are drove by PD controllers with the predefined angles as desired angles. A general equation of each PD controller of each actuator τ is given below:

$$\tau_{Actuator}^{Leg} = Kc_{Actuator}^{Phase} * (\theta_d - \theta_{Actuator}^{Leg}) + Kd_{Actuator}^{Phase} * (\dot{\theta}_d - \dot{\theta}_{Actuator}^{Leg}) \quad (1)$$

Where: θ_d and $\dot{\theta}_d$ are the desired angle and velocity values.

Leg corresponds to Stance or Swing leg. *Phase* corresponds to one of the eights walking phases described in [21],[22], [23],[24]. *Actuator* corresponds to Hip, Knee and Ankle. *Plan* corresponds to Sagittal, Frontal, and Transversal. In our

approach robot speed is controlled by simple variation of $\theta_{Max_S}^{UB}$ and $\theta_{Min_K_S}^{SW}$ as described in [29], and the values of Kc, Kd parameters are changed at each phase.

C. Inspired Control strategy (Walking Phase)

In our control strategy described in details in [29], the control algorithm is divided into 12 states. States 1,3,4,5,6,10 correspond to the swing leg and states 2,7,8,9,11 correspond to the stance leg. They will actually happen in the same time if right & left leg motion during a half walking cycle is studied. The initial position is defined when the two legs are in contact with ground and the robot is standing up but one of the legs is behind the other. Applying techniques inspired from human locomotion, Robian dynamic walk can be modeled with a Petri Net Algorithm (Fig. 4).

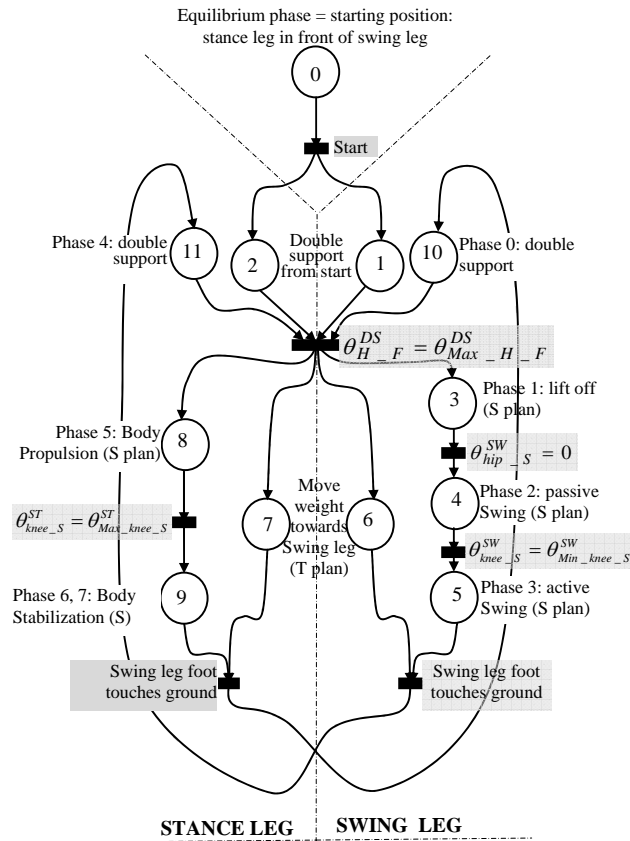


Fig. 4. Petri Net Algorithm for Robian Dynamic Walk (S_Plan or Sagittal Plan, F_Plan or Frontal Plan)

With this control approach, we obtained previously a infinite walking cycle with a dynamic simulation of ROBIAN builded under OpenHRP (Open Architecture Humanoid Robotics Platform). Fig 5 shows the superposition of ten successive walking phases extracted from a long walk of the biped [29].

The correspondence between the Petri Net Algorithm states and the human walking phases are detailed in [29].

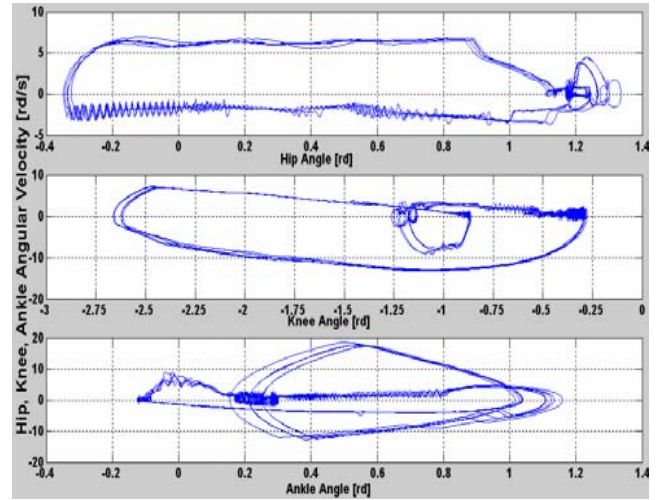


Fig.5. Phase succession for ten walking cycles (angular velocity of Hip, Knee and ankle function of their corresponding angle)

III. CONTROL STRATEGY FOR STOP AND WALK CYCLE

In order to stop ROBIAN walk cycle and keep it in a standup stable position, we modified our algorithm in order to introduce a stop phase. This algorithm will start after swing leg touches ground (State 9 in Fig. 4.). Where we swap stance and swing leg. Starting this point we extend the Stance leg Knee to its maximum in order to propagate the robot forward, same for Swing Leg Knee, so that Swing leg feet will touche ground in approximately the same position as that of the Stance leg. From this point, we start extending upper body by activating right and left hip motors, so that the upper body will be in a vertical position. Left and right ankle motors will be activated in order to follow their corresponding hip angles and therefore keep robot upper body parallel to ground. Once Swing leg feet touches ground, we continue the same process till all corresponding angles reach their destination, and after that we control all robot motors in order to memorize the reached positions. At this time, the robot is in stable standup position for which the distance between right and left legs foot is low. Then, the robot stops and remains standing up in a stable position. From this position we can control the robot to walk again by simple reapplying our walking control techniques previously described in II.C. This contributed to a stable walk-standup-walk cycle.

A. States description

State 12: (Swing Leg - S Plan) “Lift-Off & Passive Swing State”. It starts after the Stop Switch is activated. At that position most Body weight is applied on Stance leg. During this State, three actions happen in parallel:

1. Apply an active torque to the Hip for accelerate slowly the dynamic move of the oscillatory leg and create body inertial effects until thigh is in vertical position:

$$\tau_{H_S}^{SW} = Kc_{H_S}^3 * (\theta_{Max_H_S}^{SW} - \theta_{H_S}^{SW}) + Kd_{H_S}^2 * (\dot{\theta}_{Max_H_S}^{SW} - \dot{\theta}_{H_S}^{SW})$$

2. Extend Knee:

$$\tau_{K_S}^{SW} = Kc_{K_S}^3 * (\theta_{Min_K_S}^{SW} - \theta_{K_S}^{SW}) + Kd_{K_S}^2 * (\dot{\theta}_{Min_K_S}^{SW} - \dot{\theta}_{K_S}^{SW})$$

3. Activate Ankle for keeping feet parallel to ground:

$$\tau_{A_S}^{SW} = Kc_{A_S}^3 * (\theta_{Max_A_S}^{SW} - \theta_{A_S}^{SW}) + Kd_{A_S}^3 * (\dot{\theta}_{Max_A_S}^{SW} - \dot{\theta}_{A_S}^{SW})$$

Whenever Stop transition is validated the robot will start the stopping phase after Swing Leg touches ground

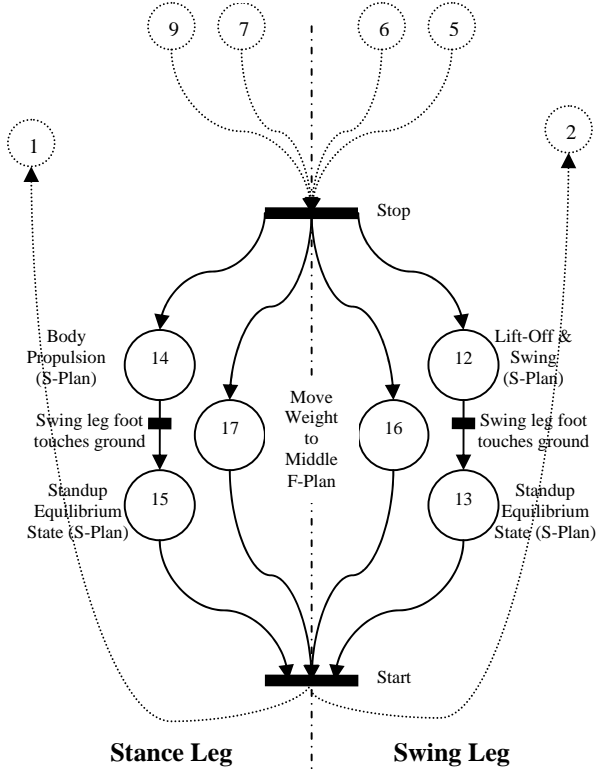


Fig.6. Petri Net Algorithm for Robian Stop Walk

State 14: (Stance Leg – S Plan) “Body Propulsion State”. It starts after the Stop Switch is activated. Here we extend Knee for moving body forward. During this state, three actions happen in parallel:

1. Hip is activated for stabilizing upper body:

$$\tau_{H_S}^{ST} = Kc_{H_S}^5 * (\theta_{Max_S}^{UB} - \theta_{H_S}^{ST}) + Kd_{H_S}^5 * (\dot{\theta}_{Max_S}^{UB} - \dot{\theta}_{H_S}^{ST})$$

2. Extend Knee for moving body forward:

$$\tau_{K_S}^{ST} = Kc_{K_S}^5 * (\theta_{Max_K_S}^{ST} - \theta_{K_S}^{ST}) + Kd_{K_S}^5 * (\dot{\theta}_{Max_K_S}^{ST} - \dot{\theta}_{K_S}^{ST})$$

3. Keep Ankle free:

$$\tau_{A_S}^{ST} = 0$$

State 13: (Swing Leg - S Plan) “Standup Equilibrium State”. It starts after the Swing Leg touches ground. Here we try to memorize the last position values. During this state, three actions happen in parallel:

1. Apply an active torque to Hip for maintaining upper body in a vertical position:

$$\tau_{H_S}^{SW} = Kc_{H_S}^3 * (\theta_{Max_H_S}^{SW} - \theta_{H_S}^{SW}) + Kd_{H_S}^3 * (\dot{\theta}_{Max_H_S}^{SW} - \dot{\theta}_{H_S}^{SW})$$

2. Stabilize Knee:

$$\tau_{K_S}^{SW} = Kc_{K_S}^3 * (\theta_{Min_K_S}^{SW} - \theta_{K_S}^{SW}) + Kd_{K_S}^3 * (\dot{\theta}_{Min_K_S}^{SW} - \dot{\theta}_{K_S}^{SW})$$

3. Stabilize Ankle:

$$\tau_{A_S}^{SW} = Kc_{A_S}^3 * (\theta_{Max_A_S}^{SW} - \theta_{A_S}^{SW}) + Kd_{A_S}^3 * (\dot{\theta}_{Max_A_S}^{SW} - \dot{\theta}_{A_S}^{SW})$$

State 15: (Stance Leg – S Plan) “Standup Equilibrium State”. It starts after the Swing Leg touches ground. Here

we try to memorize the last position values. During this state, three actions happen in parallel:

1. Apply an active torque to Hip for maintaining upper body in a vertical position (here we apply an active torque opposite to that applied and the Swing Leg for stabilizing & counteracting the torque applied on the swing leg):

$$\tau_{H_S}^{ST} = Kc_{H_S}^5 * (\theta_{Max_S}^{UB} - \theta_{H_S}^{ST}) + Kd_{H_S}^5 * (\dot{\theta}_{Max_S}^{UB} - \dot{\theta}_{H_S}^{ST})$$

2. Stabilize Knee:

$$\tau_{K_S}^{ST} = Kc_{K_S}^5 * (\theta_{Max_K_S}^{ST} - \theta_{K_S}^{ST}) + Kd_{K_S}^5 * (\dot{\theta}_{Max_K_S}^{ST} - \dot{\theta}_{K_S}^{ST})$$

3. Stabilize Ankle:

$$\tau_{A_S}^{ST} = 0$$

State 16: (DS – F Plan) Move Weight to Middle position. During this state, two actions happen in parallel:

1. Apply an active torque to Hip for maintaining upper body in a vertical position:

$$\tau_{H_F}^{SW} = Kc_{H_F}^0 * (\theta_{Max_H_F}^{DS} - \theta_{H_F}^{SW}) + Kd_{H_F}^0 * (\dot{\theta}_{Max_H_F}^{DS} - \dot{\theta}_{H_F}^{SW})$$

2. Activate Ankle in order to follow the inverse of the current Hip angle, this will lead to maintaining body parallel to ground:

$$\tau_{A_F}^{SW} = Kc_{A_F}^0 * (-\theta_{H_F}^{SW} - \theta_{A_F}^{SW}) + Kd_{A_F}^0 * (-\dot{\theta}_{H_F}^{SW} - \dot{\theta}_{A_F}^{SW})$$

State 17: (DS – F Plan) Move Weight to Middle position. During this State, two actions happen in parallel:

1. Apply an active torque to Hip for maintaining upper body in a vertical position (here we apply an active torque opposite to that applied and the Swing Leg for stabilizing & counteracting the torque applied on the swing leg):

$$\tau_{H_F}^{ST} = Kc_{H_F}^4 * (\theta_{Max_H_F}^{DS} - \theta_{H_F}^{ST}) + Kd_{H_F}^4 * (\dot{\theta}_{Max_H_F}^{DS} - \dot{\theta}_{H_F}^{ST})$$

2. Activate Ankle in order to follow the inverse of the current Hip angle:

$$\tau_{A_F}^{ST} = Kc_{A_F}^4 * (-\theta_{H_F}^{ST} - \theta_{A_F}^{ST}) + Kd_{A_F}^4 * (-\dot{\theta}_{H_F}^{ST} - \dot{\theta}_{A_F}^{ST})$$

B. Implementation of control Algorithm on simulated ROBIAN biped

We applied our algorithm on a simulated model of our biped robot ROBIAN developed in the LISV Laboratory University of Versailles in France. Because LISV is involved in the Joint Japanese-French Robotics Laboratory (JRL), the simulator used in this work is OpenHRP (Open Architecture Humanoid Robotics Platform), which is a software platform for dynamic simulation of humanoid robots, developed by AIST, the University of Tokyo and MSTC. The OpenHRP model of the ground/foot contact is based on a spring-damper system. In our simulation, the parameters K_p (spring coefficient) and C_p (damper coefficient) are calculated experimentally:

- Spring-damper coefficients parallel to axis [X, Y, Z]:
 $K_p = [10^5, 10^5, 10^5]$ in N/m; $C_p = [10^4, 10^4, 10^4]$ in N/(m/sec);

- Spring-damper coefficients rotational around axis [X,Y,Z]:
 $K_r = [1.610^4, 10^3, 10^3]$ in Nm/rad;
 $C_r = [1.610^3, 10^2, 10^2]$ in Nm/(rad/sec);

ROBIAN foot contains 5 similar plots in rubber. Static and sliding friction factors are: *Static* = 0.6, *sliding* = 0.5.

C. Simulation results

Walking speed and stability is primary influenced by four parameters: $\theta_{Max_S}^{UB}$ affects the robot center of mass position. $\theta_{Max_H_S}^{SW}$, $\theta_{Max_K_S}^{Stance}$ and $\theta_{Min_knee_xy}^{SW}$ affect the step length and body propulsion velocity. So, with a simple variation of those four reference angle values, the step length and hence the walking speeds of the robot are controlled [29]. Walking snapshots from 3D space are represented on Fig. 7. In this simulation we are showing the progression of the walking cycle, from Stand-up right leg in front of left leg to walk to standup left and right leg are in the same position to walking again. Because our aim is to more understand human being locomotion, our biped robot was designed in an anthropomorphic way. Then it could reproduce human gaits with its lower part [18]. This is very important in medical applications, where we need to test different kind of prosthesis on robot before implementing them on humans. Finally, the results obtained rely on the fact that ROBIAN was able to walk infinitely into the simulator without falling [29]. This is due to the continuous control of the PD controllers parameters based on the walking phases, thus based on the joint positions. In order to implement the control algorithm on the real robot ROBIAN, the technological limits of the actuators has been taken into account. First, the power consumption analysis issued from those motors was within the real limits. Then, the value of the motor velocity reduction is changed in order to have the maximum velocity and maximum torque for each motor under the values specified into the datasheet. Stability margin for different referenced angles is summarized in the Table II. In Fig.8 we are showing the joints angle variations for hip, knee and ankle of left and right legs. In Fig.9 we are showing the speed variation of the robot, taking into consideration that the stop phase will start at T=2sec, and finishes at T=8sec.

IV. DISCUSSION AND CONCLUSION

We achieved our goal in making in simulation the ROBIAN robot walk, Stop and then walk again in 3D. We approached the human walking angles without a lot of complexity in the control. ROBIAN foot has a passive and adjustable flexibility, which highlight the necessity of active and flexible feet. Also the feet land parallel to ground, the swinging leg will have the foot moving as human in the beginning of the swing phase. In passive swing phase, ankle motors were controlled in order to keep the foot parallel to ground. In the active swing phase, the foot was stabilized in this position in order to hit the ground in a way so that the ground reaction forces will be distributed equally on the five foot plots. In order to approach the smoothness of human neuromuscular commands, a muscle model created by Serhan et al. [28] is added to ROBIAN low level control stage. This model is based on a PID controller and a DC motor. Hence, it will be easily integrated into our control diagram without making any changes, neither to the underlying hardware system nor to the high level control algorithm (Dynamic Walk Algorithm). This integration shall

add active compliance. Also this technique is not very consuming in processing time (algorithm execution time is about 8ms on a Pentium IV processor running at 3.0 Ghz with 1 GB RAM), so it could be implemented in real-time.

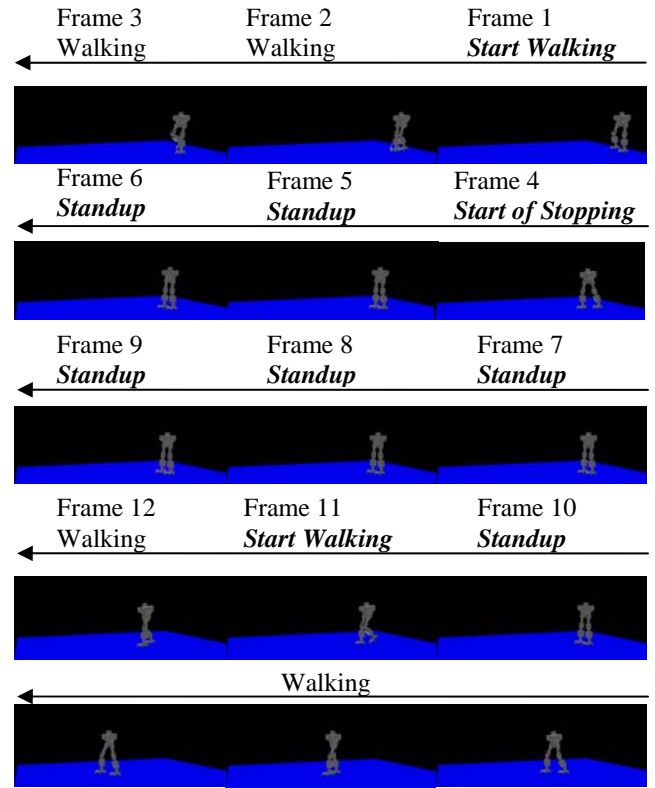


Fig.7. Walking snapshots in 3D for Walk-Stop-Walk cycle. Frames captured from right to left. Time between each snapshot for Frames from 1 to 5 is 1000ms. Time between each snapshot for Frames from 5 to 10 is 1000ms

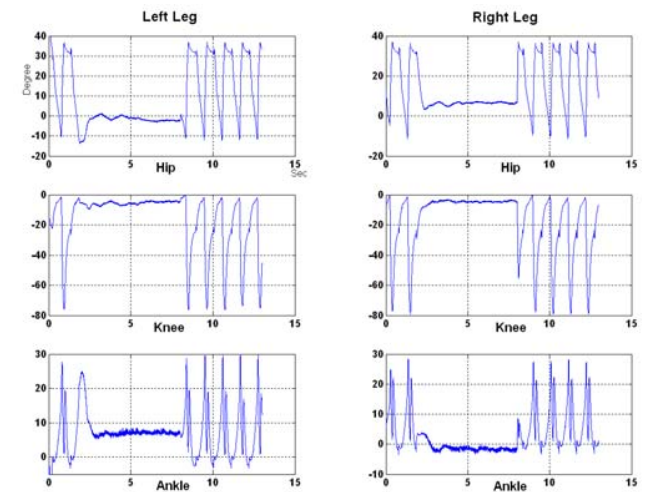


Fig.8. graphs for hip, knee, and ankle Walking-Stop-Walking angles of ROBIAN for two walking cycles at Start, Stop Period (2 to 8 s) and 4 walking cycles after.

This work was a first step in our study. We are working now to validate our approach on the real biped ROBIAN. A next work will concern perturbations effects on robot stability during the walking phase and the stop phase on irregular terrain. The third step concerns implementing our

muscle model [28] in ROBIAN for introducing active compliance. In conclusion, the proposed approach, allows us to achieve Robian dynamic walking and standup similar to human being.

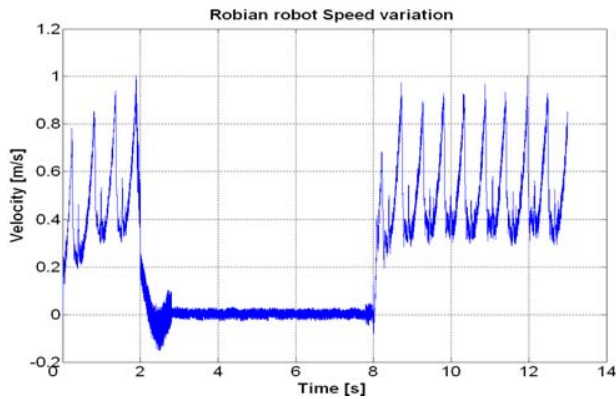


Fig.9. Robot velocity, showing the Walk-Standup-Walk cycles

Table II
Stability Limit Margins

$\theta_{Max_H_S}^{SW}$		$\theta_{Max_H_S}^{ST}$		$\theta_{Min_K_S}^{SW}$	
Acceptable margin	Currently allocated	Acceptable margin	Currently allocated	Acceptable margin	Currently allocated
20° to 40°	30°	-20° to -10°	-15°	-30° to -2°	-4°
$\theta_{Max_K_S}^{SW}$		$\theta_{Max_K_S}^{ST}$		$\theta_{Max_A_S}^{SW}$	
Acceptable margin	Currently allocated	Acceptable margin	Currently allocated	Acceptable margin	Currently allocated
-85° to -55°	-75°	0° to 50°	35°	0° to 50°	35°
$\theta_{Max_S}^{UB}$		$\theta_{Max_H_F}^{DS}$			
Acceptable margin	Currently allocated	Acceptable margin	Currently allocated		
0° to 18°	6°	0° to 10°	4°		

REFERENCES

- [1] G. Taga, Y. Yamaguchi, H. Shimizu, "Self-organized control of bipedal locomotion by neural oscillators in unpredictable environment," *Biological Cybernetics* 65, 1991, 147-159
- [2] G. Endo, J. Morimoto, J. Nakanishi, G. Cheng, "An empirical exploration of a neural oscillator for biped locomotion control," *IEEE, Int. Conference on Robotics and Automation*, 2004, pp. 3036-3042
- [3] T. Matsubara, J. Morimoto, J. Nakanishi, M. Sato, K. Doya, "Learning CPG-based biped locomotion with a policy gradient method," *Robotics and Autonomous Systems* 54, 2006, 911-920
- [4] M. Garcia, A. Chatterjee, A. Ruina, and M. Coleman, "The simplest walking model: Stability, complexity, and scaling," *ASME J. Biomechan. Eng.*, vol. 120, no. 2, pp. 281-288, 1998.
- [5] A. Goswami, B. Espiau, and A. Keramane, "Limit cycles in a passive compass gait and passivity-mimicking control laws," *Autonomous Robots*, vol. 4, no. 3, pp. 273-286, 1997.
- [6] A. Goswami, B. Thuilot, and B. Espiau, "A study of the passive gait of a compass-like biped robot: Symmetry and chaos," *I. J. Robot. Res.*, vol. 17, no. 12, pp. 1282-1301, 1998.
- [7] T. McGreer, "Passive dynamic walking," *Int. J. Robot. Res.*, vol. 9, no. 2, pp. 62-82, 1990.
- [8] C. Sabourin, O. Bruneau, J-G. Fontaine, "Pragmatic rules for real-time control of the dynamic walking of an under-actuated biped robot," *Proc. IEEE Conf. on Rob. and Automation*, 2004, 4216-4221
- [9] H. Ohta, M. Yamakita, and K. Furuta, "From Passive to active dynamic walking," in *Proc. IEEE Conf. Decision Control*, Phoenix, AZ, Dec. 1999, pp. 3883-3885.
- [10] M.W. Spong, "Bipedal locomotion, robot gymnastics, and motor air hockey: A rapprochement," in *Proc. TITech COE/Super Mechano-Systems workshop*, Tokyo, Japan, Feb. 1999, pp. 34-41.

- [11] M.W. Spong, "Passivity based control of the compass gait biped," in *Proc. IFAC Triennial World Congr.*, Beijing, China, vol. 3, July 1999, pp. 19-23.
- [12] A.D. Kuo, "Stabilization of lateral motion in passive dynamic walking," *Int. J. Robot. Res.*, vol. 18, no. 9, pp. 917-930, 1999.
- [13] S.H. Collins, M. Wisse, and A. Ruina, "A three-dimensional passive dynamic walking robot with two legs and knees," *Int. J. Robot. Res.*, vol. 20, 4, no. 3, pp. 273-286, 1997.
- [14] M.W. Spong and F. Bullo, "Controlled symmetries and passive walking," *IEEE trans. Aut. Contr.*, vol. 50, no. 7, pp. 1025-1031, 2005.
- [15] A.D. Ames, R.D. Gregg, E.D.B. Wendel, and S. Sastry, "Towards the geometric reduction of controlled three-dimensional robotic bipedal walkers," in *Proc. Workshop Lagrangian Hamiltonian Methods Nonlinear Control*, Nagoya, Japan, July 2006, pp. 117-124.
- [16] M.W. Spong and G. Bhatia, "Further results on control of the compass gait biped," in *Proc. IROS 2003*, Las Vegas, Nevada, 27-30 Oct., 2003, pp. 1933-1938.
- [17] A. Konno, R. Sellaoui, F.B. Amar, and F.B. Ouezdou, "Design and development of the biped prototype ROBIAN," in *IEEE Int. Conf. on Rob. and Aut. (ICRA)*, D.C., U.S.A., 2002, pp. 1384-1389.
- [18] R. Sellaoui and F. B. Ouezdou, "Design and control of a 3DOFs parallel actuated mechanism for biped application," *Mechanism and Machine Theory*, Volume 40, N°12, pages 1367-1393, decembre 2005.
- [19] C. Chevallereau, Y. Aoustin, "Optimal reference trajectories for walking and running of a biped robot," *Robotica* 19 (2001), pp. 557-569.
- [20] Z. Tang, Z. Sun, C. Zhou and L. Hu, "Reference Trajectory Generation for 3-Dimensional Walking of a Humanoid Robot" *Tsinghua Science & Technology* Volume 12, Issue 5, October 2007, Pages 577-584
- [21] E. Viel, "La marche humaine, la course et le saut," Masson, 2000.
- [22] S. Bouisset, B. Maton, "Muscles, posture et mouvement," Hermann, 1995.
- [23] P. Allard, J.-P. Blanchi, "Analyse du mouvement humain par la biomécanique," 2000, Décarie.
- [24] D. Winter, "Biomechanics and Motor Control of Human Movement," Third Edition, Wiley 2005.
- [25] Bouchet A., Cuilleret J., "Anatomie topographique, descriptive et fonctionnelle", T3b le membre inférieur. SIMEP 3rd ed. 1997
- [26] Kapandji I., "Physiologie articulaire, tome Membre inférieur", Maloine, 5th edition, 1999
- [27] C. Williams, "Tuning a PID Temperature Controller," <http://newton.ex.ac.uk/teaching/CDHW/Feedback/Setup-PID.html>.
- [28] Serhan H., Nasr C., Henaff P., "Designing a Muscle Like System Based on PID Controller and Tuned by Neural Network", *IEEE WCCI Congress*, Vancouver, 2006, pp. 10090-10097
- [29] Serhan H., Nasr C., Henaff P., BenOuezdou F., "A New Control Strategy for ROBIAN Biped Robot Inspired from Human Walking", to appear in *Proc. IEEE of IROS 2008*, Nice.

- 9.6 Flexible and Stable Pattern Generation by Evolving Constrained Plastic Neurocontrollers. Adaptive Behavior** , sous presse 2011

Adaptive Behavior

<http://adb.sagepub.com/>

Flexible and multistable pattern generation by evolving constrained plastic neurocontrollers

Thierry Hoinville, Cecilia Tapia Siles and Patrick Hénaff

Adaptive Behavior published online 16 May 2011

DOI: 10.1177/1059712311403631

The online version of this article can be found at:

<http://adb.sagepub.com/content/early/2011/05/13/1059712311403631>

Published by:



<http://www.sagepublications.com>

On behalf of:

ISAB

International Society of Adaptive Behavior

Additional services and information for *Adaptive Behavior* can be found at:

Email Alerts: <http://adb.sagepub.com/cgi/alerts>

Subscriptions: <http://adb.sagepub.com/subscriptions>

Reprints: <http://www.sagepub.com/journalsReprints.nav>

Permissions: <http://www.sagepub.com/journalsPermissions.nav>

Flexible and multistable pattern generation by evolving constrained plastic neurocontrollers

Adaptive Behavior
0(00) 1–21
© The Author(s) 2011
Reprints and permissions:
sagepub.co.uk/journalsPermissions.nav
DOI: 10.1177/1059712311403631
adb.sagepub.com



Thierry Hoinville¹, Cecilia Tapia Siles¹ and Patrick Hénaff²

Abstract

In evolutionary robotics, plastic neural network models proved to be promising for evolving adaptive behaviors. In particular, neurocontrollers incorporating hebbian synapses have been shown to be useful for implementing conflicting sub-behaviors. Numerous interesting complex tasks assume such flexibility. However, those evolved controllers often exhibit behavioral instability, as simulation time is extended beyond the short limit used during evolution. In this paper, we propose constrained plastic models inspired by neural homeostasis phenomena, in order to evolve flexible and stable pattern generators for single-legged locomotion. Comparative results show that constrained controllers perform better than unconstrained ones in both terms of evolvability and behavioral stability. Functional analyses of the best evolved controller unveil the adaptivity, robustness and homeostasis arising from the statically constrained plasticity. Interestingly, homeostasis evolved implicitly without relying on any active homeostatic mechanisms and is implemented through hebbian plasticity, usually considered destabilizing.

Keywords

Homeostasis, plastic neural networks, adaptive synapses, legged locomotion, evolutionary robotics

1. Introduction

A key issue of evolutionary robotics is to seek convenient control building blocks for specific behaviors (Harvey, Husbands, Cliff, Thompson, & Jakobi, 1996; Jakobi & Quinn, 1998; Floreano & Urzelai, 2000). In particular, many efforts have been carried out in exploring new neural network models intended to be more evolvable, scalable and adaptive. Among them, Continuous Time Recurrent Neural Networks (CTRNNs) (Beer, 1995), GasNets (Husbands et al., 1998) and Plastic Neural Networks (PNNs) (Floreano & Mondada, 1996) received most attention. Comparing, hybridizing and refining these models, as initiated by McHale and Husbands (2004a, 2004b), turn out to be essential for better defining their areas of applicability (Bullock, 2006). Ultimately, it could allow us to predict which neural networks are suitable to evolve which behaviors. Here, we propose to study particular refinements applied to a CTRNN/PNN hybrid model.

The standard PNN model involves hebbian learning rules used to make each synapse adaptive. It was first introduced by Floreano and Mondada (1996) to explore evolution of plasticity, rather than evolution and plasticity. One of their later results (Urzelai & Floreano, 2001) was to show, through a Khepera performing a light-switching task, that evolved controllers embedding adaptive synapses can sequentially exhibit multiple possibly conflicting sub-behaviors. Such behavioral versatility constitutes a desired trait of autonomy and adaptivity. This contrasts with the monotonic and minimalistic behaviors usually obtained

¹Italian Institute of Technology, Genoa, Italy

²ETIS, University of Cergy-Pontoise, France

Corresponding author:

Dr. Thierry Hoinville Italian Institute of Technology Via Morego,
30 – 16163 Genoa Italy
Email: thierry.hoinville@iit.it

in prior works (Urzelai & Floreano, 2001; Doncieux, 2003). Subsequently, on evolving conflicting phototactic behaviors, Di Paolo (2003) obtained a similar result using another kind of adaptive-synapse controllers, based on spike-timing dependent plasticity (STDP).

However, PNNs face also several shortcomings. For example, Stanley, Bryant, and Miikkulainen (2003) reported that PNNs evolved slower than fixed-weight networks, for a dangerous food foraging task. Furthermore, Tuci and Quinn (2003) were unable to evolve PNNs to solve a learning task. This appears very surprising when one considers the many links observed in nature between hebbian plasticity and learning. Finally, for our concern in this paper, there are several evidences that PNN-controlled robots are inherently subject to some behavioral instability. First, none of the PNNs evolved by Urzelai and Floreano (2001) have retained permanently the desired light-switching behavior. That is, these controllers were unable to lead the Khepera to put the light on more than once, i.e. more than required during evolution. Second, PNNs evolved by McHale and Husbands (2004a, 2004b) for quadrupedal and bipedal locomotions tend to generate unstable gaits, as simulation time is extended beyond the limit used during evolution. We suggest that this last issue is an instance of the flexibility/stability dilemma. In other words, one can wonder how to evolve robot controllers that are plastic – thus providing some flexibility – and yet bearing behavioral stability?

In (Hoinville & Hénaff, 2004a, 2004b), we proposed to take inspiration from neurophysiological observations, which suggest that neurons counteract the destabilizing influence of hebbian plasticity using several homeostatic plasticity mechanisms (Turrigiano & Nelson, 2000). Practically, we have shown, for a single-legged locomotion task, that CTRNN/PNN hybrid models constrained by static rules inspired by homeostatic mechanisms are more evolvable and behaviorally stable than unconstrained ones. In this paper, we detail these results and extend them by performing functional analyses of the best evolved controller, so as to uncover the actual roles of the so-called homeostatic constraints we proposed.

This paper is organized as follows. The next section gives, first, the biological background that inspired our work and second, a state of the art of evolving homeostatic neurocontrollers for robots. The third section describes the neuronal models we compared in evolving single-legged locomotion controllers. Results are presented in the fourth section and discussed later in the fifth section. Then, we conclude in the last section by giving some perspectives.

2. Background

In this section, we draw the biological background that inspired our work and review how neuronal homeostasis has been addressed in evolutionary robotics.

2.1. Homeostasis in neurophysiology

Hebbian plasticity is known as a ubiquitous phenomenon involved in major cognitive and adaptive functions, like learning and memory. Moreover, it plays a key role in refining synaptic connections during brain development. Thus, by reinforcing correlation between simultaneously active neurons, hebbian plastic rules participate in the significant flexibility of nervous systems. However, raw modeling of this principle comes with serious instability issues, since reinforcing neuronal correlations increases likeliness of simultaneous activations and so on. Early modeling studies yielded many mathematical adjustments intended to cope with synaptic saturation impacting the basic Hebb's rule (Miller & MacKay, 1994). Some experimental observations supported more plausible advanced auto-stabilizing hebbian rules, like the BCM law (Bienenstock, Cooper, & Munro, 1982) or some forms of STDP (Kepecs, van Rossum, Song, & Tegnér, 2002).

A more general perspective has emerged around the notion of neuronal homeostasis. Homeostasis refers to all phenomena that contribute to preserve some entity's internal physiological state (e.g. temperature) from internal or external disturbances (e.g. weather) (Kitano, 2007). Applied to neurons, this notion can be defined as the capacity, of a neuron or a network, to maintain a characteristic activity (i.e. state).

In the following sections, we will focus on two homeostatic phenomena, namely excitability regulation and synaptic scaling, which have inspired the present work. These are believed to take a great part in counteracting the destabilizing effect of hebbian plasticity on neuronal activity (Turrigiano & Nelson, 2000). The first one operates on intrinsic properties of neurons for regulating their excitability. The second one occurs at synaptic level to regulate the total synaptic input strength of each neuron.

2.1.1. Excitability regulation. The excitability of a neuron can be defined as its probability to fire action potentials according to input signal levels it receives. Neuronal excitability can be illustrated by a sigmoidal curve that relates the total synaptic drive received in input to the mean firing rate obtained in output (Figure 1). According to Turrigiano and Nelson (2000), excitability regulation can be described as a process that shifts this curve to maintain a long term

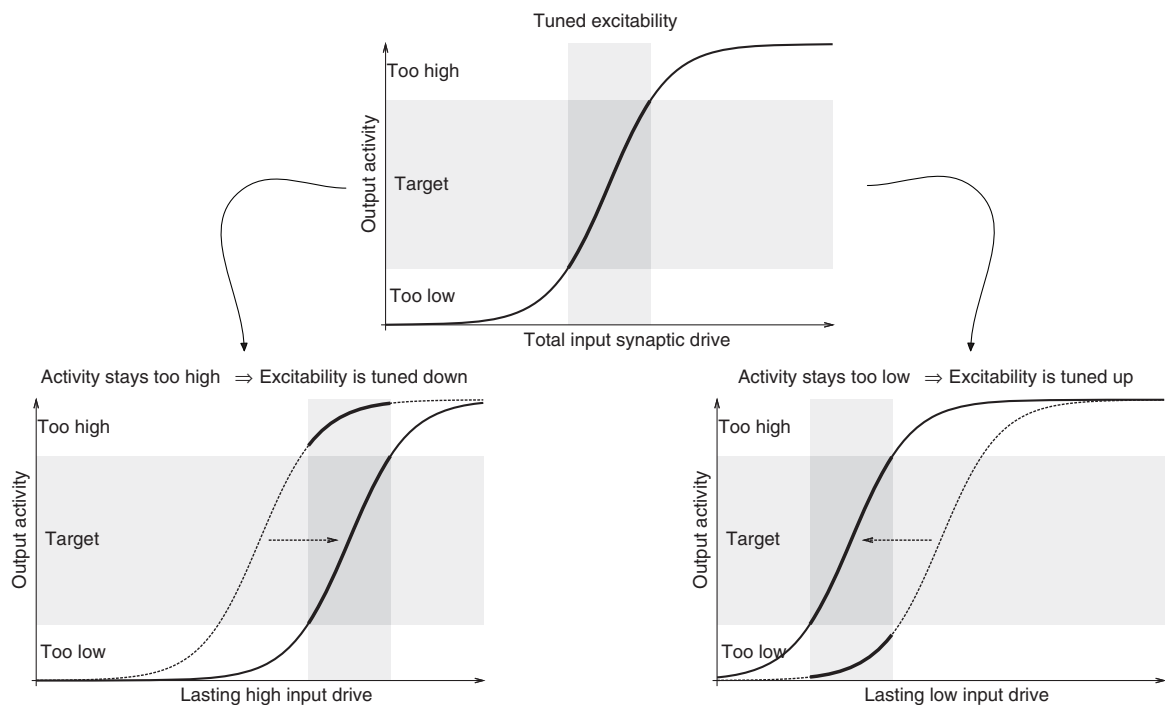


Figure 1. Homeostatic regulation of neuronal excitability. Typically, for a neuron, a sigmoid activation function can be used to relate the total synaptic drive received as input to the mean firing rate (i.e. activity) obtained as output. Depending on the range of synaptic drive received (vertical gray strips), only a part of the activation function is used (darkened curve parts). If this part corresponds to too high (left example) or too low (right example) neuronal activity, the activation function is shifted (i.e. excitability is tuned) in order to get back to a target activity level (horizontal gray strips).

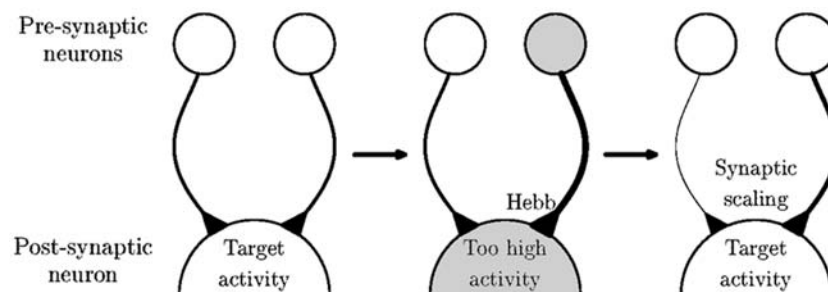


Figure 2. Interaction between hebbian plasticity and synaptic scaling. One post-synaptic neuron is receiving excitatory inputs from two pre-synaptic neurons. When one of the latter fires enough for activating the post-synaptic neuron, the synapse involved is potentiated via hebbian plasticity, resulting in higher post-synaptic activity and so on. If this too high post-synaptic activity lasts too long, then synaptic scaling acts on all input synapses proportionally to their weights in order to get back to a normal activity. It leads to synaptic competition in that one potentiated synapse induces depression of the other.

“normal” activity. When input excitation induces too high firing rate for a long time, excitability is decreased (left example on Figure 1). Conversely, if input excitation is not sufficient, thus making the considered cell silent for a long while, then its excitability is increased (right example on Figure 1).

2.1.2. Synaptic scaling. This homeostatic phenomenon consists of scaling the global synaptic input gain

of a neuron, that counteracts prolonged perturbation of its activity (Figure 2). Thus, when activity remains too low for a long while, all excitatory inputs are potentiated and all inhibitory inputs are depressed. Opposite modifications are expressed when lasting over-activity occurs. Moreover, as reported by Turrigiano and Nelson (2000), synaptic scaling appears to be multiplicative, i.e. input synapses are potentiated or depressed in proportion to their strength. As such, synaptic

scaling has been shown to induce synaptic competition (Figure 2) and to overlay hebbian plasticity in a very complementary way (Turrigiano & Nelson, 2000).

2.2. Neuronal homeostasis in evolutionary robotics

2.2.1. Models and main results. Evolution of homeostatic neurocontrollers has been mostly performed by Di Paolo and his colleagues. The first model proposed in (Di Paolo, 2000) implements homeostasis through conditional activation of various hebbian plasticity rules. Some activation bounds are defined so as to divide neuronal activation in homeostatic and non-homeostatic ranges. Whenever any neuron fires out of the homeostatic bounds, genetically-specified hebbian laws are triggered on all its input synapses in order to recover a normal activity. This model has been used to evolve controllers for simulated Braitenberg-like agents equipped with two actuated wheels and two light sensors. The targeted task was to reach light sources in a 2D infinite space (this behavior will be referred in the following as “phototaxis”).

Then, this model has been slightly modified in further studies (two homeostatic ranges instead of one; other hebbian plasticity formulations). These ones still involved phototactic tasks, but following more complex scenarios designed to address some questions about cognition:

- modeling perseverative reaching in infants (Wood & Di Paolo, 2007);
- minimal dynamics for behavioral preference (Iizuka & Di Paolo, 2007);
- linkage between internal and behavioral stability (Iizuka & Di Paolo, 2008).

In another work aiming for simpler phototactic behavior, Di Paolo (2002a) proposed a model of “homeostatic oscillators” based on standard CTRNNs incorporating a fast mechanism of excitability regulation. This one consists of turning the bias of each neuron into a dynamic variable, which acts to keep constant its average firing rate to a given homeostatic value. Because of the fast speed of this plasticity, neurons tend to oscillate.

Di Paolo (2002b) also considered a more complex model based on spiking neurons, where each synapse is subject to both spike-timing dependent plasticity (STDP) and synaptic scaling. Basically, STDP drives synaptic weight changes according to the time difference between pre- and post-synaptic potentials. Among the compared models, one implements in addition to STDP a mechanism of synaptic scaling. Spike trains of each neuron are leaky integrated to estimate

the current firing rate which is then compared to a given reference firing rate. The difference is minimized by scaling up or down the weights of afferent synapses, depending on whether they are excitatory or inhibitory. Here again, the controllers based on these mechanisms were evolved to satisfy simple phototaxis.

All the above homeostatic models have shown improved evolvability compared to the control models, in term of both final scores and evolving speeds. The resulting neurocontrollers performed phototactic tasks more reliably and adaptively. For example, some proved to be robust to neural noise or randomization of initial state. Most of them were even able to adapt to strong sensorimotor disruptions never seen during evolution, like motor distortions, sensor swapping or removal (Di Paolo, 2000, 2002a, 2002b; Iizuka & Di Paolo, 2008). Besides, it appears that such homeostatic models provide some behavioral stability. In fact, except in (Di Paolo, 2000, 2002b), the obtained behaviors showed extended durability or readaptability (i.e. recovering from cancelling the sensorimotor perturbations).

In contrast, Williams (2005, 2006) reported poor evolvability results using another homeostatic network model. This one consists of a CTRNN model endowed with an excitability regulation and a synaptic scaling mechanism. Like in (Di Paolo, 2002a), excitability regulation is implemented by an adaptive bias. However, the dynamics involved here are slower and the plasticity just operates when neuronal activity goes outside a homeostatic range (i.e. activity is not constrained to a unique value as in Di Paolo’s model). In the same way, if a neuron fires outside this given homeostatic range, all its afferent synapses are multiplied down or up, depending on their sign and in way to return the neuron to a normal firing activity.

This model was assumed to be more evolvable than the classic CTRNNs because of the network-level properties it provides: increased sensitivity, better signal propagation and increased likelihood of oscillations (Williams & Noble, 2007). Nevertheless, evolution attempts made by Williams (2005, 2006) on a task of 1D catching falling objects (with or without shape discrimination) showed that homeostatic plastic CTRNNs were generally outperformed by non-plastic CTRNNs, although the formers evolved quicker and with more consistency. Williams (2005, 2006) showed however that such homeostatic plasticity, when used as developmental mechanism rather than lifetime plasticity, sometimes led to more evolvable controllers than non pre-developed ones.

2.2.2. Discussion. While enforcing the idea that neuronal homeostasis promotes evolution of adaptive

behaviors, the previous studies can be discussed in relation to their methods.

Firstly, in order to fit with short-duration simulations required for affordable fitness computation, all the modeled homeostatic mechanisms have been parameterized such as to act much faster than the inspiring biological processes. Although most authors ensured to significantly separate timescales of neuronal activation and homeostatic plasticity, we think that this approximation tends to denaturate the given mechanisms and lowers biological relevance of the results. For example, in (Di Paolo, 2002a) and (Di Paolo, 2002b), fast excitability regulation gives rise to rapid rhythmic neuronal activity. It hardly relates to nature considering the slow dynamics of the real mechanism, yet this function is useful here and evolutionary robotics in general may not be adapted to achieve biomimetic results.

Secondly, except in (Di Paolo, 2002b), homeostatic plasticity is envisaged as a solution to implement short-term behavioral adaptivity or robustness (which again is unrealistic given its slowness), rather than as a stabilizing counterpart to other plasticities, like it has been argued for the hebbian one (Turrigiano & Nelson, 2000).

Thirdly, the phototactic tasks may inherently favor the homeostatic models. In fact, even randomly generated homeostatic CTRNNs (i.e. before any evolution is applied) were shown in significant proportion to perform phototaxis when embodied in Braitenberg agents (Williams, 2004, 2006).

3. Methods

3.1. Models

All the models we have reviewed involve homeostatic plastic mechanisms that dynamically monitor neuronal activity and trigger some parameter changes to recover homeostasis when needed. In contrast, we propose to follow an approach based on static mathematical constraints intended to decrease the likelihood of neuronal activity being saturated or silent. These “homeostatic constraints” restrict the dynamical neural networks to parameter subspaces similar to those asymptotically reached using homeostatic plasticity. This keeps unchanged the dimension of the dynamics space (i.e. no new state variables) and discards the issue of slow timescale dynamics. We expect this would encourage evolutionary search to focus on richer dynamics (Mathayomchan & Beer, 2002) which also show proper behavioral stability, although in the presence of disturbing plasticity such as the hebbian one.

Furthermore, an open question is whether this approach could implicitly lead to evolve some homeostasis.

In particular, our present work investigates the impact of two specific homeostatic constraints on the evolvability, behavioral stability and the internal dynamics of CTRNN/PNN hybrid controllers. The first one mimics the outcome of excitability regulation. The second one is inspired from synaptic scaling.

3.1.1. CTRNN/PNN hybrid model. Firstly proposed in (McHale & Husbands, 2004a, 2004b) and (Hoinville & Hénaff, 2004a, 2004b), this model combines the leaky integrator neurons of CTRNNs with the hebbian synapses of PNNs.

Adapted from (Beer, 1995), neurons are modeled by

$$y_i^{t+\Delta t} = y_i^t + \frac{\Delta t}{\tau_{mi}} \left(-y_i^t + k_i \sum_{j=1}^N \pm_{ji} w_{ji}^t o_j^t + I_i^t \right) \quad (1)$$

$$i = 1, 2, \dots, N$$

where $y_i^{t+\Delta t}$ and y_i^t are the i th node's states at respectively times $t + \Delta t$ and t ; Δt is the simulation time slice (set to 10 ms); τ_{mi} is the time constant of the i th node; $k_i = 1/(\sum_{j=1}^N |\pm_{ji}|)$ is a normalizing factor; \pm_{ji} is the sign of the connection from the j th to the i th nodes (1: excitatory, -1: inhibitory, 0: off); $w_{ji}^t \in [0, 1]$ is the corresponding connection strength which is subject to plastic changes; I_i^t is the input value in case of a sensory neuron, otherwise it is set to 0; o_j^t is the output of the j th node computed through a sigmoid activation function, such as

$$o_i^t = \frac{1}{1 + \exp(-g_i(y_i^t + \theta_i))} \quad (2)$$

with g_i and θ_i are respectively the i th node's gain and bias.

According to the PNN model (Floreano & Mondada, 1996), connection weights are updated through time as follows:

$$w_{ji}^{t+\Delta t} = w_{ji}^t + \frac{\Delta t}{\tau_{sji}} \Delta w_{ji}^t \quad (3)$$

where τ_{sji} is the plasticity time constant (the equivalent learning rate is $\eta_{ji} = \Delta t/\tau_{sji}$) of the connection from the j th to the i th nodes and Δw_{ji}^t is one of the following four hebbian rules:

1. Plain Hebb rule

$$\Delta w_{ji}^t = (1 - w_{ji}^t) o_j^t o_i^t \quad (4)$$

2. Post-synaptic rule

$$\Delta w_{ji}^t = w_{ji}^t (-1 + o_j^t) o_i^t + (1 - w_{ji}^t) o_j^t o_i^t \quad (5)$$

3. Pre-synaptic rule

$$\Delta w_{ji}^t = w_{ji}^t o_j^t (-1 + o_i^t) + (1 - w_{ji}^t) o_j^t o_i^t \quad (6)$$

4. Covariance rule

$$\Delta w_{ji}^t = \begin{cases} (1 - w_{ji}^t) \mathcal{F}(o_j^t, o_i^t) & \text{si } \mathcal{F}(o_j^t, o_i^t) > 0 \\ w_{ji}^t \mathcal{F}(o_j^t, o_i^t) & \text{otherwise} \end{cases} \quad (7)$$

where the function $\mathcal{F}(o_j^t, o_i^t) = \tanh[4(1 - |o_j^t - o_i^t|) - 2]$ measures the similarity between pre- and post-synaptic activities.

3.1.2. Excitability regulation by center-crossing constraint. Homeostatic regulation of neuronal excitability can be achieved by providing the sigmoid activation function with an adaptive bias. Such a solution has been carried out in (Di Paolo, 2002a; Williams & Noble, 2007). Moreover, there is evidence that this mechanism tends to maintain CTRNNs close to the so-called ‘‘center-crossing position’’ of the parameter space (Williams & Noble, 2007; Williams, 2006).

Center-crossing CTRNNs are defined as satisfying a special condition on neuronal bias given by

$$\theta_i^* = \frac{-\sum_{j=1}^N \pm_{ji} w_{ji}^t}{2} \quad (8)$$

which insures that the operating range of each neuron is centered on the most sensitive region of its activation function (Mathayomchan & Beer, 2002).

Here, we propose to use this condition to constrain the neurons, rather than possibly reach homeostatic states through dynamic implementation of excitability regulation. However, setting θ_i to θ_i^* is quite restrictive and reduces the number of parameters. Therefore, to keep the parameterization unchanged between our models and let evolution seek the nearby center-crossing condition, we chose the more flexible solution of correcting the activation function asymmetry:

$$y_i^{t+\Delta t} = y_i^t + \frac{\Delta t}{\tau_{mi}} \left(-y_i^t + k_i \sum_{j=1}^N \pm_{ji} w_{ji}^t (2o_j^t - 1) + I_i^t \right) \quad (9)$$

Indeed, for symmetric odd activation functions such as $(2o_j^t - 1)$, the center-crossing condition is verified for $\theta_i^{**} = 0$.¹ So, in this model, by both keeping evolving θ_i and symmetrizing the activation function (yet just for neuronal summation, not for output), we center the evolutionary search around the center-crossing condition and keep our models homogeneous.

3.1.3. Synaptic scaling by normalization constraint. As we have seen before, synaptic scaling regulates global input gain of neurons to maintain a typical activity. Besides, it can lead to synaptic competition that properly complements hebbian plasticity.

Normalization constraints have been proposed in the past (Miller & MacKay, 1994) for modeling synaptic competition that takes place notably during ocular dominance column development.² The principle is to keep either $\sum_{j=1}^N \pm_{ji} w_{ji}^t$ or $\sum_{j=1}^N (w_{ji}^t)^2$ constant, the latter giving usually more uniform synaptic distributions (Miller & MacKay, 1994).

We propose to use $\sum_{j=1}^N (w_{ji}^t)^2 = 1$. Indeed, we assume that constraining synaptic input gains to unity would balance information propagation throughout neurons and so would promote maintaining meaningful neuronal activity, i.e. not saturating, neither silent.

To be consistent with synaptic scaling observations, we implemented the normalization constraint in a multiplicative way, as follow:

$$w_{ji}^{t+\Delta t} = \frac{w_{ji}^t + \frac{\Delta t}{\tau_{sji}} \Delta w_{ji}^t}{\sqrt{\sum_{k=1}^N \left(w_{ki}^t + \frac{\Delta t}{\tau_{ski}} w_{ki}^t \right)^2}} \quad (10)$$

and

$$y_i^{t+\Delta t} = y_i^t + \frac{\Delta t}{\tau_{mi}} \left(-y_i^t + \sqrt{k_i} \sum_{j=1}^N \pm_{ji} w_{ji}^t o_j^t + I_i^t \right) \quad (11)$$

3.1.4. Summary. It is important to note that, contrary to the usual approach, we do not introduce any new dynamics for enforcing homeostasis. That is, the neuronal activity is not monitored and there is no mechanism that dynamically corrects any parameter. In fact, homeostasis is not ensured to be maintained in the short- or long-term. However, our hypothesis is that the chosen constraints would make homeostasis more likely and lead to evolve plastic controllers implementing stable behaviors.

To evaluate this hypothesis, we compared four neuronal models instantiated from the two homeostatic constraints defined above:

- CTRL, the plain CTRNN/PNN hybrid model (our control model);
- CC, the CTRNN/PNN model with center-crossing neurons;
- NS, the CTRNN/PNN model with normalized adaptive synapses;
- CCNS, the CTRNN/PNN model with both center-crossing neurons and normalized adaptive synapses.

3.2. Application

We compared the performances of these four models as substrates for evolving controllers for a single-legged locomotion task. Such a test bed is preferred to the

phototactic tasks usually found in related works for two main assumptions.

Firstly, as we pointed out in the background section, phototactic tasks might inherently suit to homeostatic networks, as the success of random homeostatic CTRNNs in (Williams, 2004, 2006) tends to show. Although adaptation to unseen perturbation like sensory swapping (Di Paolo, 2000) requires non trivial dynamics, phototaxis of Braitenberg-like agents can rely at low level on basic sensory-motor reactions. On the contrary, even involving a single leg, locomotion requires generating rhythmic pattern, possibly from scratch, for coordinating several degrees of freedom. Possible control strategies are multiple since they rely on adapting any combination of frequencies, amplitudes and phases.

Secondly, considering homeostasis, maintaining a “normal” oscillatory activity underlying some locomotor behavior appears challenging. Defining simple homeostatic activity ranges, as usually proposed, would not be enough to keep stable a specific neuronal activity pattern. To be effective, active homeostatic mechanisms would thus require to monitor several oscillatory signal properties such as phases and frequencies. Here, our approach based on homeostatic parameter constraints avoids this issue.

3.2.1. Single-legged robot. The simulated robot² we used consists in a cuboid base supporting a leg articulated about three degrees of freedom (Figure 3). The hip and knee are universal and hinge joints respectively, allowing protraction/retraction, elevation/depression and extension/flexion of the leg.

In addition, a linear joint between the base and the ground constrains the robot to move along a horizontal straight line (Y -axis, 12 cm high). This rail generates viscous friction that brakes the robot sliding. The friction force $F_{fr}(t)$ is proportional to the base’s instantaneous speed $V(t)$, such as

$$F_{fr}(t) = -k_{fr}V(t) \tag{12}$$

where k_{fr} is the viscous friction coefficient.

3.2.2. Fitness. The task of the robot is to respect an instantaneous desired velocity $V_d(t)$. In addition, the robot must cope with an external perturbation that consists in sudden increase of the friction coefficient k_{fr} .

In order to evolve controllers satisfying this behavior, we defined a fitness based on three evaluation scenarios. As shown in Table 1, each scenario is defined by time-varying profiles of the desired speed $V_d(t)$ and the friction coefficient $k_{fr}(t)$:

- the scenario A aims to follow a triangular desired speed profile;

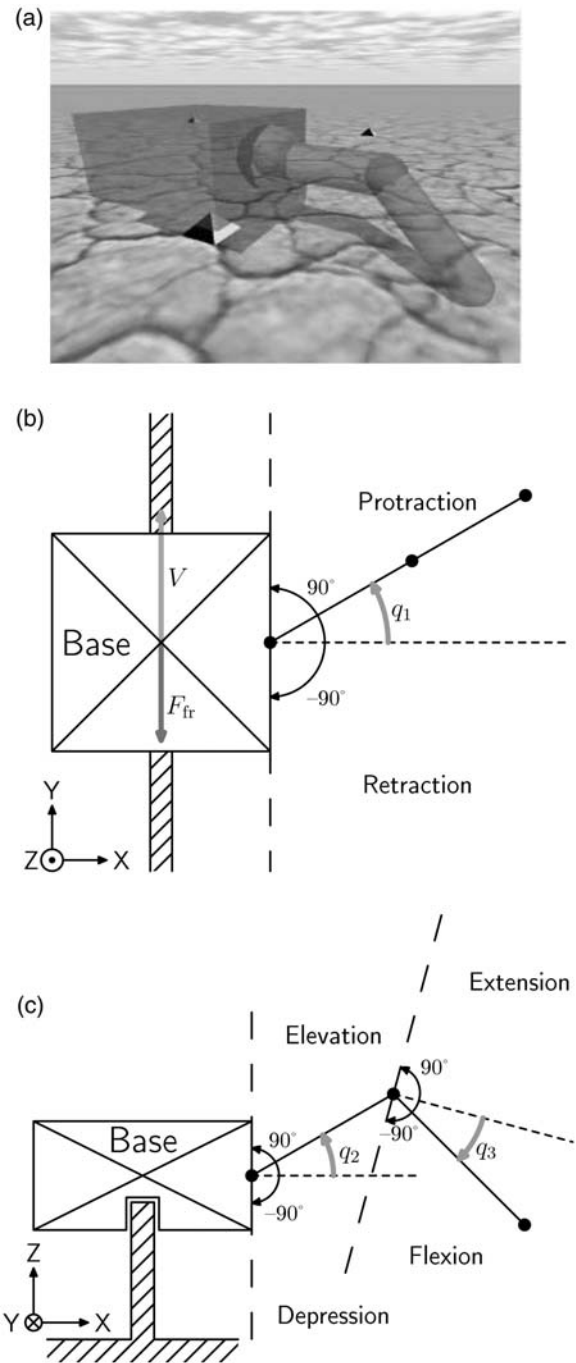
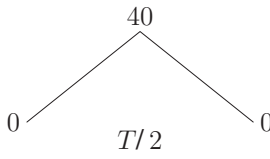
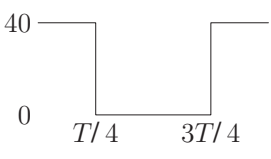
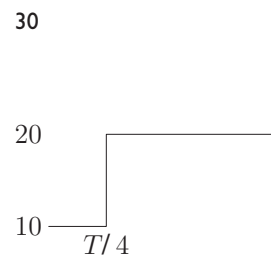
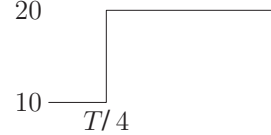


Figure 3. The single-legged robot model. The base is a box of $20 \times 20 \times 10$ cm and 3 kg. The leg segments are cylinders of $15 \times \varnothing 4$ cm and 0.5 kg. In the initial configuration (i.e. $q_1, q_2, q_3 = 0$), the thigh is parallel to the ground and the tibia is bent to 45° toward the ground. From this configuration, all rotation ranges are $\pm 90^\circ$. (a) Simulation (b) Top view (c) Back view.

Table 1. The three evaluation scenarios ($T = 10$ s)

	Scenario A	Scenario B	Scenario C
$V_d(t)$ [cm/s]			
$k_{fr}(t)$ [kg/s]	10	10	

- the scenario B rewards the capacity to inhibit the locomotion behavior;
- the scenario C promotes adapting to the perturbation.

Each scenario involves simulating $T = 10$ seconds of real time (i.e. 1000 timesteps), during which the mean error E between the desired velocity $V_d(t)$ and the instantaneous mean velocity $V_m(t)$ is computed according to

$$E = \frac{1}{T} \int_0^T |V_d(t) - V_m(t)| dt \quad (13)$$

where $V_m(t)$ is measured through applying a second order low-pass filter on the raw velocity $V(t)$. This filtering discards the high amplitude variations of $V(t)$ occurring at each stride of the robot.

From the mean errors E_A , E_B , E_C obtained in the three scenarios, the fitness of a controller, which should be minimized, is given by:

$$fitness = \sqrt{E_A^2 + E_B^2 + E_C^2} \quad (14)$$

This quadratic combination encourages minimizing error in all the three scenarios by reducing the local optima corresponding to solving one or two scenarios only.

This fitness function has been built empirically by several evolution trials. Gathering these three scenarios allows to reward controllable and generalizing closed-loop locomotor behaviors. Note that, since each scenario lasts for a short time of ten seconds, the fitness function does not reward for long-term behavioral stability.

3.3. Controllers architecture and genetic encoding

For all the four models, we used the same control architecture of eight neurons (Figure 4). As we will describe further, the network topology was encoded in the genotype of each controller. We only kept fixed the connections with sensors and actuators. The neurons no. 1 and 2 are sensorineurons. The first is fed by the instantaneous error $e(t) = V_d(t) - V_m(t)$ between the desired velocity and the current average velocity. The second neuron receives boolean value from a ground contact sensor at the tip of the leg. Activations of the neurons no. 6, 7 and 8 give, after linear scaling, the angles of the three articulations³. The three remaining neurons are hidden interneurons.

As our controller models share the same parameterization and architecture, we used a common genetic encoding scheme (Figure 5). Each genotype is composed of eight neuron blocks. Each neuron block i includes the intrinsic properties (i.e. τ_{mi} , g_i and θ_i) of the coded neuron and a list of eight synapse blocks. These are coding for the neuron inputs (including the connection from itself). Each synapse block j of the neuron block i codes: whether the connection exists or not (thereby it defines the network topology), the synapse sign \pm_{ji} , one of the four hebbian rules and the plasticity time constant τ_{sji} .

In total, each genotype consists in 280 genes. Each numerical gene can take one value among five (Table 2). Hence, there are 3.74×10^{138} possible genotypes in the search space.

3.4. Genetic algorithm

We evolved the controllers using a classical generational and elitist genetic algorithm (Goldberg, 1989). That is, populations were initially randomly generated and

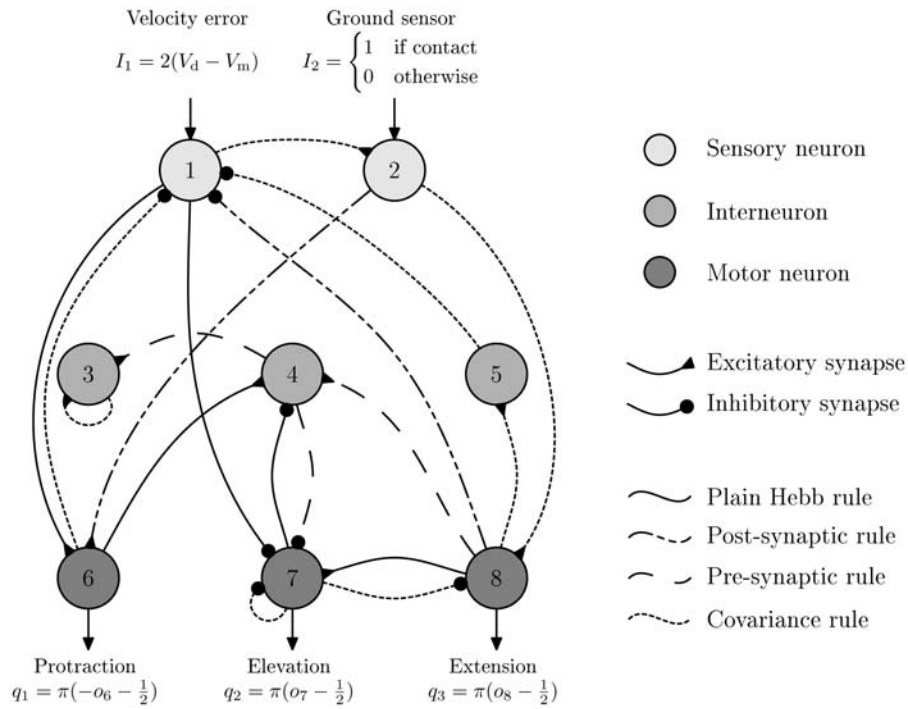


Figure 4. Evolved controller phenotype (corresponding to the best CCNS controller later analyzed).

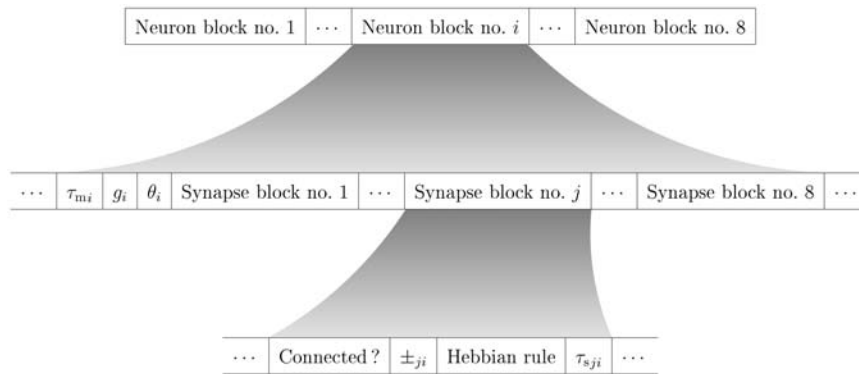


Figure 5. Genetic encoding scheme.

entirely renewed at each generation (except the best performing individual). To avoid premature convergence, we used stochastic universal sampling (Baker, 1987) and linear ranking selection with selective pressure of 1.1 (Goldberg, 1989). New individuals were generated using two genetic operators: simple allelic mutation ($P_{mut} = 0.001$) and uniform crossover ($P_{cross} = 0.6$).

4. Results

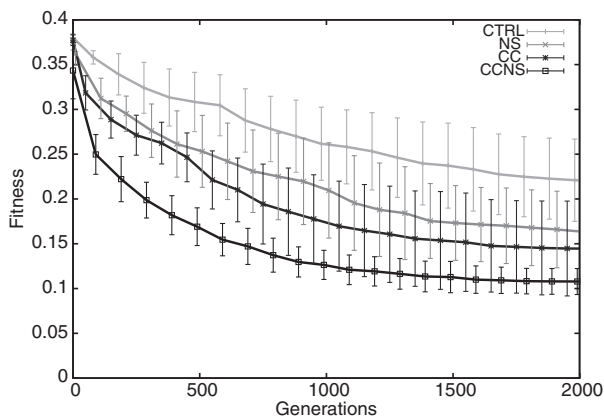
For each of the four controller models, we performed ten independent evolutionary runs, initialized with distinct populations of 200 random individuals. Each evolution was conducted for 2000 generations.

4.1. Evolvability

Figure 6 shows, for each model, the fitness profile of the best controllers averaged across all evolutionary runs. The non-homeostatic model (CTRL) is less evolvable than all the homeostatic models (CC, NS, CCNS) both in terms of final result and convergence speed (p -values $< .01$). Final performances of the CC and NS models are not significantly different (p -value $> .1$), meaning that one homeostatic constraint does not outperform the other. In contrast, the CCNS model combining both constraints appears to be the most evolvable especially regarding results consistency

Table 2. Allele sets of numerical genes

Genes	Alleles				
τ_{m_i}	0.02	0.165	0.31	0.455	0.6
g_i	2.46	3.53	5.34	9.43	31.26
θ_i	-0.2	-0.1	0.0	0.1	0.2
τ_{s_p}	0.2	2.65	5.1	7.55	10.0

**Figure 6.** Average best fitness for each neuronal model.

(indicated by the lower standard deviation) and convergence speed. Although, considering average final performances, the difference between the CC and CCNS model is not significant enough (p -value = .06).

4.2. Behavioral stability and robustness

In order to verify whether the homeostatic constraints promote behavioral stability or not, we evaluated the fitness of every best controller in time-extended simulation lasting ten times the evolutionary one (i.e. $T = 100$ seconds). Figure 7 shows the obtained global fitness and errors specific to each scenario.

Leading to the lowest error values, the scenario A appears to be the least challenging. For all models, prolonged simulations drop the global fitness. This is especially marked for the CC model (whose fitness got worse by 72%) and less important for the CTRL, NS and CCNS models (resp. 43%, 33% and 20% worse). In term of absolute performance, the superiority of the CCNS is confirmed, whereas the CTRL model displays the worst results. Again, differences between the CC and NS models are not significant, although we notice a slight but systematic inversion in their relative fitness whatever the scenario, compared to evolutionary fitness.

To focus on robustness to the friction perturbation, we further evaluated the behavioral stability of the best

controllers using a custom test based on the scenario C. In that test, controllers were given a constant desired speed $V_d = 30$ cm/s. A controller would succeed in the test if the robot reached the distance of 150 meters in less than 1000 seconds. Notice that, for a perfect controller continually respecting the desired velocity, this distance should be covered in 500 seconds. Compared to the previous fitness-based test, this custom test is more demanding on behavioral stability, since it involves longer simulated time.

We performed this test in perturbed condition and control condition. In the latter, we held constant the friction coefficient at $k_{fr} = 10$ kg/s, whereas with perturbation, we increased it twofold such as $k_{fr} = 20$ kg/s at the instant $t = 250$ s.

Table 3 shows the success rate and average time-to-goal of each controller type in both environments. These results confirm and refine the previous ones. Showing almost optimal performances in both environments, CCNS controllers are more stable and robust than the others. The CTRL model gives the worst results, especially poor in the perturbed environment. CC and NS controllers display similar average success and good time-to-goal as long as the perturbation does not occur. However, in the perturbed environment, the success rate of the CC model drops critically, whereas NS controllers performances are only slightly affected, thus showing a great robustness (comparable to the CCNS one).

4.3. Behavioral analysis

In order to understand why homeostatic constraints led to improved performances, we analyzed both the behavior and internal dynamics (in the next section) characterizing the best obtained CCNS controller (whose phenotype is depicted on Figure 4). We first verified the capacity of this controller to follow entirely new desired velocity profiles, with and without perturbations. For instance, Figure 9a shows that the controller is able to respect a sinusoidal command while being perturbed. In Figure 9b, the square signal example clearly demonstrates the three basic capabilities of “speed control”, “stop-and-go” and “perturbation compensation”, which correspond to the three evolutionary scenarios.

To achieve such behaviors, the controller strategy is the following. For propulsion, the three leg’s degrees of freedom oscillate in phase at a constant frequency about 2 Hz. Amplitudes are small compared to the available ranges. Very simply, the locomotion cycle splits in two alternating phases: a stance phase (protraction, elevation and flexion) and a swing phase (retraction, depression and extension).

For braking, the controller uses a passive method. Thus, when the actual velocity is high above the desired

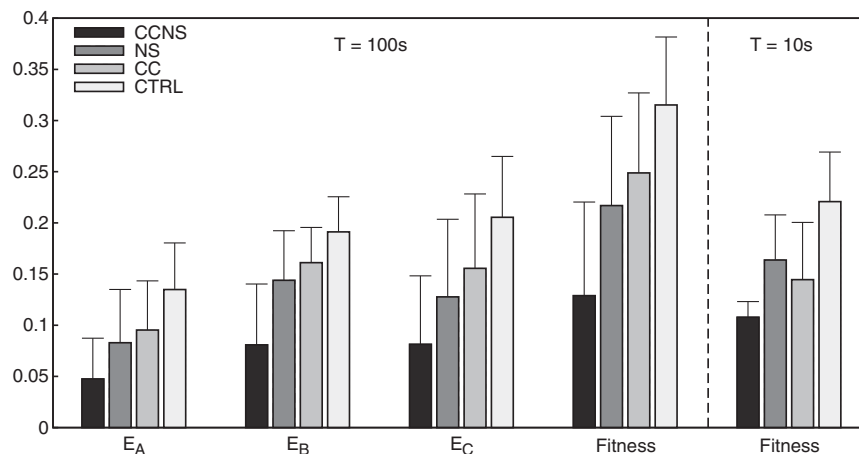


Figure 7. Mean performances (+ std dev.) of the best controllers in long-lasting scenarios ($T = 100$ s). For every model, each best controller of each evolutionary run was evaluated ten times with different random initial synaptic weights.

Table 3. Success rate of the best controllers in reaching 150 m distance at desired speed $V_d = 30$ cm/s, without and with friction perturbation (i.e. $k_{fr} = 10 \rightarrow 20$ kg/s at $t = 250$ s). Average times-to-goal of the successful controllers are also reported. For every model, each best controller of each evolutionary run was evaluated ten times with different random initial synaptic weights

Models	Without perturbation		With perturbation	
	Success [%]	Time-to-goal [s]	Success [%]	Time-to-goal [s]
CCNS	90	537	80	586
NS	72	597	70	669
CC	64	576	37	675
CTRL	55	792	20	849

velocity, the controller triggers a reflex similar to a high amplitude swing phase. This lifts the leg off the ground and let the friction force brakes the robot advance. Quickly after the reflex occurred, the leg stabilizes off the ground in a stationary configuration ($q_1 = -4^\circ$, $q_2 = 21^\circ$, $q_3 = -76^\circ$).

For compensating the friction perturbation, as well as adjusting the robot velocity to the desired value, the best CCNS controller uses a common strategy. The latter consists in shifting and modulating the amplitude of both protraction and extension oscillations. As shown in Figure 8, this results in modulating the stride length. Then, because the gait frequency remains constant, it ends up by either changing the overall robot speed or compensating the varying friction force.

4.4. Functional analysis

Functional analyses were conducted on the same CCNS controller, in order to understand how the plastic and homeostatic mechanisms it integrates work in

concert for giving rise to the efficient behavior previously described.

4.4.1. Neuronal and synaptic dynamics. The first question we addressed was how the controller's internal dynamics correlate to the exhibited behavior. Figure 10 shows timeplots of the neurons activities and synaptic weights obtained for the square velocity command of Figure 9(b). None of the eight neurons is saturating or silent. Synaptic weights spread uniformly over the whole value range. Propulsion and stopping phases are characterized by two clearly different synaptic configurations. During propulsive behavior, both neuronal and synaptic dynamics are oscillatory, at a unique frequency matching the gait cycle. In contrast, during stops, both neuronal and synaptic configurations are mostly stationary. In addition, whether the friction perturbation occurs or not during propulsive phases also influences controller's dynamics, yet to a small extent.

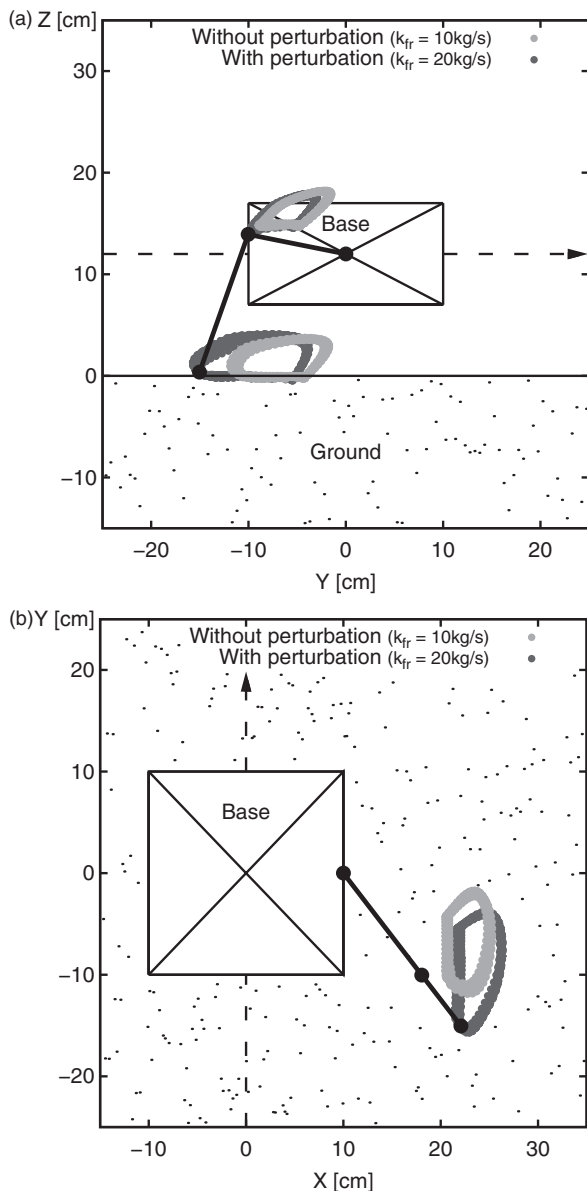


Figure 8. Trajectories of leg's knee and tip, with and without the friction perturbation. (a) Side view (b) Top view.

4.4.2. Synaptic plasticity and homeostatic constraints. In order to uncover the specific roles of synaptic plasticity and each homeostatic constraint, we disrupted each mechanism independently. Here again, we used the square velocity example.

Figure 11(a) shows the behavior arising from a random synaptic configuration not ever modified by any plastic rules (yet respecting both the NS constraint and the sign \pm_{ji} in the genotype). Whatever the random configuration generated, the controller was not able to properly control the robot speed. However, it likely led to oscillatory regimes moving the robot. Moreover, oscillation frequencies obtained were often similar to the normal controller's one.

We then investigated freezing the synaptic plasticity during both propulsive and stopping phases (Figure 11(b) and 11(c)). In the former case, the CCNS controller is still capable of respecting a high velocity, yet compensating a bit worse for the friction perturbation. But, interestingly, the controller is then unable to inhibit entirely the gait when the command becomes null. In the case of freezing the plasticity during a stopping phase, gait inhibiting is still performed well, whereas the controller is not more able to reach a high speed command.

Using the same method, we tested relaxing each homeostatic constraint in different contexts. For both constraint relaxations, consequences did not depend on the moment (i.e. on the controller state) we elicited them. Thus, here we only report results obtained for initial suppression of each constraint.

As shown in Figure 12, relaxing one constraint or the other induces totally distinct effects. On one hand, removing synaptic normalization (NS) still results in a working controller yet it has poor performance. In particular, the controller is unable to stop the robot when asked. In addition, as one could expect, this saturates several synapses. In contrast, as for neuronal activity, no saturation or silencing is induced. On the other hand, removing the center-crossing constraint (CC), by switching from the mixed equation (9) and equation (11) to equation (11), results in a controller that does not work at all. Logically, most neurons become either saturated or silent, paralysing the whole robot.

4.4.3. Plasticity dynamics. We further investigated how the plasticity dynamics impacts the emerging behavior, by achieving two more experiments on the same best CCNS controller.

First, we increased tenfold every plastic time constant $\tau_{s_{ji}}$ to slow down the plasticity dynamics. Figure 13(a) shows that this modification does not significantly affect the asymptotic behaviors. In particular, notice that the frequency of oscillatory regimes remains unchanged. However, behavioral transitions take much more time to finish and are easier to observe. Interestingly, this confirms that compensating the friction perturbation relates more to small parameter adjustments, than to an actual change in the dynamics.

Second, we tested whether individual differences in the dynamics of adaptive synapses were relevant to proper operation of the controller. Thus, we set all the plastic time constants to their mean value $\bar{\tau}_s = 1.56$ seconds. Results (Figure 13(b)) show that the controller is still able to generate a performing walking gait. However, some instability arises in the presence of the friction perturbation. Furthermore, the controller is unable to inhibit the locomotion cycle when subjected to a null command.

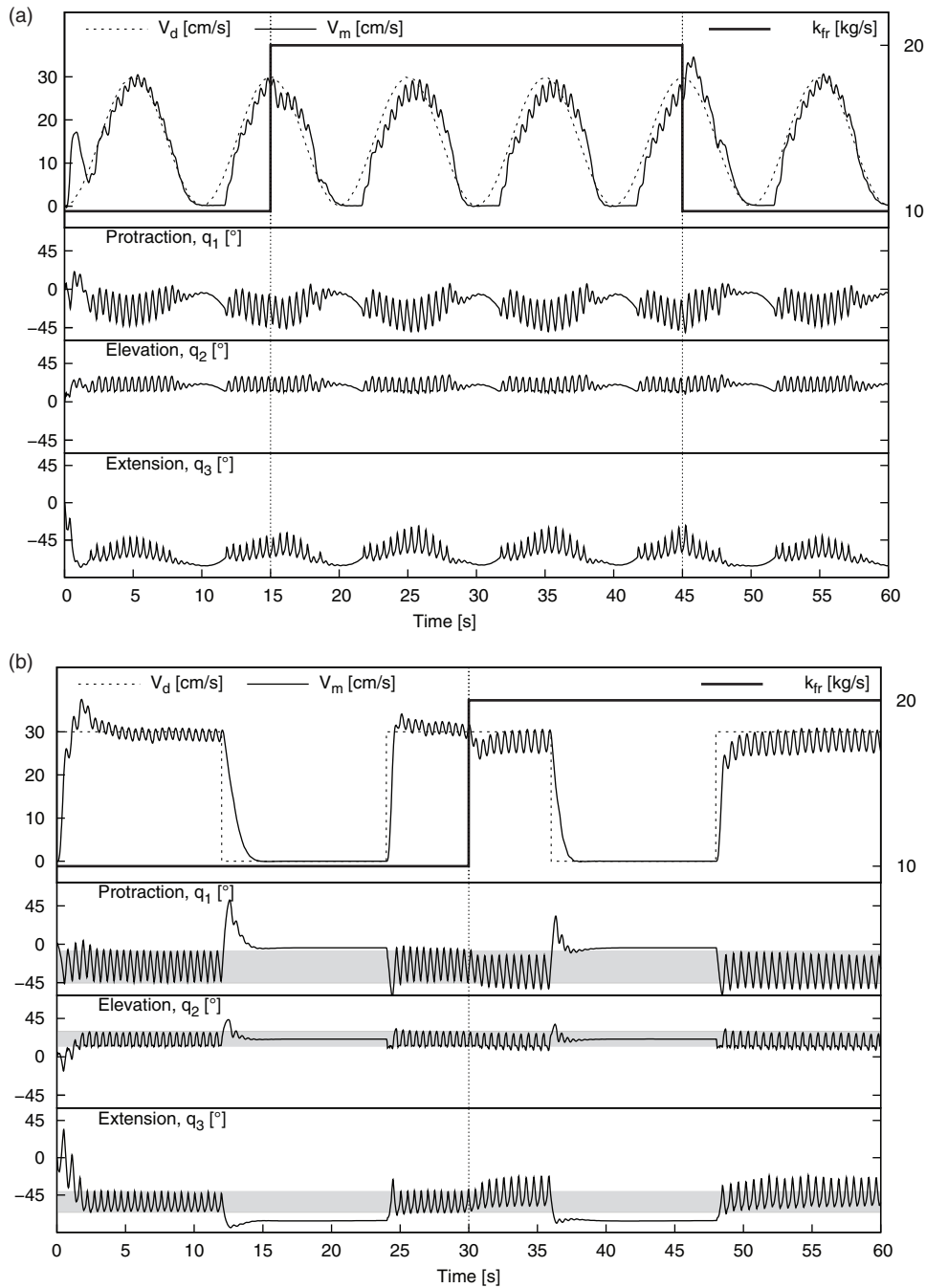


Figure 9. Velocity control and joint angles of the best CCNS controller with and without friction perturbation: (a) Sinusoidal velocity command of period 10 seconds; (b) Square velocity command of period 24 seconds. Instantaneous mean velocity V_m is oscillating (yet damped by the low-pass filter) according to every robot stride. An initial transitory phase lasting three seconds is noticeable. Gray stripes denote joint angle amplitudes characterizing the locomotion at $V_d = 30$ cm/s without friction perturbation (i.e. $k_{fr} = 10$ kg/s). Applying and removing the friction perturbation (highlighted by the vertical dash lines) induces undershoots and overshoots of the robot velocity respectively, which are quickly corrected by the controller.

4.4.4. Robustness. We ended our functional analysis of the best CCNS controller by studying its robustness to internal perturbation. While interacting with the environment, we disrupted the whole synaptic configuration by setting every synaptic weight to a new

random value. As shown in Figure 14, the controller proved to be very robust whatever the context in which we did the randomization. In every instance, the performance, activity and synaptic steady states are quickly recovered (in two to three seconds).

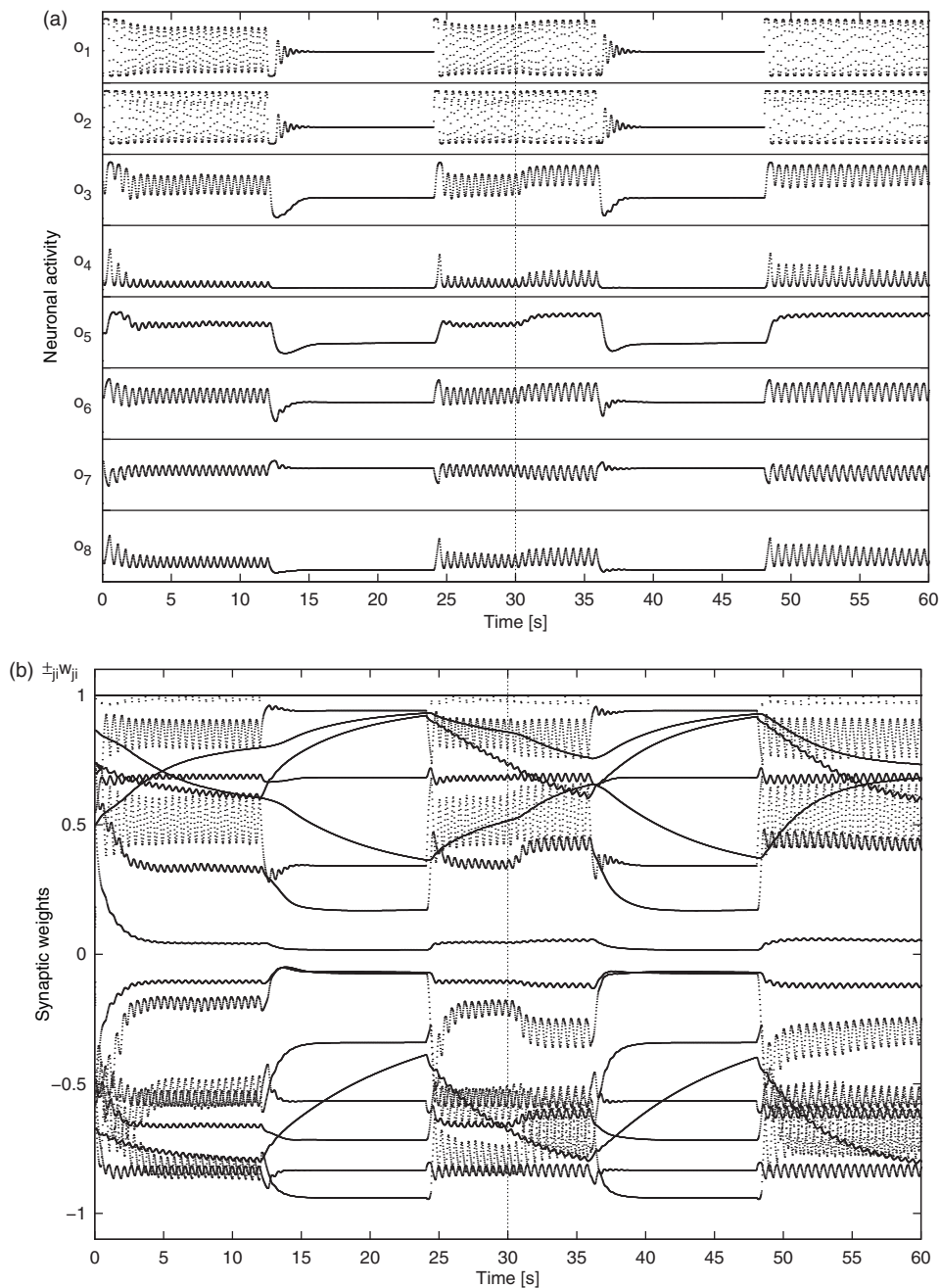


Figure 10. Dynamics of neuronal activity (a) and synaptic weights (b) of the best CCNS controller in the case of the square velocity example (Figure 9(b)). As two neurons (no. 2 and 5) each receives only one synaptic input, the normalization constraint keeps their weights fixed to 1.

5. Discussion

5.1. Hebbian plasticity underlies behavioral adaptivity

From the plasticity blocking tests performed on the best CCNS controller (Figure 11), we can first infer that adaptive synapses are not taking part in oscillatory

pattern generation. Actually, the contrary would have been possible given the relatively fast dynamics of the learning rules (yet slower than the neuronal activation dynamics). Pattern generation arises instead from the particular network topology and tuning of neuronal time constants, as confirmed by the fact that random fixed synaptic configurations still lead to effective propulsive behavior (Figure 11(a)). Furthermore, changes in plasticity speed (Figure 13) do not stop

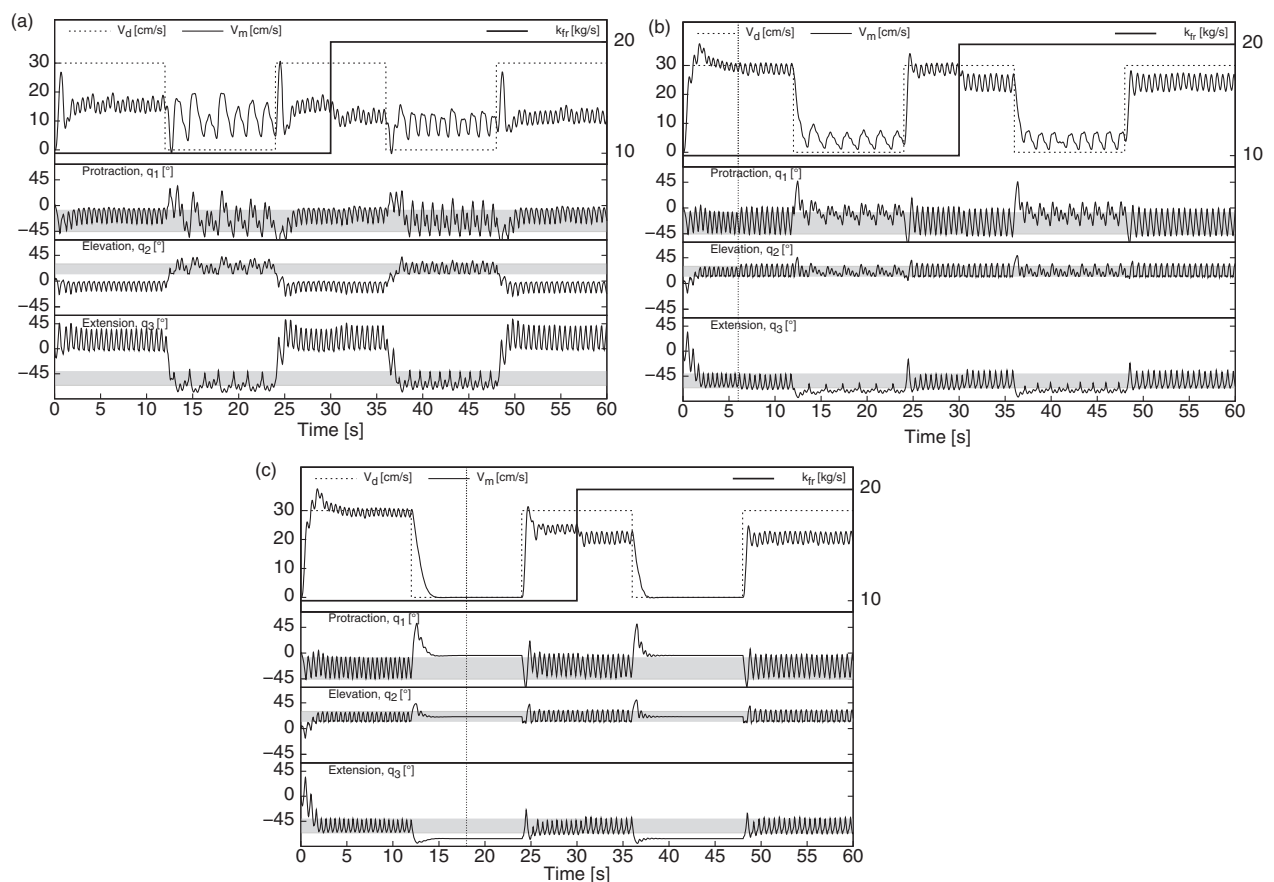


Figure 11. Effects of synaptic plasticity suppression: (a) Keeping the random initial synaptic configuration (yet normalized) makes the controller unable to control the robot speed, but a locomotion behavior is still exhibited. (b) Synaptic plasticity frozen during propulsion phase (from $t=6$ s): The controller is then unable to inhibit the robot locomotion. (c) Synaptic plasticity frozen during stop phase (from $t=18$ s): The controller is then unable to fully respect a high speed command. (Gray stripes corresponding to the intact controller behavior are reported here and in subsequent figures for comparison).

gait generation, neither do they modify oscillation frequency or phase.

The plasticity blocking tests however clearly show the role of synaptic plasticity in on-line behavior adaptation. The controller displays two distinct dynamics according to the required velocity: one for high speed propulsion and one for movement inhibition. Freezing all the synaptic weights in one dynamics prevents switching afterwards to the other (Figure 11). Besides, this relation between behavior transitions and internal plasticity becomes obvious when slowing down all adaptive synapses (Figure 13(a)).

On the other hand, adaptivity is less apparent when the controller compensates for the friction perturbation. In fact, small variations in synaptic weights are noticeable (Figure 10(b)) and induce only some changes in leg movement amplitude (Figure 8), whereas frequency and phase remain constant. Moreover, blocking the plasticity while being in the

propulsive regime without perturbation does not impair the later controller performance significantly, when the friction coefficient is to be doubled (Figure 11(b)). Therefore, plasticity does not seem involved in compensating such a perturbation. One possible explanation might be that the friction perturbation is not that demanding. Another explanation, which may add to the previous one, lies in that, in our setup, controllers could hardly detect this perturbation. Indeed, both friction perturbation and some increase in required velocity lead to similar rise in the control error perceived by controllers. In this way, a null velocity command makes a greater perturbation for controllers to deal with.

To summarize, the plasticity due to adaptive synapses is involved in adapting the control to big changes in required velocity, rather than in compensating the probably trivial friction perturbation. Consistent with previous work (Urzelai & Floreano, 2001), this result

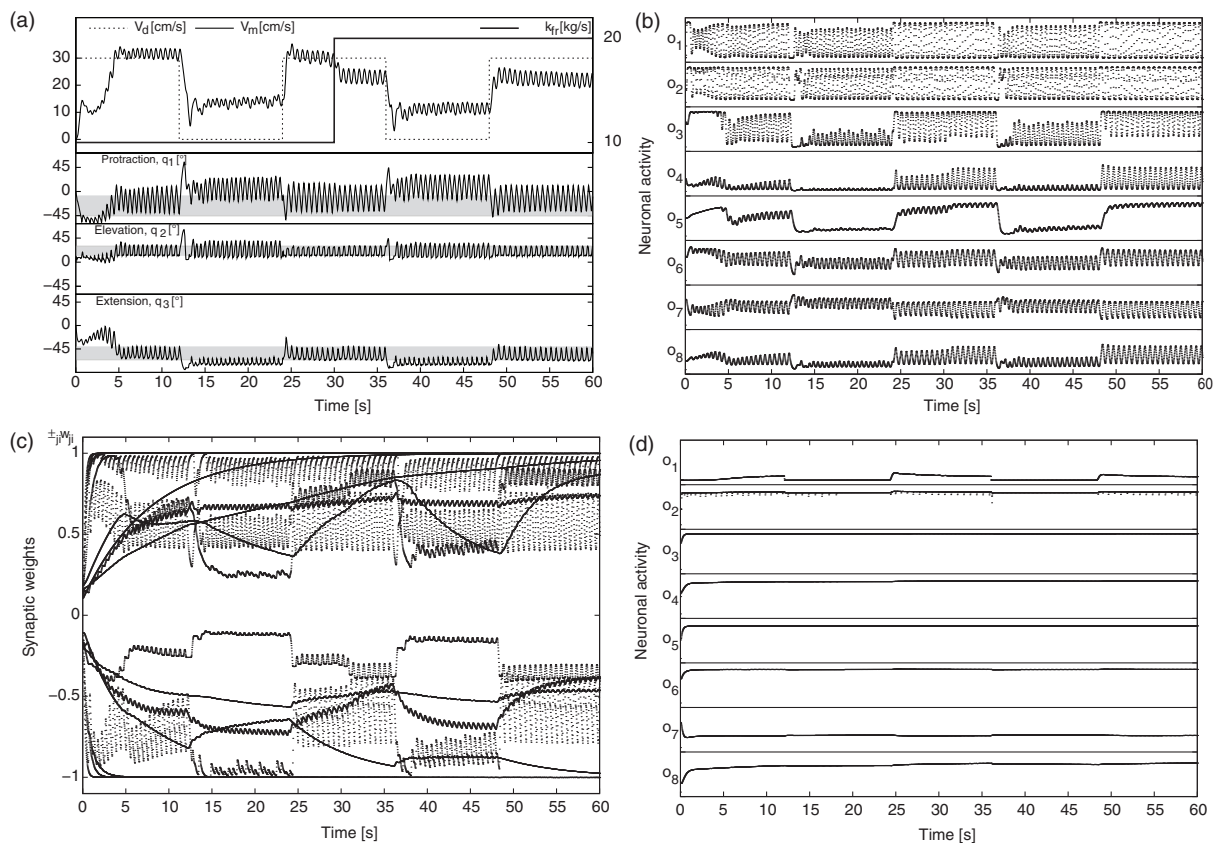


Figure 12. Effects of homeostatic constraint suppression: (a) Synaptic normalization constraint relaxation: The initial transitory regime is lengthened. The controller is unable to stop the robot. A significant part of synaptic weights is saturating (c), whereas neuronal activity remains oscillatory without saturating (b). (d) Center-crossing constraint relaxation: Almost all neurons display flat saturated activity, resulting in a non functional controller.

supports that the adaptive synapses model allows evolving controllers implementing conflicting sub-behaviors.

5.2. Synaptic normalization supports multistability

As pointed out in the introduction, the adaptivity of controllers evolved with adaptive synapses (namely the PNN model) has been offset by some behavioral instability before (McHale & Husbands, 2004a, 2004b; Urzelai & Floreano, 2001). This drawback is confirmed by our stability tests (Table 3) performed on the CTRL controllers (i.e. unconstrained CTRNN/PNN hybrid model), which display poor results especially in perturbed environment. On the other hand, evolving controllers with normalized synapses (i.e. NS and CCNS controllers) led to far more reliable behaviors with and without perturbation (Table 3).

Our functional analyses of the best CCNS controller give more insight into how the normalization constraint interacts with plasticity. Indeed, relaxing this constraint results in the controller inability to stop the robot when asked (Figure 12(a)). That is, synaptic

normalization seems to help stable coexistence of the two conflicting regimes underlying propulsion and movement inhibition. To explain such a multistability, we hypothesize a canalizing effect of the normalization constraint on synaptic plasticity. In particular, the induced synaptic competition might contribute to a global self-organization of individual plastic rules.

On the other hand, synaptic normalization does not seem to implicitly constrain neuronal activity as expected. If it does effectively prevent synaptic weights from saturating (Figure 12(c)), its effect on neuronal activity is not significative. In fact, when synaptic normalization is removed (Figure 12(b)), neuronal activity is left oscillating in the proper range for effective propulsion and there is not any saturated or silenced neuron.

5.3. Center-crossing constraint prevents neuronal saturation, but not robustness

In contrast to synaptic normalization, the center-crossing constraint obviously contributes to the sensitivity of the best CCNS controller. Indeed, suppressing this

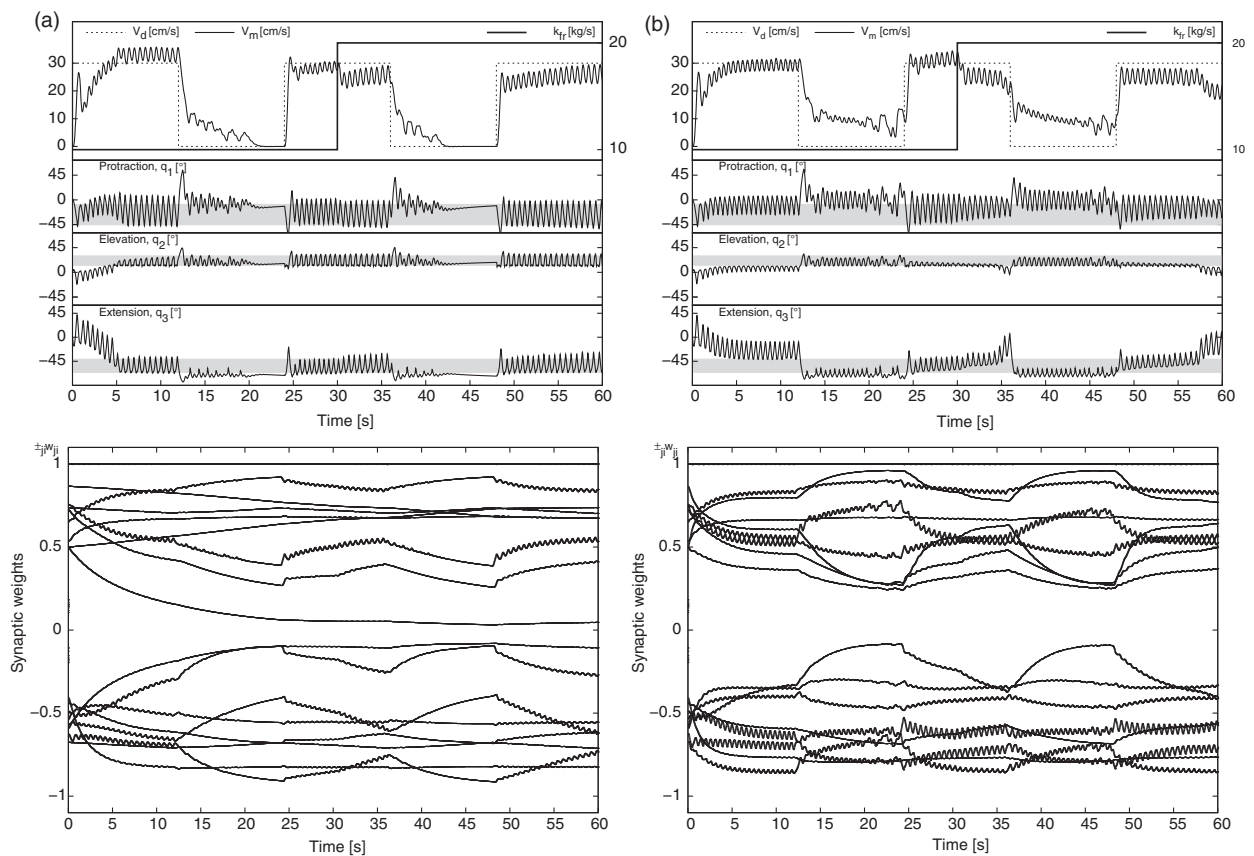


Figure 13. Effects of disturbing plasticity dynamics: (a) Tenfold multiplication of the plasticity time constants τ_{s_j} : Behavior transitions are made slower, whereas asymptotic behaviors remain unchanged. (b) All the plasticity time constants set to their mean value $\bar{\tau}_s = 1.56$ s: The controller is unable to stop the robot when needed. Propulsion behavior is also altered and seems unstable as noticed after $t = 35$ s and $t = 58$ s.

constraint leads to neuronal activity saturation, and the controller becomes fully paralyzed (Figure 12(d)).

However, the center-crossing constraint does not appear to promote the evolution of robust behaviors. Thus, as shown in our comparative analyses, performance of CC controllers drops critically in long lasting simulations when friction perturbation is triggered (Table 3). So, the enhanced evolvability of CC controllers likely comes from focusing evolutionary search on more sensitive and/or oscillating neural networks (Mathayomchan & Beer, 2002).

5.4. Robustness through homeostatic hebbian plasticity

As shown by the quick recovery from synaptic randomization perturbations (Figure 14), dynamics of the best CCNS controller is governed by very stable dynamic attractors, which are shaped by each specific plastic rule and rate tuned at every synapse (Figure 13(b)). Furthermore, in each condition of desired speed and

friction coefficient, the dynamics perturbed by random synaptic values return to the *same* previous steady state.

Therefore, the best CCNS controller clearly demonstrates the property of homeostasis as defined in the background section (Kitano, 2007). Moreover, this homeostasis underlies behavioral robustness, as it results in maintaining the evolved locomotor performance. Interestingly, homeostasis is here supported by hebbian plasticity, which is often considered destabilizing as being of positive-feedback nature. This may be due to the multistability, canalizing and sensitivity properties provided by the static constraints we proposed.

Another interesting point is that, compared to previous works, homeostasis evolved implicitly without relying on any active regulation mechanism explicitly homeostatic. Hence, it blends the distinction between homeostatic and non homeostatic mechanisms, leaving this property for rather characterizing controller dynamics.

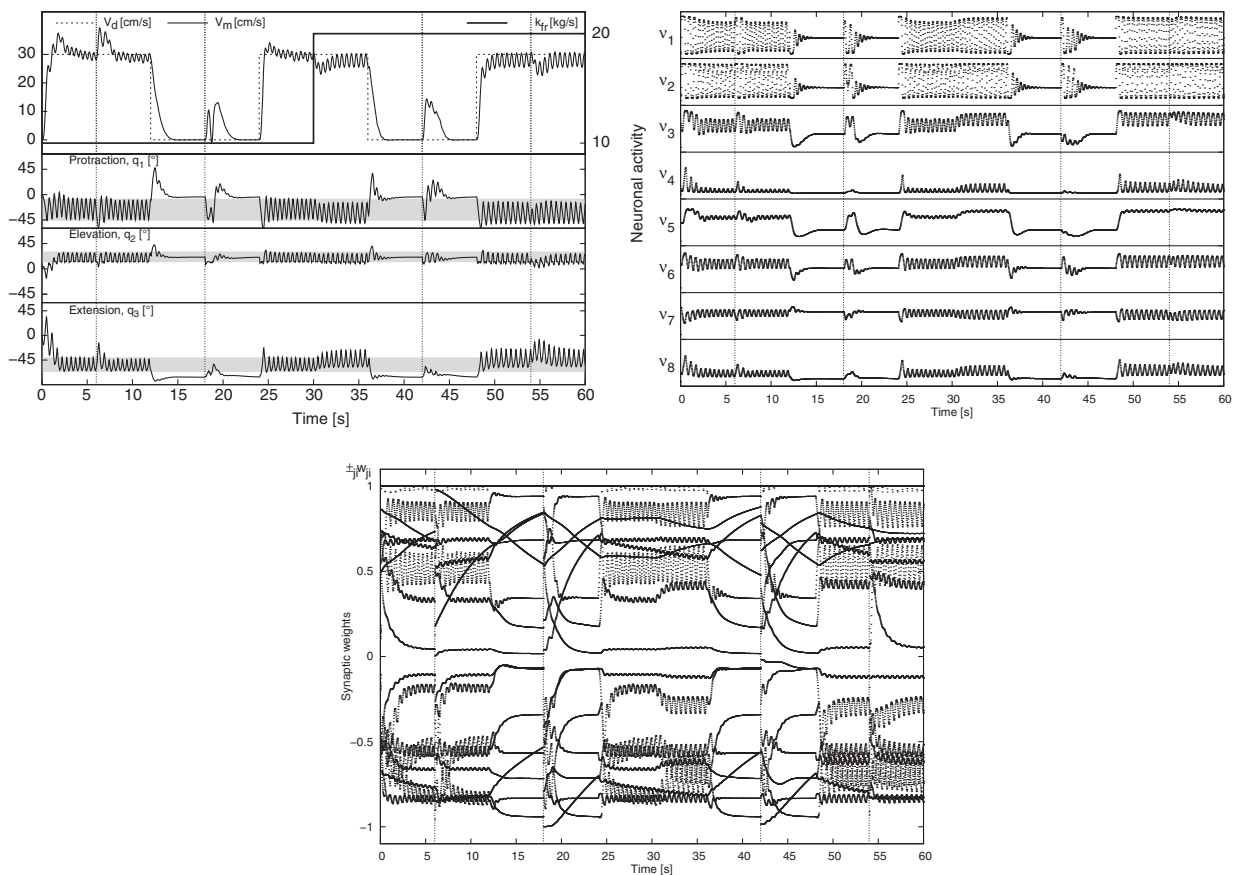


Figure 14. Robustness and homeostasis of the best CCNS controller: All synaptic weights were set to new random values, at $t = 6$ s, 18 s, 42 s and 54 s. Whatever the behavior phase in which we triggered this perturbation, the controller quickly got back to the same steady state (both activity and synapses), thus recovering the performance level.

5.5. Task-independence

So far, our findings are valid for the single-legged locomotion task only. However, we would assume that our models could be used for other problems. By being based on CTRNNs and adaptive synapses, our model should be practicable for evolving behaviors that have already been tackled in former investigations. Moreover, the proposed constraints are not related to locomotion, nor to any other task. Admittedly, more results are required to better define the area of applicability of the model.

6. Conclusion

In this paper, we have shown that plastic neurocontrollers based on adaptive synapses are more evolvable when subjected to static parameter constraints inspired from homeostatic processes of biological neurons. In particular, we could evolve both flexible and multi-stable pattern generators, leading to both adaptive and robust single-legged robot locomotion. Moreover, we

have seen that homeostasis can evolve implicitly without any active homeostatic mechanisms and be implemented through constrained hebbian plasticity.

As stable flexibility is required in many interesting behaviors, more investigations in constraining neural network models seem promising. This is especially true for plastic models which hold more potential for adaptivity but also for instability.

Further progress could be achieved through mathematical analyses of model parameter spaces (Mathayomchan & Beer, 2002; Beer, 2006; Beer & Daniels, 2010; Prinz, Bucher, & Marder, 2004; Achard & De Schutter, 2006). Unfortunately, for some complex models this might not be practical. Another approach, that we followed here, is to take inspiration from how biological neural networks tune their parameters. However, although we know developmental process and homeostatic plasticity are involved, a global understanding is not yet available (Marder, Tobin, & Grashow, 2007; Goaillard, Taylor, Schulz, & Marder, 2009). Then, how to abstract these experience-dependent processes in static parameter constraints? For a given task, which constraints would promote to evolve

homeostasis, multistability, robustness and adaptivity? A comparison between static constraints and active homeostatic mechanisms would be very helpful.

Notes

1. For a CTRNN defined with an odd activation function $\sigma(y_i^f + \theta_i)$, the center-crossing condition is verified if there is an equilibrium point for which the activations of all neurons are zero (Beer, 1995; Mathayomchan & Beer, 2002). Since $\sigma(0) = 0$, this condition occurs when $y_i^f = -\theta_i$. After substitution into equation 1 and solving for θ_i in steady state (with $I_i^f = 0$), we obtain that the center-crossing condition is given by $\theta_i^{**} = 0$.
2. Rigid body dynamics simulation has been carried out using the open source, high performance library *Open Dynamics Engine* (opende.sourceforge.net).
3. Each angle was set by applying a torque computed through a PD control loop.

Funding

This research received no specific grant from any funding agency in the public, commercial, or not-for-profit sectors.

References

- Achard, P., & De Schutter, E. (2006). Complex parameter landscape for a complex neuron model. *PLoS Computational Biology*, 2(7), e94.
- Baker, J. E. (1987). Reducing bias and inefficiency in the selection algorithm. In J. J. Grefenstette (Ed.), *Proceedings of the International Conference on Genetic Algorithms and their Application* (pp. 14–21). Hillsdale, NJ: Lawrence Erlbaum Associates.
- Beer, R. D. (1995). On the dynamics of small continuous-time recurrent neural networks. *Adaptive Behavior*, 3(4), 469–509.
- Beer, R. D. (2006). Parameter space structure of continuous-time recurrent neural networks. *Neural Computation*, 18, 3009–3051.
- Beer, R. D., & Daniels, B. C. (2010). *Saturation probabilities of continuous-time sigmoidal networks*. (arXiv preprint 1010.1714).
- Bienenstock, E. L., Cooper, L. N., & Munro, P. W. (1982). Theory for the development of neuron selectivity: orientation specificity and binocular interaction in visual cortex. *Journal of Neuroscience*, 2(1), 32–48.
- Bullock, S. (2006). The fallacy of general purpose bio-inspired computing. In L. M. Rocha, L. S. Yaeger, M. A. Bedau, D. Floreano, R. L. Goldstone & A. Vespignani (Eds.), *Artificial Life X : Proceedings of the International Conference on the Simulation and Synthesis of Living Systems* (pp. 540–545). Cambridge, MA: MIT Press.
- Di Paolo, E. A. (2000). Homeostatic adaptation to inversion of the visual field and other sensorimotor disruptions. In J.-A. Meyer, A. Berthoz, D. Floreano, H. L. Roitblat & S. W. Wilson (Eds.), *From Animals to Animats 6: Proceedings of the International Conference on the Simulation of Adaptive Behavior, SAB2000* (pp. 440–449). Cambridge, MA: MIT Press.
- Di Paolo, E. A. (2002a). *Evolving robust robots using homeostatic oscillators (Cognitive Science Research Paper No. 548)*. Brighton, UK: School of Cognitive and Computing Sciences, University of Sussex.
- Di Paolo, E. A. (2002b). Spike-timing dependent plasticity for evolved robots. *Adaptive Behavior*, 10(3–4), 243–263.
- Di Paolo, E. A. (2003). Evolving spike-timing-dependent plasticity for single-trial learning in robots. *Philosophical Transactions of the Royal Society of London A*, 361(1811), 2299–2319.
- Doncieux, S. (2003). *Évolution de contrôleurs neuronaux pour animats volants : méthodologie et applications. Doctoral dissertation in French, AnimatLab/LIP6* (pp. 149–150). Paris: Université Pierre et Marie Curie.
- Floreano, D., & Mondada, F. (1996). Evolution of plastic neurocontrollers for situated agents. In P. Maes, M. J. Mataric, J.-A. Meyer, J. Pollack & S. W. Wilson (Eds.), *From Animals to Animats 4: Proceedings of the International Conference on Simulation of Adaptive Behavior, SAB1996* (pp. 402–410). Cambridge, MA: MIT Press.
- Floreano, D., & Urzelai, J. (2000). Evolutionary robots: The next generation. In T. Gomi (Ed.), *Evolutionary Robotics III* (pp. 231–266). Ontario, Canada: AAI Books.
- Goaillard, J. M., Taylor, A. L., Schulz, D. J., & Marder, E. (2009). Functional consequences of animal-to-animal variation in circuit parameters. *Nature Neuroscience*, 12, 1424–1430.
- Goldberg, D. E. (1989). *Genetic algorithms in search, optimization, and machine learning*. Reading, MA: Addison-Wesley.
- Harvey, I., Husbands, P., Cliff, D., Thompson, A., & Jakobi, N. (1996). Evolutionary robotics at Sussex. *Proceedings of the International Symposium on Robotics and Manufacturing, ISRAM1996* (Vol 6, pp. 293–298). New York: ASME Press.
- Hoinville, T., & Hénaff, P. (2004a). Comparative study of two homeostatic mechanisms in evolved neural controllers for legged locomotion. *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS2004* (pp. 2624–2629). Piscataway, NJ: IEEE.
- Hoinville, T., & Hénaff, P. (2004b). Evolving plastic neural controllers stabilized by homeostatic mechanisms for adaptation to a perturbation. In J. Pollack, M. A. Bedau, P. Husbands, T. Ikegami & R. A. Watson (Eds.), *Artificial Life IX: Proceedings of the International Conference on the Simulation and Synthesis of Living Systems* (pp. 81–87). Cambridge, MA: MIT Press.
- Husbands, P., Smith, T. M. C., O'Shea, M., Jakobi, N., Anderson, J., & Philippides, A. O. (1998). Brains, gases and robots. In L. Niklasson, M. Boden & T. Ziemke (Eds.), *Proceedings of the International Conference on Artificial Neural Networks, ICANN98* (pp. 51–64). Berlin: Springer-Verlag.
- Iizuka, H., & Di Paolo, E. A. (2007). Toward spinozist robotics: Exploring the minimal dynamics of behavioural preference. *Adaptive Behavior*, 15, 359–376.
- Iizuka, H., & Di Paolo, E. A. (2008). Extended homeostatic adaptation: Improving the link between internal and behavioural stability. In M. Asada, J. C. T. Hallam, J.-

- A. Meyer & J. Tani (Eds.), *From Animals to Animats 10: Proceedings of the International Conference on the Simulation of Adaptive Behavior, SAB2008* (pp. 1–11). Berlin: Springer-Verlag.
- Jakobi, N., & Quinn, M. (1998). Some problems (and a few solutions) for open-ended evolutionary robotics. In P. Husbands & J.-A. Meyer (Eds.), *Proceedings of the European Workshop on Evolutionary Robotics, EvoRobot1998* (pp. 108–122). Berlin: Springer-Verlag.
- Kepecs, A., van Rossum, M. C., Song, S., & Tegnér, J. (2002). Spike-timing-dependent plasticity: common themes and divergent vistas. *Biological Cybernetics*, 87(5–6), 446–458.
- Kitano, H. (2007). Towards a theory of biological robustness. *Molecular Systems Biology*, 3(137), 1–7.
- Marder, E., Tobin, A. E., & Grashow, R. (2007). How tightly tuned are network parameters? Insight from computational and experimental studies in small rhythmic motor networks. *Progress in Brain Research*, 165, 193–200.
- Mathayomchan, B., & Beer, R. (2002). Center-crossing recurrent neural networks for the evolution of rhythmic behavior. *Neural Computation*, 14, 2043–2051.
- McHale, G., & Husbands, P. (2004a). GasNets and other evolvable neural networks applied to bipedal locomotion. In S. Schaal, A. J. Ijspeert, A. Billard, S. Vijayakumar, J. Hallam & J.-A. Meyer (Eds.), *From Animals to Animats 8: Proceedings of the International Conference on the Simulation of Adaptive Behavior, SAB2004* (pp. 163–172). Cambridge, MA: MIT Press.
- McHale, G., & Husbands, P. (2004b). Quadrupedal locomotion: GasNets, CTRNNs and hybrid CTRNN/PNNs compared. In J. Pollack, M. A. Bedau, P. Husbands, T. Ikegami & R. A. Watson (Eds.), *Artificial Life IX: Proceedings of the International Conference on the Simulation and Synthesis of Living Systems* (pp. 106–112). Cambridge, MA: MIT Press.
- Miller, K. D., & MacKay, D. J. C. (1994). The role of constraints in Hebbian learning. *Neural Computation*, 6, 100–126.
- Prinz, A. A., Bucher, D., & Marder, E. (2004). Similar network activity from disparate circuit parameters. *Nature Neuroscience*, 7, 1345–1352.
- Stanley, K. O., Bryant, B. D., & Miikkulainen, R. (2003). Evolving adaptive neural networks with and without adaptive synapses. *Proceedings of the Congress on Evolutionary Computation, CEC2003* (Vol. 4, pp. 2557–2564). Piscataway, NJ: IEEE.
- Tuci, E., & Quinn, M. (2003). Behavioural plasticity in autonomous agents: A comparison between two types of controller. In S. Cagnoni et al (Eds.), *Proceedings of the International Conference on Applications of Evolutionary Computing, EvoWorkshops2003* (Vol. 2661, pp. 661–672). Berlin: Springer-Verlag.
- Turrigiano, G. G., & Nelson, S. B. (2000). Hebb and homeostasis in neuronal plasticity. *Current Opinion in Neurobiology*, 10(3), 358–364.
- Urzelai, J., & Floreano, D. (2001). Evolution of adaptive synapses: Robots with fast adaptive behavior in new environments. *Evolutionary Computation*, 9(4), 495–524.
- Williams, H. (2004). Homeostatic plasticity in recurrent neural networks. In S. Schaal, A. J. Ijspeert, A. Billard, S. Vijayakumar, J. Hallam & J.-A. Meyer (Eds.), *From Animals to Animats 8: Proceedings of the International Conference on Simulation of Adaptive Behavior* (pp. 344–353). Cambridge, MA: MIT Press.
- Williams, H. (2005). Homeostatic plasticity improves continuous-time recurrent neural networks as a behavioural substrate. *Proceedings of the International Symposium on Adaptive Motion in Animals and Machines, AMAM2005*. Ilmenau, Germany: Technische Universität.
- Williams, H. (2006). Homeostatic adaptive networks. *Doctoral dissertation*. UK: University of Leeds.
- Williams, H., & Noble, J. (2007). Homeostatic plasticity improves signal propagation in continuous-time recurrent neural networks. *Biosystems*, 87(2–3), 252–259.
- Wood, R., & Di Paolo, E. A. (2007). New models for old questions: Evolutionary robotics and the “A not B” error. In F. Almeida e Costa (Ed.), *Proceeding of the European Conference on Advances in Artificial Life, ECAL2007* (pp. 1141–1150). Berlin: Springer-Verlag.

About the Authors



Thierry Hoinville received an M.Sc. in artificial intelligence from Pierre et Marie Curie University (Paris 6) in 2002 and a Ph.D. in robotics from Versailles University (UVSQ) in 2007. Since 2008, he has been a postdoctoral fellow at the Italian Institute of Technology (IIT). He has been involved in projects covering aspects of bio-inspired control for adaptive legged locomotion, reconfigurable morphology for multimodal legged locomotion, and perception in virtual reality. His research interests include evolutionary robotics, adaptive behavior, neuronal modeling, dynamic simulation, and virtual reality. He is especially interested in engineering biological principles for giving life-like motion to artificial creatures. E-mail: thierry.hoinville@iit.it



Cecilia Tapia Siles received her electromechanical engineer degree in 2003 from Universidad Privada Boliviana. In 2005, she received a scholarship from the French government to study robotics, obtaining her M.Sc. degree in robotics from Ecole Centrale de Nantes on 2006. She is currently a Ph.D. student in co-direction by Pierre et Marie Curie University (Paris 6) and Istituto Italiano di Tecnologia (IIT) Genoa University. Her research is focused on adaptive locomotion for bio-inspired underwater robots. Her research interests include biologically inspired robotics and locomotion generation and control. E-mail: cecilia.tapia@iit.it.



Patrick Hénaff received an M.Sc. degree in robotics from the University of Pierre et Marie Curie (Paris 6) in 1989, and a Ph.D. in robotics in 1994 at the Paris Robotic Laboratory, University of Pierre et Marie Curie. Since 1997, he has been Associate Professor at the Institute of Technology of Cergy Pontoise. He worked as a researcher at the LISV (Systems Engineering Laboratory of the University of Versailles) until 2009. He is now researcher at ETIS lab, University of Cergy Pontoise, CNRS UMR 8051. His topics of interest concern biologically inspired control, particularly the neural network biological models dedicated to locomotion to control dynamical behaviors and to compensate certain deficiencies of legged robots and humanoids robots. E-mail: patrick.henaff@u-cergy.fr.

9.7 On the role of sensory feedbacks in Rowat-Selverston CPG to improve robot legged locomotion, *frontiers in Neurobotics*, 2010



On the role of sensory feedbacks in Rowat–Selverston CPG to improve robot legged locomotion

Elmira Amrollah¹ and Patrick Henaff^{2*}

¹ LISV, University of Versailles St Quentin, Vélizy, France

² ETIS, UMR 8051CNRS-ENSEA-UCP, University of Cergy Pontoise, Cergy-Pontoise, France

Edited by:

Jun Tani, RIKEN Brain Science Institute, Japan

Reviewed by:

Quan Zou, University of Nevada Reno, USA

Hugues Berry, INRIA, France

*Correspondence:

Patrick Henaff, ETIS, UMR CNRS 8051, Université de Cergy-Pontoise, Site de ST Martin, 2 rue A. Chauvin, 95302 Cergy-Pontoise Cedex, France
e-mail: patrick.henaff@u-cergy.fr

This paper presents the use of Rowat and Selverston-type of central pattern generator (CPG) to control locomotion. It focuses on the role of afferent exteroceptive and proprioceptive signals in the dynamic phase synchronization in CPG legged robots. The sensori-motor neural network architecture is evaluated to control a two-joint planar robot leg that slips on a rail. Then, the closed loop between the CPG and the mechanical system allows to study the modulation of rhythmic patterns and the effect of the sensing loop via sensory neurons during the locomotion task. Firstly simulations show that the proposed architecture easily allows to modulate rhythmic patterns of the leg, and therefore the velocity of the robot. Secondly, simulations show that sensori-feedbacks from foot/ground contact of the leg make the hip velocity smoother and larger. The results show that the Rowat–Selverston-type CPG with sensory feedbacks is an effective choice for building adaptive neural CPGs for legged robots.

Keywords: central pattern generator, legged locomotion, sensorimotor coordination, Rowat–Selverston

INTRODUCTION

What has become apparent in recent years is that the questions that biologists have about how walking is generated and controlled in animals are the same questions that any engineer or researcher must answer in order to build a functioning legged robot especially humanoid biped robots. In his review paper Ijspeert (2008) explains that advancing in the fundamental understanding of the functioning of the locomotor nervous system is indispensable in the fields of robotics, biology, and medicine. As in a robot, a biological mechanical apparatus, consisting of a series of articulated legs, must produce coordinated movements allowing animal walking, even on an irregular terrain or even if damages occur to part of its body. A central command is generated and sensors allow online corrections necessary to adapt the movements to the internal or external constraints (Nadim and Manor, 2000; Marder and Bucher, 2001).

It is well-known that biological neural architectures dedicated to locomotion in animals are very complex (see Cattaert and Le Ray, 2001 for a review of the neural organization complexity of one crayfish leg). Nevertheless, for many years, great progress has been made in the understanding of how the central nervous system works. This is the case for swimming with the lamprey spinal cord preparation (Grillner et al., 1995; Di Prisco et al., 2000; Grillner and Wallen, 2002). However, the questions are much more complex when addressed to walking systems for which the central neuronal networks must generate complex patterned activities and control multi-joint appendages. In the nervous systems of invertebrates (mainly insects), the organization of neural networks and sensory signals mediating coordination of multi-segmental organs for locomotion has been studied (for a review, see Büschges, 2005). Many neurons have been identified and their properties and synaptic contacts analyzed. Several recent trials have been made to simulate realistic simulation models of insect walking (Cruse et al., 1995, 1998). For example, in the crayfish walking system, biologists have

shown that the organization of the walking system is largely modular (Gill and Skorupski, 1996; Cattaert and Le Ray, 2001). Each leg is controlled by its own central neuronal network, and there are a large number of neurons for one joint articulation. Biologists have identified different types of neurons that are fundamental to produce a coherent rhythmic pattern: rhythmic neurons, premotor interneurons, and motoneurons. Motoneurons induce the muscle control signal to the muscle fibers. Rhythmic neurons can produce oscillations that are modulated by premotor interneurons. Interneurons are involved in local joint control. Some interneurons which have the role of sensory neurons measure angular position, velocity, or forces of the joint. Other interneurons coordinate interneurons. They activate monosynaptic excitatory synapses onto motoneurons of different joints and establish disynaptic inhibitory connections with antagonistic motoneurons. Most of these interneurons are non-spiking neurons.

As for neuronal connections, biologists have proved that connections among the neuronal network controlling an articular joint of a crayfish leg are of three kinds: a classical chemical synapse activated by the occurrence of a presynaptic activity (some are excitatory, others are inhibitory depending on neurotransmitter and postsynaptic receptors); a tonic chemical synapse that does not require the spike to liberate the neurotransmitter; and electrical connections described between motoneurons of the same pool between sensory terminals of the same sensory organ (Marchand et al., 1997) and between premotor interneurons and motoneurons.

Despite this complexity, it is possible to reduce the architecture of an articular joint of a biological neuronal network to a model that respects its fundamental properties. Mc Crea and Ryback (2008) suggest an interesting neural organization in which a central pattern generator (CPG) is built with a rhythmic generator (RG) level based on oscillatory neurons, and a pattern formation (PF) circuit level based on interneurons responsible for motoneuron activation

(motoneurons are connected to extensor and flexor muscles). In their architecture, the locomotor rhythm and the durations of flexor and extensor phases are determined by the RG network which controls the activity of the PF network by a combination of direct excitation and inhibition mediated by the afferent sensory inputs. Locomotion is initiated by a tonic excitatory drive signal from mesencephalic locomotor region (MLR). Indeed, the architecture of Rybak and McCrea seems very efficient to control the articulation of a robot.

Biologists have shown the fundamental role of rhythmic neurons in CPG architectures. They pointed out the four main behaviors of a biological rhythmic neuron: quiescent, plateau potentials, damped oscillating, and oscillating. For many years, artificial neural networks have been studied to control biped or other legged robot, in robotics. Many studies have focused on dynamic neural controllers based on continuous-time recurrent neural networks (CTRNN) because of their ability to be universal approximators (Beer and Chiel, 1992; Gallagher et al., 1996; Brunel, 2000; Vogels and Abbott, 2005; Beer, 2006). CTRNNs are also of interest in bio-inspired control, thanks to their capability to reproduce the full range of nerve cell phenomenology qualitatively. In CTRNNs (which are a type of RNN) the activities of neurons are determined by current synaptic inputs and by the history of neural states. This property allows CTRNNs to be superior to discrete time RNN models in the production of continuous sensori-motor sequences (Yamashita and Tani, 2008). CTRNNs make it possible to show adaptivity properties based on homeostatic plastic mechanisms (Hoinville and Henaff, 2004; Williams, 2007). Nevertheless, the neuron model used in CTRNN does not seem realistic enough for locomotion genesis because it is not an intrinsic rhythmic model. Indeed, as has shown Beer, the probability of oscillations of a CTRNN depends on the number of neurons and on certain synaptic connections.

Another well-known neuron model was suggested by Matsuoka (1985). This model has been largely used for many years in locomotion especially in bipedal walking (Taga et al., 1991; Endo et al., 2005; Matsubara et al., 2006; Lei et al., 2008; Woosung et al., 2008). This model is more biological than the CTRNN because it takes into account the adaptation mechanism of neurons which is critical in the generation of oscillations. However, in this model, it is not easy to control its rhythmic behavior to have the four main behaviors of a biological rhythmic neuron.

To get a flexible and dynamical legs coordination to generate the ongoing motion Pitti et al. (2009) show that CPGs can be seen as chaotic controllers that maintain the dynamical phase synchronization between them and the robot body dynamics. One way to achieve this aim is to use relaxation oscillators. This type of CPG models has been used frequently to control biped robots (Dutra et al., 2003; De Pine Filho et al., 2005), quadruped (Buchli et al., 2006; Liu et al., 2009), or snake robot (Conradt and Varshavskaya, 2003). Nevertheless, to our knowledge, the robot is often in an open-loop relative to its environment; there are no sensory neurons to improve the rhythmic activity, especially no sensory feedback from the ground. Then, from our point of view, the fundamental biological principle of sensory/motor loop that maintains the dynamical phase synchronization between the controller and the robot interacting with its environment is not respected.

The originality of this paper resides in the introduction of afferent exteroceptive and proprioceptive signals on a CPG controller based on intrinsic chaotic rhythmic neurons. The CPG follows the three levels organization of Rybak and McCrea (pattern generation level, PF level, and motor level). This architecture seems very efficient to control the articulation of a robot. Due to the number of articulations (a biped robot usually has six joints on each leg), it is much too complex to design a global neural controller of a biped robot. Nevertheless, we suggest to preliminarily study how one can control, with such kind of neurocontroller, the rhythmic pattern, and the velocity of a two-joint planar robot leg that slips on a rail. The rhythmic neuron is modeled as the one proposed by Rowat and Selverston (1993, 1997) which is a generalization of the Van der Pol's relaxation oscillator. This model is interesting because one single neuron is intrinsically capable to generate different kinds of oscillatory signals. Indeed, it has two state-parameters which allow to control and modify the neuron phase among the four biological cited above. RS neurons were used for the first time in a robotic locomotion problem in order to design (with genetic algorithms) neurocontrollers for various multi-legged robots (Hoinville, 2007). This work showed that the RS neuron model is very well suited to generate adaptive rhythmic locomotion for legged robots because it may show properties of plasticity through its parameters.

This paper is organized as follows. In the second section, the model of neuron proposed by Rowat and Selverston is briefly described and simulations that illustrate its efficiency are carried out. First, an architecture based on two RS neurons is proposed for a one-joint neurocontroller and then for a two-joint planar leg that slips on a rail. In Section "Results and Discussion," simulations show the interest of RS model: firstly, the easiness to control the rhythmic of the walking gait of the leg, and secondly the importance of ground/foot sensory feedback in the fluidity of rhythmic activity. In Section "Conclusion," conclusion and future works are presented.

MATERIALS AND METHODS

ROWAT–SELVERSTON RHYTHMIC NEURON MODEL

A reminder on RS model

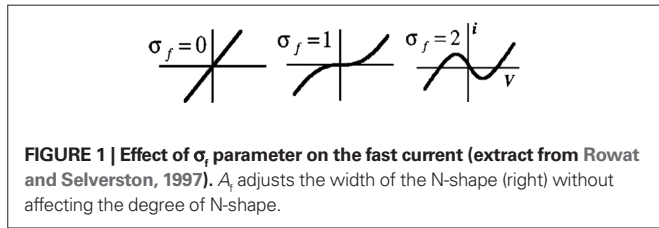
Biological neurons with several ionic channels are complex, hence difficult to model. Rowat and Selverston (1993) present a simple model of a neuron for which two groups of currents are identified: a fast current and a slow current, each defined by a first order differential equation. Fast current is defined by Eq. 1 and slow current by Eq. 2.

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

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

with $\tau_m < \tau_s$, τ_m is the time constant of the neuron membrane, τ_s is the time constant of slow currents activation. I_{inj} is the injected current, V the cellular membrane voltage, and q the slow current.

$F(V, \sigma_f)$, defined in Eq. 3, is a non-linear current–voltage function for the fast current. We see in **Figure 1** that the fast current shape $F(V, \sigma_f)$ can be linear ($\sigma_f = 0$), non-linear but bijective ($\sigma_f = 1$), and non-linear and non-bijective ($\sigma_f = 2$). $F(V, \sigma_f)$ is a fundamental part of the RS model because this function induces different behaviors for the neuron (damped oscillating, plateau potentials, or oscillating) following the value of the fast current gain, σ_f .



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

Meanwhile, the steady-state value of the slow current is linear in V , with conductance σ_s :

$$q_\infty(V) = \sigma_s V \quad (4)$$

In Eqs. 1 and 2, q , V , and I_{inj} have the dimension of an electrical current. It means that a current is represented by enough potential to drive the current through the membrane leak conductance, g_L , which is a constant. For deep mathematical study of the model please refer to (Rowat and Selverston, 1993, 1997).

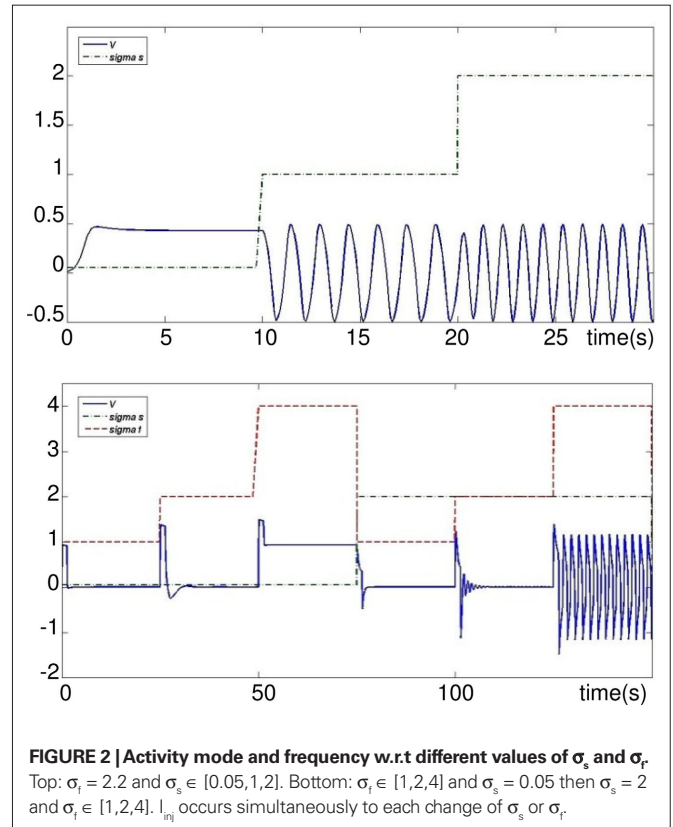
Simulation and study of the cell model

Studying the equations of current for the cell model shows that the frequency of oscillation is basically determined by the value of σ_s , and its amplitude determined by the value of A_f . Moreover, if σ_f is reduced to $\sigma_f < 1$ while σ_s is large enough, the cell does not oscillate anymore. However, if $\sigma_f > 1$, but σ_s is reduced below a certain value, the model does not oscillate but shows plateau potentials.

In the following, the neuron model of Rowat and Selverston is simulated under MATLAB-Simulink. The activity of this model for different values of variables σ_s , σ_f , τ_s , and τ_m is explained. The primary values of these variables are $\tau_s = 1$ s, $\tau_m = 0.05$ s, and $A_f = 2$. Input I_{inj} is a pulse of amplitude 1 and duration of 1 s. In **Figure 2** (up) it can be seen that for a rather high value of σ_f ($=2.2$ in this case), but small values of σ_s , the cell does not oscillate and shows plateau potentials. As the value of this parameter increases above one, the cell starts to oscillate, and the frequency of this oscillation is proportional to the value of σ_s . On the other hand, if σ_s is large but σ_f small, the neuron output shows damped oscillations (see **Figure 2**, bottom).

With different combinations values of σ_s and σ_f , the neuron shows different modes of activity (quiescent, damped oscillating, plateau potentials, or oscillating). Different zones of this activity can be defined as shown in the bifurcation figure diagram of **Figure 3**.

This diagram has been obtained manually and through a detailed study of different combinations of values for parameters σ_f and σ_s . Note that in this bifurcation study, the transition time to toggle from one activity mode to another is not taken into account. Nevertheless, one can see in **Figure 2** that this time seems rather short. Therefore, the RS model allows for the simulation of the fundamental intrinsic functional properties met in biological networks which generate rhythmic gaits. In particular, it is a plastic intrinsic model through its parameters σ_f and σ_s , which can be modulated by internal or external signals. In Section ‘‘Conclusion’’ we will show that this plasticity is very efficient to produce adaptive walking gaits for legged robot.



CPG MODEL FOR ONE JOINT ARTICULATION

The neural architecture suggested in this paper (see **Figure 4**) uses two Rhythmic Neurons, two Sensory Neurons for feedback from angular position, and two Velocity Sensory Neurons that code velocity feedback of the joint. Two cells which we call Motoneurons modulate the output of each Rhythmic Neurons depending on Sensor Neuron outputs.

Sensory neurons are modeled as stretch receptors similar to those in (Geng et al., 2006; Wörgötter et al., 2006; Manoonpong et al., 2008). This is a model inspired by a sensor neuron proposed by Wadden and Ekeberg (1998).

Outputs from position sensor neurons are given by:

$$v_{ES} = \tanh(k_p(\theta_{ES} - \theta)) \quad (5)$$

$$v_{FS} = \tanh(k_p(\theta - \theta_{FS})) \quad (6)$$

Where the parameter θ_{ES} limits the joint articulation to maximum angle of extension, θ_{FS} defines the minimum value of flexion and $k_p = 5$.

Outputs from velocity sensor neurons are given by:

$$v_{FVS} = \tanh(-k_d \cdot \dot{\theta}) \quad (7)$$

$$v_{EVS} = \tanh(k_d \cdot \dot{\theta}) \quad (8)$$

Where $k_d = 5$.

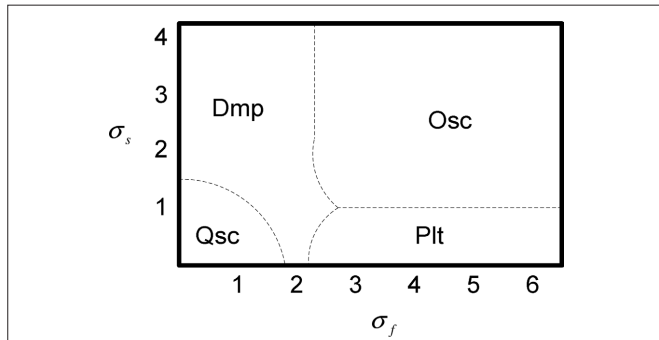


FIGURE 3 | Bifurcation diagram for values of σ_f and σ_s . Qsc, quiescent; Dmp, damped oscillating; Plt, plateau potentials; Osc, oscillating. $\tau_s = 1$ s, $\tau_m = 0.05$ s, $A_i = 2$.

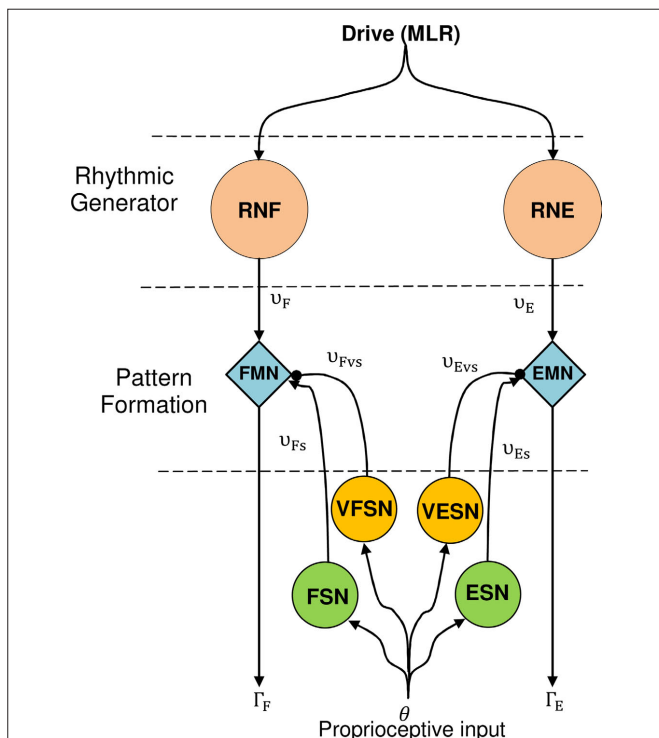


FIGURE 4 | The CPG model for one articulation. Proprioceptive inputs come from the angular position of the articular joint. RNF and RNE are Rowat–Selverston rhythmic neurons (flexor and extensor). EMN and FMN are extensor and flexor motor neurons, ESN and FSN are extensor, and flexor sensory neurons. VFSN and VESN are the velocity extensor and flexor sensory neurons.

Like in (Wörgötter et al., 2006), the torque applied on the articulation is given by:

$$\gamma = \Gamma_E - \Gamma_F \tag{9}$$

Where extensor and flexor torques are produced by motoneurons FMN and EMN:

$$\Gamma_E = \Gamma_{\max} v_E (v_{ES} - v_{EVS}) \tag{10}$$

$$\Gamma_F = \Gamma_{\max} v_F (v_{FS} - v_{FVS}) \tag{11}$$

Simulation of 1-DOF

In this subsection, the architecture described in the previous sub-section is applied to a leg with 1-DOF, in MATLAB-Simulink (Figure 5). All simulations in this paper are in the plane.

The leg is simulated like a pendulum, fixed on the hip and is not in contact with the ground. $\theta_{ES,h}$ and $\theta_{FS,h}$ define respectively maximum extension and flexion angles of the hip.

Drive MLR is an one pulse signal of 0.1 s that is sufficient enough to trigger the CPG. Outputs of the two motor neurons RNE and RNF are shown Figure 6, torque and angular joint on Figure 7.

It can be seen (on Figure 6) that after a transition phase, the two motor neurons EMN and FMN oscillate in opposite phase on a stable regime. This transition is also visible on Figure 7 takes 0.75 s. The torque applied to the hip and its angle show that, after the transition, the joint maintains in a stable behavior the defined extension and flexion angular constraints $\theta_{ES} = 120^\circ$, $\theta_{FS} = 60^\circ$. In addition, one can clearly see how the torque is synchronized to the sensory neuron. Note that this oscillation maintains itself indefinitely, which means that our CPG generates a robust limit cycle.

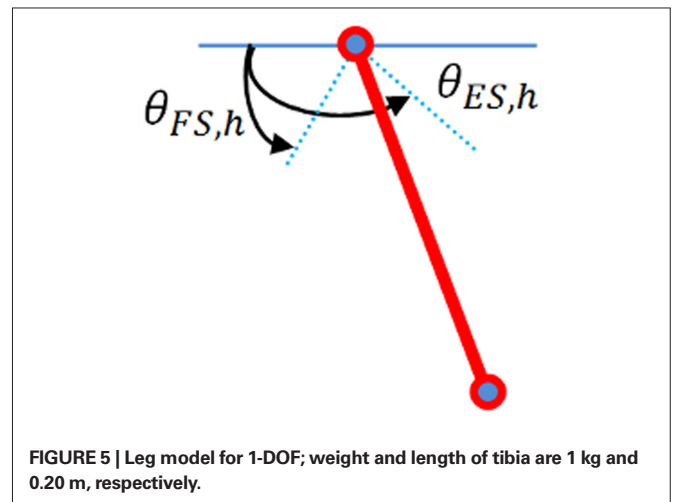


FIGURE 5 | Leg model for 1-DOF; weight and length of tibia are 1 kg and 0.20 m, respectively.

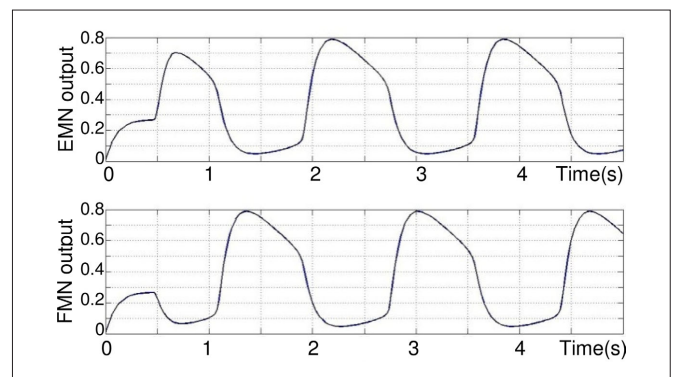
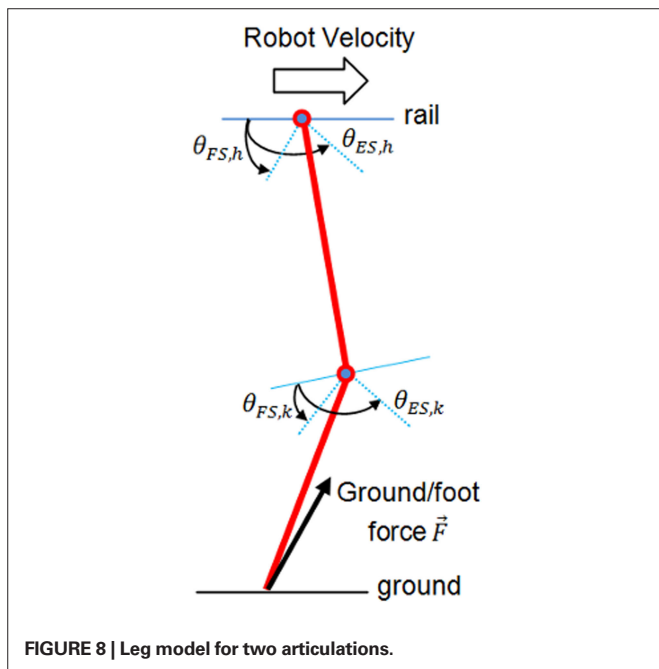
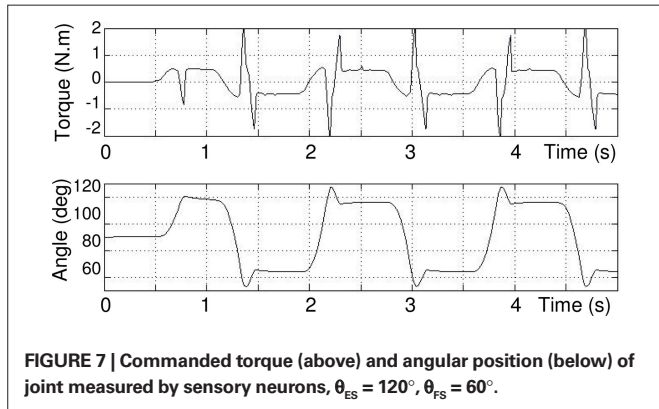


FIGURE 6 | Oscillations in Rowat–Selverston-type neurons implemented in extensor and flexor joint controllers, respectively. The output is normalized. $\tau_m = 0.05$ s, $\tau_s = 1$ s, $\tau_m = 0.05$ s, $\sigma_f = \sigma_s = 3$, $A_i = 5$.



CPG MODEL FOR ONE LEG WITH TWO PLANAR JOINTS

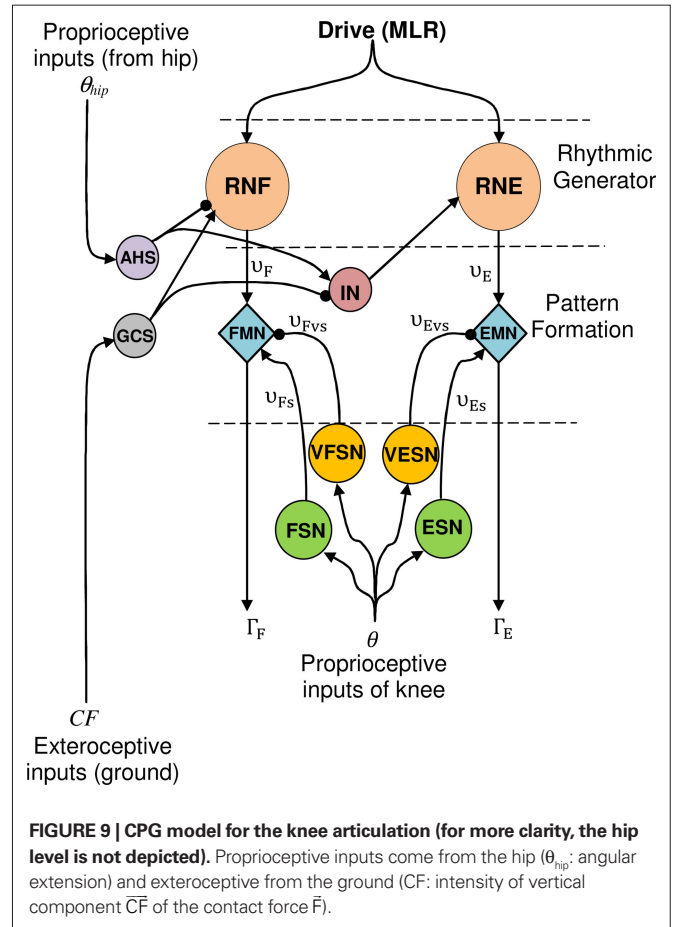
Locomotion problem

In this section, the aim is to control a planar leg which slides on a rail (see **Figure 8**). This leg has a punctual foot, one joint at hip level, and another on the knee. The hip is joined to an ideal rail (no friction), and slides forward by pushing the point foot on to the ground (like in skating). Thigh and tibia are both 20 cm long and weight 1 kg. The problem is similar to Wörgötter et al.’s (2006) experiment, in the way that both models are in the sagittal plane and are based on sensory feedback from the ground (forces) and articular joints.

Leg CPG model with afferent sensory inputs

To control this leg with two articulations, one CPG presented before (**Figure 4**) is needed for each articulation. It is necessary here to coordinate the angular movement of the knee with the hip. For this purpose, two sensory neurons are used (see Eqs 8 and 9 respectively and **Figure 9**).

The first one, the anterior hip sensor (AHS), measures the anterior hip angle and excites the RG level of the knee. The second, called ground contact sensor (GCS), is a force sensor neuron that



senses the intensity of vertical component \bar{CF} of the contact force \bar{F} between the ground and the foot. GCS excites knee Rhythmic Flexor Neuron and inhibits knee Rhythmic Extensor Neuron. Outputs of AHS and GCS neurons are given by:

$$v_{AHS} = \tanh(k_{ah}(\theta_{ES,h} - \theta_{hip})) \tag{12}$$

$$v_{GCS} = \tanh(k_{gc}(F_v)) \tag{13}$$

These two sensory loops work as follows. As long as the hip angle is smaller than its extreme value ($\theta_{ES,h}$), the AHS fires and inhibits the knee extensor (if $\theta_{hip} < \theta_{ES,h}$, then $v_{AHS} > 0$ and inhibits the RNF). When the hip reaches its maximum extension AHS liberates the knee extensor and consequently the knee starts to extend. Meanwhile, the GCS is activated by the ground force applied to the foot as soon as the foot touches the ground GCS is activated, inhibits the knee extensor and strengthens the knee flexor. So the knee extensor neuron is inhibited, either when the foot is in contact with the ground or when the thigh is not at its maximum extension phase. Thus, an interneuron IN is introduced. It is activated if at least one of the two sensory neurons AHS and GCS is active. This neuron is defined as Eq. 14 and its output Σ has a value between 0 and 1 according to its input v

$$\Sigma = \frac{1}{1 + e^{(1-v)}} \text{ and } v = v_{AHS} - v_{GCS} \tag{14}$$

In case of a two legged robot, sensory neurons at hips level are also needed to coordinate the two legs. In this paper the architecture is limited to a single leg, then sensory neurons are not added to the hip and its architecture is the same as the one described in the previous Section “Results and Discussion.”

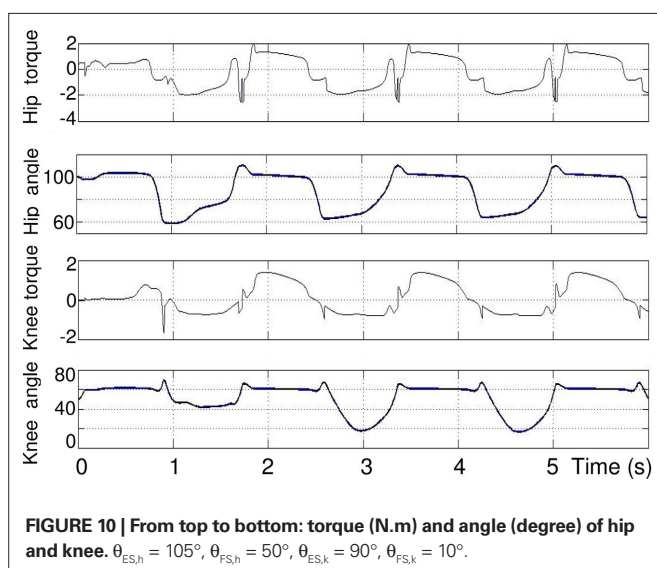
RESULTS AND DISCUSSION

In **Figure 10** the applied torque and the correspondent angle of each joint (hip and knee) are presented.

As it can be seen in the second and third plots, when the hip reaches its maximum extension angle ($\theta_{ES,h}$), the extensor in the knee is activated and thus the torque applied to knee joint increases. As for one articulation, there exists a transitory phase, that lasts about 2 s, which is the time needed to stabilize the permanent rhythmic activity. The length of this temporary phase also depends on the initial position and angle. The hip rhythmic activity has the same appearance as the one in Section “Modulation of the Rhythmic Activity” except frequency and amplitude that changed because of coordination setting. The rhythmic activity of the knee has the same frequency as that of the hip. In the extension phase of the knee cycle, the foot touches the ground, so the tibia stops extending before reaching its maximum extension angle $\theta_{ES,k} = 90^\circ$.

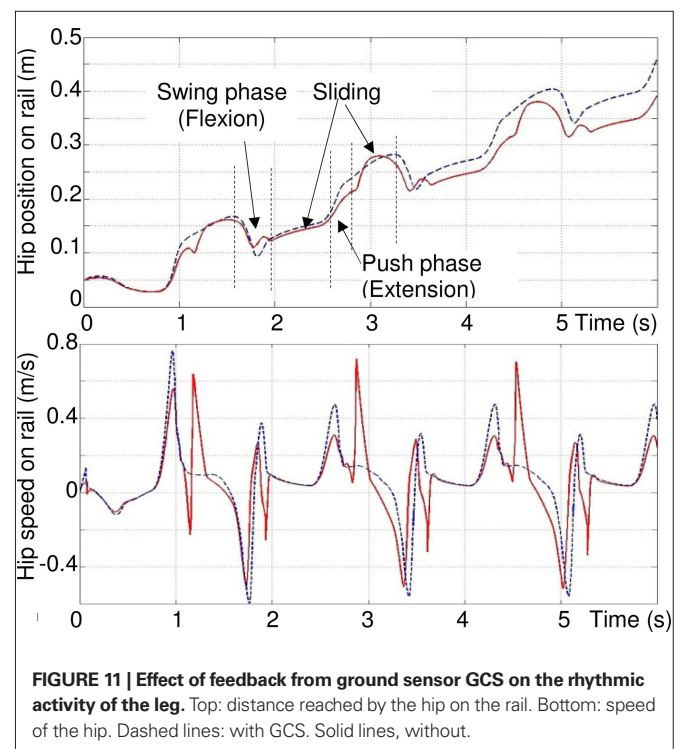
GROUND/FOOT CONTACT EFFECT ON THE RHYTHMIC ACTIVITY

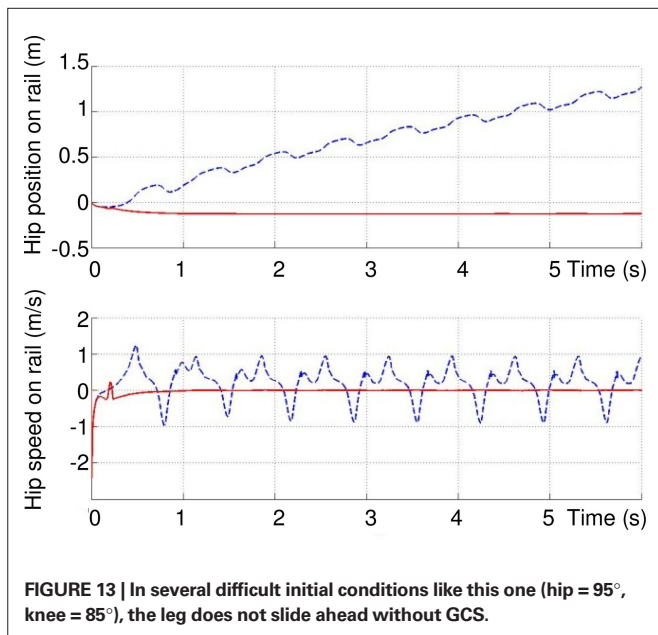
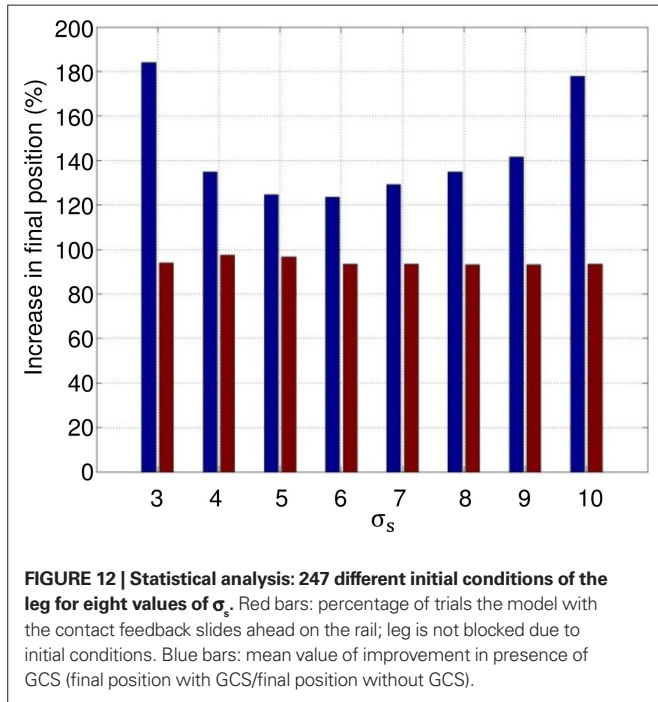
Figure 11 compares the hip position and velocity in presence of ground contact neuron GCS (dashed lines) and without GCS (solid lines). This figure illustrates the importance of the GCS in the leg, already proven to exist in biology (Cattaert and Le Ray, 2001). The figure (top) shows that the movement of the hip is smoother and faster in presence of GCS. After six seconds, the distance reached by the hip on the rail is longer with GCS than without. The figure (bottom) shows that there is less variation of speed when there is a GCS sensory loop.



There are three phases in one walking cycle. From 0.8 to 1.7 s, due to the push of the foot against the ground (extension of articulations), the hip velocity on the rail increases rapidly (acceleration): and the hip slips on the rail. The maximum velocity is highest with GCS. Then, from 1.7 to 2 s, it is the swing phase of the leg. In this phase, the rapid flexion of the two articulations induces (without GCS, at around 1.3, 1.7 and 1.9 s and 1.7 s with GCS) an inverse dynamic effect which moves back the hip (negative speed) because the friction forces on the rail are low. Finally, from 2 to 2.6 s, the hip continues to slip and the foot is in contact with the ground and pushes again (positive speed). The sudden oscillation (rapid negative/positive peaks at $t = 1.3, 1.9, 3.6$ s...) are rejected with GCS. Other negative peaks ($t = 1.7, 3.5, 5.1$ s) are the same with and without GCS, because of the rapid flexion of the leg.

Figure 12 shows that the effect of the GCS is statistically larger when the initial conditions are varied. Two hundred forty-seven initial conditions of the leg are simulated with and without the contact sensor: the hip angle is incremented from 60° to 120° , with intervals of 5° and the knee angle, with respect to the thigh, for a range of $0-90^\circ$. Simulation is run for 6 s in each case, for eight integer values of σ_s varying from 3 to 10. The average speed and the final position of the hip are measured. Results show that in more than 93% of the trials (red bars in **Figure 12**), the model with the contact feedback slides ahead on the rail (for some difficult trials, the leg remains blocked), while this number is a little lower for the model without GCS, yet around 90% (not depicted). The average improvement between the final position reached at $t = 6$ s in the presence of CGS is at least 20%, and can go up to 80% for certain values of σ_s (blue





bars). There have been some cases where, without GCS, the leg does not slide ahead (difficult initial configuration), like the one depicted in Figure 13.

All these results show that the sensory feedback improves the dynamic coordination of the leg. Therefore, the dynamical interaction between the CPGs, the robot and the environment favors rhythm production and phase synchronization as mentioned in (Pitti et al., 2009).

MODULATION OF THE RHYTHMIC ACTIVITY

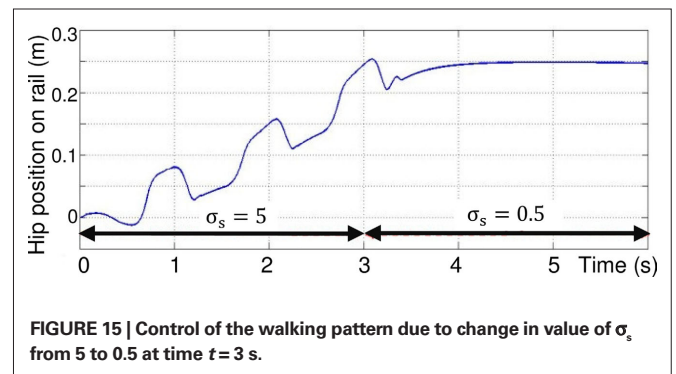
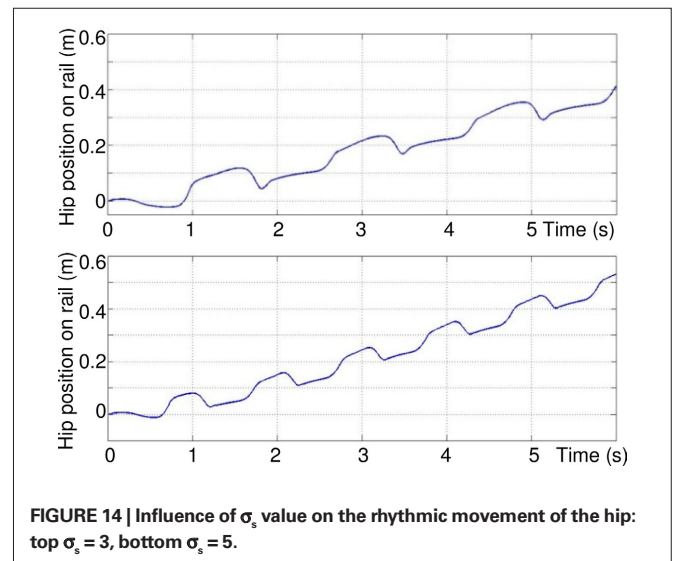
Figure 14 shows the influence of conductance σ_s values on the walking cycle. By increasing the value of this parameter for the four rhythmic neurons, the frequency of the walk increases.

In Figure 15, at time $t = 3$ s, the value of this parameter decreases from 5 to 0.5, so that the activity mode of the motor neurons change from oscillatory to plateau potentials. This switch in activity mode stops the sliding of the leg.

These last two simulations show that the walking cycle can easily be controlled by the modulation of a single parameter in rhythmic neurons of the proposed architecture of Figure 9, and then stops the robot walking.

Note that the transition from one walking gait to another is continuous and stable because of the robustness of the Rowat–Selverston neuron model. Hence, it is logical to assume that this modulation can be applied from a high-level controller to modify, for example, the desired velocity of the legged robot.

Figure 16 shows the walk on an uneven terrain, for values of σ_s that increase robot velocity. Three vertical stairs appear between $t = 4$ and 8 s. This simulation clearly shows the robustness of the neural controller. Indeed, the rhythmic pattern is



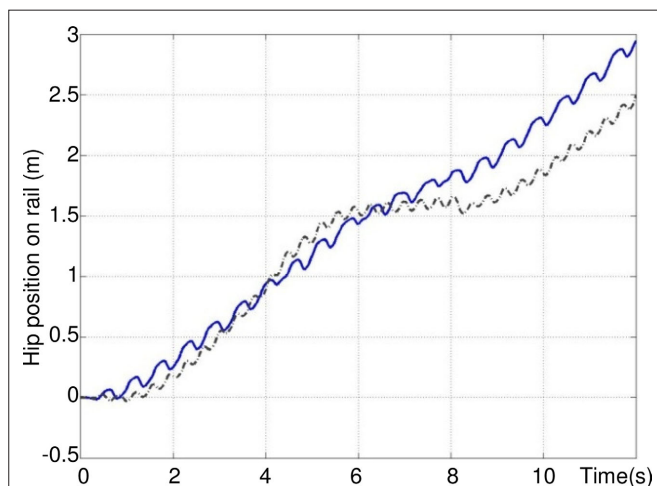


FIGURE 16 | Control of the walking pattern on an uneven terrain for $\sigma_s = 10$ (dashed black) and $\sigma_s = 20$ (solid blue). At time $t = 4$ s a vertical up stair of 7 cm appears (ground is at +7 cm), at $t = 6$ s, a vertical down stair of 4 cm appears (ground is at +3 cm), at $t = 8$ s, a vertical down stair of 3 cm appears (ground is at 0 cm).

REFERENCES

- Beer, R. (2006). Parameter space structure of continuous-time recurrent neural networks. *Neural Comput.* 18, 3009–3051.
- Beer, R., and Chiel, H. (1992). A distributed neural network for hexapod Robot Locomotion. *Neural Comput.* 4, 356–365.
- Brunel, N. (2000). Dynamics of networks of randomly connected excitatory and inhibitory spiking neurons. *J. Physiol.* 94, 445–463.
- Buchli, J., Iida F., and Ijspeert A. (2006). “Finding resonance: adaptive frequency oscillators for dynamic legged locomotion,” in *Proceeding of IEEE International Conference on Intelligent Robots and Systems (IROS)*, Beijing, 3903–3910.
- Büschges, A. (2005). Sensory control and organization of neural networks mediating coordination of multisegmental organs for locomotion. *J. Neurophysiol.* 93, 1127–1135.
- Cattaert, D., and Le Ray, D. (2001). Adaptive motor control in the crayfish. *Prog. Neurobiol.* 63, 199–240.
- Conradt, J., and Varshavskaya, P. (2003). “Distributed central pattern generator control for a serpentine robot,” in *Proceedings of the International Conference on Artificial Neural Networks (ICANN)* (Istanbul, Turkey), 338–341.
- Cruse, H., Bartling, C., Cymbalyuk, G., Dean, J., and Dreifert, M. (1995). A modular artificial neural net for controlling a six-legged walking system. *Biol. Cybern.* 72, 421–430.
- Cruse, H., Kindermann, T., Schumm, M., Dean, J., and Schmitz, J. (1998). Walknet—a biologically inspired network to control six-legged walking. *Neural Netw.* 11, 1435–1447.
- De Pine Filho, A. C., Dutra, M. S., and Raptopoulos, L. S. C. (2005). Modeling of a bipedal robot using mutually coupled Rayleigh oscillators. *Biol. Cybern.* 92, 1–7.
- Di Prisco, G. V., Pearlstein, E., Le Ray, D., Robitaille, R., and Dubuc, R. (2000). A cellular mechanism for the transformation of a sensory input into a motor command. *J. Neurosci.* 20, 8169–8176.
- Dutra, M. S., De Pina Filho, A. C., and Romano, V. F. (2003). Modeling of a bipedal locomotor using coupled nonlinear oscillators of Van der Pol. *Biol. Cybern.* 88, 286–292.
- Endo, G., Nakanishi, J., Morimoto, J., and Cheng, G. (2005). “Experimental studies of a neural oscillator for biped locomotion with QRIO,” in *Proceedings of IEEE International Conference on Robotics and Automation, (ICRA) 2005*, Barcelona, 596–602.
- Gallagher, J., Beer, R., Espenschied, K., and Quinn, R. (1996). Application of evolved locomotion controllers to a hexapod robot. *Rob. Auton. Syst.* 19, 95–103.
- Geng, T., Porr, B., and Wörgötter, F. (2006). A reflexive neural network for dynamic biped walking control. *Neural Comput.* 18, 1156–1196.
- Gill, M. D., and Skorupski, P. (1996). Modulation of spontaneous and reflex activity of crayfish leg motor neurons by octopamine and serotonin. *J. Neurophysiology* 76, 3535–3549.
- Grillner, S., Deliagina, T., Ekeberg, O., El Manira, A., Hill, R. H., Lansner, A., Orlovsky, G. N., and Wallen, P. (1995). Neural networks coordinating locomotion and body orientation in lamprey—biological and mathematical models. *J. Neurophysiol.* 18, 270–279.
- Grillner, S., and Wallen, P. (2002). Cellular bases of a vertebrate locomotor system—steering, intersegmental and segmental co-ordination and sensory control. *Brain Res. Rev.* 40, 92–106.
- Hoinville, T. (2007). *Évolution de contrôleurs neuronaux plastiques. De la locomotion adaptée vers la locomotion adaptative*. Ph.D. thesis (In French), University of Versailles St Quentin, Vélizy.
- Hoinville, T., and Henaff, P. (2004). “Comparative study of two homeostatic mechanisms in evolved neural controllers for legged locomotion,” in *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems*, Sendai, 2624–2629.
- Ijspeert, A. (2008). Central pattern generators for locomotion control in animals and robots: a review. *Neural Netw.* 21, 642–653.
- Lei, L. G., Habib, M. K., Watanabe, K., and Izumi, K. (2008). Central pattern generators based on Matsuoka oscillators for the locomotion of biped robots. *Artif. Life Rob.* 12, 264–269.
- Liu, C., Qijun, C., and Jiaqi, Z. (2009). “Coupled Van Der Pol oscillators utilised as central pattern generators for quadruped locomotion,” in *Proceedings of International Conference on Chinese control and Decision Conference*, Shanghai, 3677–3682.
- Manoonpong, P., Pasemann, F., and Wörgötter, F. (2008). Sensor-driven neural control for omnidirectional locomotion and versatile reactive behaviors of walking machines. *Rob. Auton. Syst.* 56, 265–288.
- Marchand, A. R., Barnes, W. J. P., and Cattaert, D. (1997). Primary afferent depolarizations of sensory origin within contact-sensitive mechanoreceptive afferents of a crayfish leg. *J. Neurophysiol.* 77, 3340–3354.
- Marder, E., and Bucher, D. (2001). Central pattern generators and the control of rhythmic movements. *Curr. Biol.* 11, 986–996.
- Matsubara, T., Morimoto, J., Nakanishi, J., Sato, M., and Doya, K. (2006). Learning CPG-based biped locomotion with a policy gradient method. *Rob. Auton. Syst.* 54, 911–920.
- Matsuoka, K. (1985). Sustained oscillations generated by mutually inhibiting neurons with adaptation. *Biol. Cybern.* 52, 367–376.
- Mc Crea, D. A., and Ryback, I. A. (2008). Organization of mammalian locomotor rhythm and pattern generation. *Brain Res. Rev.* 57, 134–146.
- Nadim, F., and Manor, Y. (2000). The role of short-term synaptic dynamics in motor control. *Curr. Opin. Neurobiol.* 10, 683–690.
- Pitti, A., Lungarella, M., and Kuniyoshi, Y. (2009). Generating spatiotemporal joint torque patterns from dynamical synchronization of

not much disturbed, the velocity decreases, and the neural CPG finds a normal rhythmic activity when the ground level is 0 cm at $t = 8$ s.

Compared to **Figure 14**, for $\sigma_s = 20$ the speed is three times greater than for $\sigma_s = 5$.

CONCLUSION

This paper presents and reproduces the functioning mechanism of a neural CPG architecture based on Rowat and Selverston neuron. First, we have shown that this model can be easily used to produce a controllable walking gait by modulation of the single conductance parameter σ_s of each rhythmic neuron. The proposed CPG architecture is based on fundamental biological principles consisting of rhythmic level, PF level, proprioceptive, and exteroceptive sensory loops. Second, through simulations, we have shown that the ground force sensory feedback has an important role in the quality of the rhythmic activity.

Future work will be to extend this architecture to a biped robot for which the modulation of a membrane neuron conductance could create adaptive locomotion. For this purpose, this parameter must be modulated with respect to external parameters of the robot, such as contact forces of each foot, as well as internal properties, like coordination between the legs and balance.

- distributed pattern generators. *Front. Neurobotics* 3:2. doi: 10.3389/neuro.12.002.2009
- Rowat, F., and Selverston, I. (1993). Modeling the gastric mill central pattern generator of the lobster with a relaxation-oscillator network. *Neurophysiology* 70, 1030–1053.
- Rowat, P., and Selverston, A. (1997). Oscillatory mechanisms in Pairs of neurons connected with fast inhibitory synapses. *J. Comput. Neurosci.* 4, 103–127.
- Taga, G., Yamaguchi, Y., and Shimizu, H. (1991). Self-organized control of bipedal locomotion by neural oscillators in unpredictable environment. *Biol. Cybern.* 65, 147–159.
- Vogels, T. P., and Abbott, L. F. (2005). Signal propagation and logic gating in networks of integrated-and-fire neurons. *J. Neurosci.* 25, 10786–10795.
- Wadden, T., and Ekeberg, O. (1998). A neuro-mechanical model of legged locomotion: single leg control. *Biol. Cybern.* 79, 161–173.
- Williams, H. J. (2007). Homeostatic plasticity improves signal propagation in continuous-time recurrent neural networks. *Biosystems* 87, 252–259.
- Woosung, Y., Chong, N. Y., Ra, S., Chang, H. K., and Bum, J. Y. (2008). “Self-stabilizing bipedal locomotion employing neural oscillators,” in *Proceedings of 8th IEEE-RAS International Conference on Humanoid robots*, Daejeon, 8–15.
- Wörgötter, F., Geng, T., and Porr, B. (2006). Fast biped walking with a sensor-driven neuronal controller and real-time online learning. *Int. J. Rob. Res.* 25, 243–259.
- Yamashita, Y., and Tani, J. (2008). Emergence of functional hierarchy in a multiple timescale neural network model: a humanoid robot experiment. *Comput. Bio.* 4, e1000220.
- Conflict of Interest Statement:** The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Received: 14 March 2010; accepted: 06 December 2010; published online: 29 December 2010.

Citation: Amrollah E and Henaff P (2010) On the role of sensory feedbacks in Rowat–Selverston CPG to improve robot legged locomotion. *Front. Neurobot.* 4:113. doi: 10.3389/fnbot.2010.00113

Copyright © 2010 Amrollah and Henaff. This is an open-access article subject to an exclusive license agreement between the authors and the Frontiers Research Foundation, which permits unrestricted use, distribution, and reproduction in any medium, provided the original authors and source are credited.

- 9.8 Experience-based Learning Mechanism for Neural Controller Adaptation : Application to Walking Biped Robots, Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2009**

Experience-based Learning Mechanism for Neural Controller Adaptation: Application to Walking Biped Robots

John Nassour, Patrick Hénaff, Fethi Ben Ouezdou and Gordon Cheng

Abstract—Neurobiology studies showed that the role of the Anterior Cingulate Cortex of the brain is primarily responsible for avoiding repeated mistakes. According to vigilance threshold, which denotes the tolerance to risks, we can differentiate between a learning mechanism that takes risks, and one that averts risks. The tolerance to risk plays an important role in such learning mechanism. Results have shown the differences in learning capacity between risk-taking and risk avert behaviors. In this paper we propose a learning mechanism that is able to learn from negative and positive feedback. It is composed of two phases, evaluation and decision-making phase. In the evaluation phase, we use a Kohonen Self Organizing Map technique to represent success and failure. Decision-making is based on an early warning mechanism that enables to avoid repeating past mistakes. Our approach is presented with an implementation on a simulated planar biped robot, controlled by a reflexive low-level neural controller. The learning system adapts the dynamics and range of a hip sensor neuron of the controller in order for the robot to walk on flat and slope terrain. Results show that success and failure maps can learn better with a threshold that is more tolerant to risk. This gives rise to robustness to the controller even in the presence of slope variations.

I. INTRODUCTION

Some cognitive studies have identified an early warning system in the human brain that can avoid to make past mistakes again. They have shown how the brain remembers details about past dangers [1]. An activity was found in the Anterior Cingulate Cortex (ACC) after making mistakes [2]. This cortex area works as an early warning system that adjusts the behavior to avoid dangerous situations. It responds not only to the sources of errors (external error feedback), but also to the earliest sources of error information available (internal error detection) [3]. It becomes active in proportion to the occurrence likelihood of an error [4][5][6]. Therefore it can learn to identify situations where humans may make mistakes, and then help to avoid such situations to occur[2]. It learns to predict error likelihood even for situations where no error occurs previously[7]. Through the observation of particular areas located in cerebral cortex in the brain responsible for cognitive control, neuropsychological studies demonstrated a switching in human learning strategies around the age of twelve years. This switch from learning with positive feedback to learning with negative feedback probably comes from the combination of brain maturing and experience[8].

F.B. Ouezdou and J. Nassour are with Versailles Saint Quentin University-France (email: fethi.benouezdou, nassour@lsv.uvsq.fr).

P. Hénaff is with Versailles Saint Quentin University and Cergy Pontoise University-France (email:patrick.henaff@uvsq.fr).

G. Cheng is with Technical University-Munich (email: gordon.cheng@ieee.org).

Our work aims to produce an early warning mechanism that can help to avoid repeating past errors in the generation of walking patterns for humanoid robots. It is necessary for such a mechanism to have experience in mistakes and other experience in success, in order to evaluate new situations before taking any decision and carrying out the test on the robot. This mechanism of selection allows to determine the state space of parameters in the zone of success and also in the zone of conflict. It is used to adapt the dynamics and range of a hip sensor neuron in a neural reflexive controller, proposed by F.Wörgötter [17], for simulated planar biped robot in order to avoid falls when the terrain slope varies.

This paper is structured as follows. The second section presents the principles of our learning mechanism in details, and introduces the concept of vigilance. The third section describes the neural reflexive controller based on sensor motor neurons proposed by F.Wörgötter. In the fourth section we use this mechanism to detect the domain of viability of the controller for walking on flat terrain and then on sloped terrain, where the effect of vigilance threshold on learning was investigated. Finally, we conclude this paper with some research perspectives.

II. LEARNING MECHANISM

The objectives of this learning mechanism is to adapt parameters of a low level controller and detecting its domain of viability, which brings more adaptation to external and internal perturbations. We designate by V the state space of those influent parameters. The mechanism must be able to learn from negative feedback (failure) and positive feedback (success), therefore it must have experience of success and other experience of failure in the state space V . As each vector \vec{v} from V leads to either success or failure, the mechanism will evaluate whether this vector belongs to the success domain or to the failure domain. The decision mechanism “go” or “nogo” described in [9] works as an early warning system similar to that in the Anterior Cingulate Cortex [2][7]. The learning architecture is then based on these two mechanisms and works as shown in Fig.1.

A. Success-failure evaluation

To represent the knowledge in success and in failure, we define two independent neural networks that are well-known Self Organizing Maps, proposed by Kohonen [10]. Success map learns in case of success trials, and failure map learns in case of failure trials. During the learning, the two maps will be self-organized in the state space that will be therefore divided into three zones: a zone of success represented by

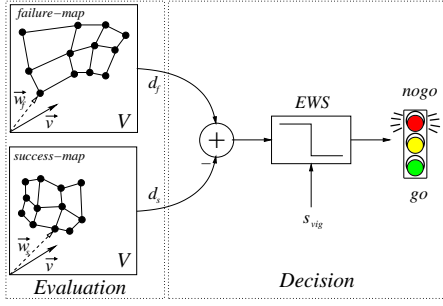


Fig. 1. Learning mechanism with evaluation and decision phases.

success map, a zone of failure represented by failure map, and a zone of conflict that corresponds to the interference between the two maps. The evaluation of any vector \vec{v} from space V belonging to success or failure is defined by the distance between \vec{v} and each map. The distance of a vector with a map is the distance between this vector and the neuron that is closest to it in the state space (the neuron winner). For each \vec{v} we have therefore two distances: one to success map called d_s , and another to failure map called d_f .

B. Decision mechanism

For a vector \vec{v} , the comparison between the distance with success map d_s and the distance with failure map d_f leads to an expected result in the case where the vector was passed to the low level controller (trial). According to expected results, if it may lead to failure, then an Early Warning Signal EWS becomes active to avoid the passing into the lower level, and the decision will be “nogo”. When EWS is inactive the decision is “go”. The decision mechanism is affected by the threshold of vigilance s_{vig} , which will be detailed later.

C. Learning algorithm

Success and failure maps represent the knowledge in success and in failure inside the state space. First maps will be initialized in the state space V . Then we take one vector \vec{v} randomly from this space. In the phase of evaluation, we calculate the distance between this vector and all the neurons of both maps, see equation 1, where \vec{d}_s^i is the distance between \vec{v} and the i^{th} neuron in *success-map*, \vec{w}_s^i is the weight vector of this neuron, \vec{d}_f^i is the distance between \vec{v} and the i^{th} neuron in *failure-map* and \vec{w}_f^i is the weight vector of this neuron. For each map, the neuron winner corresponds to the smallest distance to the vector. This distance represents therefore the distance between \vec{v} and the map, see equation 2, where d_s is the distance between \vec{v} and *success-map* while d_f is the distance with the *failure-map*. In the phase of decision, we compare d_s with d_f , by taking into account the threshold of vigilance s_{vig} which represents the tolerance to risks. If the threshold is higher than the difference between the distance to failure map and the distance to success map, the early warning signal becomes active, otherwise, this signal is inactive, see equation 3. The activation of EWS indicates that \vec{v} will lead to failure if it is passed into the lower

level. As maps are in the learning phase, it is possible that vector \vec{v} can activate EWS at a time and inactivate it at another time, because the distances with the neurons change. A decision of “nogo” corresponds to active EWS and a decision of “go” corresponds to inactive EWS . In the case where decision is “nogo”, we take another vector \vec{v} randomly from V , then we look for expected results by evaluation and decision phases as detailed before. In case where decision is *go* (\vec{v} may lead to success), the vector will be passed into the low level controller to run a trial. After each trial there is a reward, either negative (failure) or positive (success). Only one map learns \vec{v} . If the reward is negative the failure map will learn, and if it is positive the success map will learn. Next, we take other vectors randomly from V and execute the same steps until the stabilization of maps. The algorithm is as follows:

- 1) $\cup (\text{success-map}, \text{failure-map}) \in V$.
- 2) $\cup \vec{v} \in V$

a) Evaluation :

the distances to the neurons of the two maps:

$$\begin{cases} \vec{d}_s^i = -\vec{w}_s^i + \vec{v} \\ \vec{d}_f^i = -\vec{w}_f^i + \vec{v} \end{cases} \quad (1)$$

the distances to the neurons winners of the two maps:

$$\begin{cases} d_s = \min \| \vec{d}_s^i \| \\ d_f = \min \| \vec{d}_f^i \| \end{cases} \quad (2)$$

b) Decision :

$$EWS = \begin{cases} 0 & (\text{go}) & \text{if } (d_f - d_s) > s_{vig} \\ 1 & (\text{nogo}) & \text{otherwise.} \end{cases} \quad (3)$$

3) if (nogo) go to 2.

else if (go) test \vec{v} , and get a reward R .

if (R : positive) learn *success-map*,

else if (R : negative) learn *failure-map*,

go to 2.

D. Concept of Vigilance

Some psychological research studies suggest that some people are more tolerant to risk than others who are more cautious, [14][15][16]. The vigilance is related to human learning approaches and decision making [12]. In the standard psychological assessment of risk taking, people are classed as risk seeking or risk averse [13]. In our study the vigilance is represented by a threshold s_{vig} that is used to adjust the early warning signal in the decision mechanism. This threshold describes the tolerance of risk, see fig.1. By definition, the threshold of vigilance is the allowed margin of the difference between the distances of state space vector \vec{v} with failure map and with success map, for which the decision mechanism still responds with “go”, see equation 3. The threshold has a limited value according to the dimensions of the state space. In a two dimension state space $s_{vig} \in [-\sqrt{2}, +\sqrt{2}]$. Toward positive values of the threshold, the decision mechanism becomes more alert

to risk (cautious). In the opposite it has a tendency to take risks (courageous), see Fig 2, where D is the diameter of the space. For instance, if $s_{vig} = 0.1$ the early warning system stays inactive for vectors closer to success map than to failure map by 0.1. In such case, if $d_s = 0.3$ and $d_f = 0.35$, then EWS becomes active, while it is inactive if $s_{vig} = 0.04$. The change in the value of s_{vig} from 0.1 to 0.04 allows the agent to be more tolerant to risks.

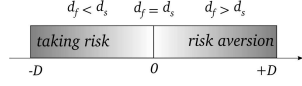


Fig. 2. Threshold of vigilance and tolerance of risk .

In this paper, we have fixed the threshold during learning. But we present the result for different values of the threshold.

III. BIOLOGICAL INSPIRED NEURAL CONTROLLERS FOR WALKING

Biological inspired locomotion controllers are based on the simple circuit that is built from sensor neuron, motor neuron, and inter-neuron [17][18][20][19]. Neurophysiological studies associate the rhythmic movement with the oscillation activity of a type of neurons, called neurons oscillators [21][22]. These oscillators can produce rhythmic activity without sensory input even without central input. But the sensory information is indispensable for walking because it allows to shape the rhythmic patterns in order to interact with the environment [23]. However, sensory informations are mainly used to adapt the controller in front of changes and perturbations. Neurophysiologists have proved that biological controllers like Central Pattern Generators (CPG) have an adaptation mechanism that belongs to plasticity properties [21][24]. To realize the learning approach, we are interested in having the low level controller interact with the environment, like the neural reflexive controller proposed by [17] and tested on a real robot. This low level controller is based on the sensor motor approach. Our learning mechanism will regulate certain parameters in this controller for walking, and to explore the domain of viability to give the ability of walking adaptation to the environment.

A. Neural model for Sensory-Motor

In the neural model for sensory motor there are direct connections between neurons sensors and neurons motors, see fig 3. A static model of sensor neuron has proposed by Ekberg, [19], it is described in the equ.4, where ρ_i is the activity of sensor neuron, α is a positive constant that denotes the dynamics of the neuron, θ is the amplitude and ϕ is the neuron input. ϕ can be an angular position, or a contact force [17]. In the other side, there is a model of neuron motor. Beer [25] has proposed a dynamic model that is described in equ.5 where y_j is the mean membrane potential of the j^{th} motor neuron, τ is a time constant, ρ_i is the activity of the i^{th} sensor neuron, w_{ij} is the synaptic weight between the i^{th} sensor neuron and the j^{th} motor neuron, u_j is the activity of this motor neuron, θ_m is the bias of this neuron.

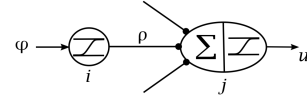


Fig. 3. A neural model of sensory motor controller.

$$\rho_i = (1 + e^{\alpha(\theta - \phi)})^{-1} \quad (4)$$

$$\begin{cases} \tau \cdot \frac{dy_j}{dt} = -y_j + \sum_i w_{ij} \cdot \rho_i \\ u_j = (1 + e^{\alpha(\theta_m - y_j)})^{-1} \end{cases} \quad (5)$$

B. neural reflexive controller

The neural architecture proposed by Wörgötter to control a simulated biped [17] is based on the sensory motor approach where sensor neurons are connected to extension and flexion motor neurons. Fig.4 shows the principles of this controller. A is a stretch receptor sensor neuron, G is a ground contact sensor neuron, FM is a flexion motor neuron, EM is an extension motor neuron. Lines with an arrow extremity indicate excitatory connections, and dotted lines terminated by a solid circle indicate inhibitory connections. Fig.?? shows the interaction between the ground contact sensor neuron of the stance leg and the flexion and extension motor neurons in this leg. Ground contact sensor neuron G in a leg excites the extension motor neuron EM in the knee and the flexion motor neuron FM in the hip of the same leg. Figure ?? shows the interaction between ground contact sensor neuron and the flexion and extension motor neuron in the other leg. It excites the flexion motor neuron in the knee and the extension motor neuron in the hip. Figure ?? shows the role of extension and flexion sensor neurons, E , F , to inhibit the corresponding motor neuron. This is the same for all joints. This behavior is referred as the articular reflex. Figure ?? shows the role of the stretch receptor sensor neuron to excite the extension motor neuron in the knee of the same leg. This behavior is referred as the extension reflex.

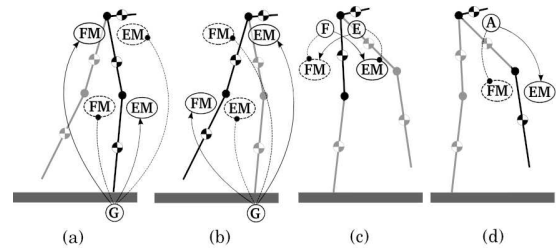


Fig. 4. Principles of the neural reflexive controller proposed by Wörgötter.(a)Interaction with the stance leg. (b)Interaction with the swing leg. (c)Articular reflex. (d)Extension reflex.

In this study, we concentrate on two parameters of this low level controller. The first α_{hip} denotes the dynamics of rhythmic movement in the hip joint (dynamics of extensor sensor neuron), while the second $\theta_{hip-max}$ represents the amplitude of this movement (amplitude in the activity of extensor sensor neuron). By controlling these two parameters,

the biped can walk and face environment changes, such as slope terrain variations.

C. Determination of viability domain of the neural controller

We have explored the domain of viability of the controller by varying the dynamics and the amplitude of the hip extensor sensor neuron (α_{hip} and $\theta_{hip-max}$) on a flat terrain. Inside a defined space for the two parameters, variations have been carried out with defined steps. For each couple ($\alpha_{hip}, \theta_{hip-max}$) the walking has been tested. According to definitions for success and failure we can know which couple leads to success or to failure. The biped has 10 seconds to walk, so if this time was passed and it was still staying, then it is a success. Otherwise, if it falls down before the time, it is a failure. In the simulation, we consider that the robot falls down when the gravity center of the trunk comes below the one of the two shanks. In such case the simulation will stop the trial. For all trials the robot has the same initial position in which one leg is in the stance phase and the other one is in the swing phase, because we are not interested here in the initial phase of walking. Figure 5 shows the results of this analytical studies related to flat terrain walk. The failure trials are represented by spots in surrounding area, while other spots represent the success trials. α_{hip} varies in $[0 : 0.5 : 20]$, while $\theta_{hip-max}$ varies in $[90^\circ : 1 : 150^\circ]$. Walking velocity is limited in our case between $0.33[m/s]$ and $0.66[m/s]$. In the simulation, the walking velocity corresponds to the averaged velocity measured for the trunk.

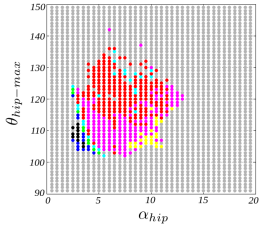


Fig. 5. Domain of viability of the low level controller in space of α_{hip} and $\theta_{hip-max}$.

IV. LEARN WALKING AND ADAPTION APPROACH

First, the biped will learn to walk on a flat terrain. The goal is to allow maps to explore the domain of viability in the state space. We will present the results for several values of vigilance threshold and discuss them. Second, we will present the results for a more complicated architecture devoted to learn how to walk on sloped terrain through an example that explains the adaptation approach for sloped terrain.

A. Learn walking on flat terrain

We present how our learning approach makes success and failure maps to explore the space of parameters in order to find the domain of viability of the controller. The simulation is run for different values of vigilance threshold s_{vig} . Fig.6 shows the control diagram in case of learning

on flat terrain. There are two loops of control, a loop of low level control represented by the interaction between the biped and the controller neuronal sensory motor, the other loop concerns the high level controller where the learning mechanism controls the low level controller and receives the result for each trial (success, failure).

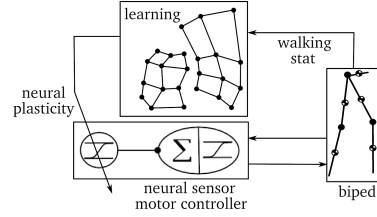


Fig. 6. Walking control diagram, composed of two control levels: neural sensor-motor controller (low level) and learning mechanism (high level).

In the learning algorithm, we initialize success map and failure map in the space of α_{hip} and $\theta_{hip-max}$. The same space has been used as studied previously for the domain of viability. The number of trials and the vigilance threshold are determined. For a random vector $\vec{v}(\alpha_{hip}, \theta_{hip-max})$ from this space there are two processing phases, the evaluation phase and the decision phase. If the early warning signal stays inactive for \vec{v} , then it may lead to success according to the past experience of this system represented by success map and failure map and also according to the risk tendency represented by vigilance threshold s_{vig} . Each vector that will lead to success has been passed to the controller sensory motor to run a trial on the biped. According to the result of each trial one map will learn. Then we look for another vector $\vec{v}(\alpha_{hip}, \theta_{hip-max})$ from the space, and so on. After learning, all the vectors that had lead to success have been incorporated into success map and all the vectors had lead to failure have been incorporated into failure map. The importance of this learning approach is double. First the maps themselves gain experience. Second there is a specific technique of election that is used before each trial and controlled by the threshold of vigilance. Figures 7 and 8 show success map and failure map after learning, with 500 trials, for two different values of vigilance threshold. The state space in our studies is normalized between 0 and 1. Each map is composed of 100 neurons. Neuron weights values (w_1, w_2) denote a configuration of the low level controller ($w_1 = \alpha, w_2 = \theta_{hip-max}$). We have therefore 100 different configurations in each map that match 100 walking gaits stored in success map.

After learning with 500 trials, if $s_{vig} = 0.05$, we obtain 98% of succeeded trials, while 2% of failure. With another threshold $s_{vig} = 0$ we obtain 96% of succeeded trials, and 45% of success with $s_{vig} = -0.1$ and 28% of success with $s_{vig} = -0.2$. In the last case, as there is 72% failure the failure map was learned better than in the other cases.

In figure 7 and 8 all neurons in the success map lead to success (walk), but in the last map the domain of viability presented by the zone occupied by success map is bigger than in the other cases, which allows to have more stability

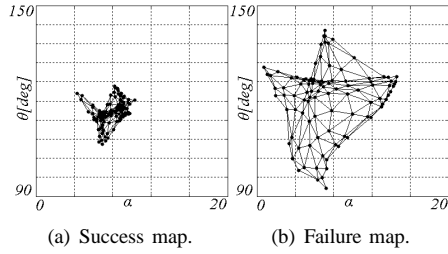


Fig. 7. Success and failure maps after learning on flat terrain with vigilance threshold $s_{vig} = 0.05$.

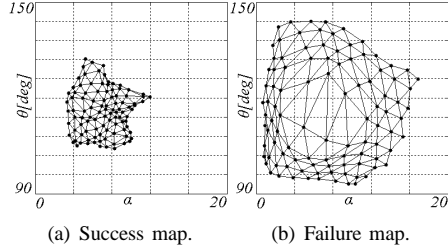


Fig. 8. Success and failure maps after learning on flat terrain with vigilance threshold $s_{vig} = -0.2$.

and more walking gaits. So we can distinguish between two different behaviors for the system, risk taking and risk averse.

Thanks to the two behaviors the system can gain experience in walking, and in case of risky behavior the system learns better. Figure 9 presents the rate of success as a function of vigilance threshold.

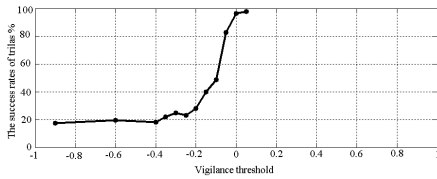


Fig. 9. Rate of succeeded trials as a function of vigilance threshold.

We can divide this figure into 3 zones. The first zone corresponds to $s_{vig} > 0.05$ where there is no decision, no trials, then no learning. The second zone corresponds to $s_{vig} < -0.4$, the system is more risky, and for a more negative threshold the decision will be *go* for all vectors. The middle zone is the most important, because it is a zone of switching between two different behaviors. In our studies we fixed the vigilance threshold during the learning phase, but changing this variable from a trial to another may be worth investigating.

B. Learning on slope terrain

The objective from the previous study is to represent the zone of success in the state space by success map to justify the analytical study of the domain of viability. Our objective now is to generalize the controller on slope terrains. The modification in the maps structure consists of adding a third dimension to describe the terrain slope γ . Now the maps will learn in space of α_{hip} , $\theta_{hip-max}$ and γ . In our study the

slope is limited between $+10^\circ$ and -10° . In the learning phase the biped learns to walk on terrains with different slopes that had been chosen randomly. After learning, the two SOM must be organized in the three dimension state space to represent success and failure experience. Figures 10 and 11 show success and failure maps after learning for different values of vigilance threshold.

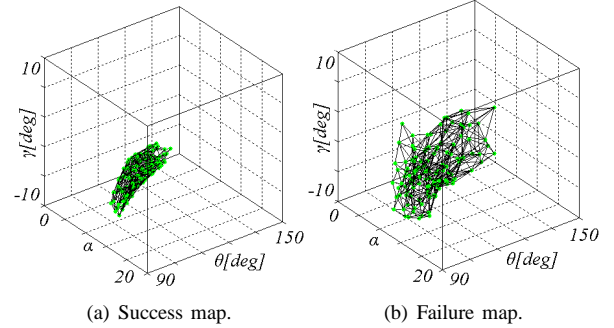


Fig. 10. Success and failure maps after learning on different terrain slopes with vigilance threshold $s_{vig} = 0.0$.

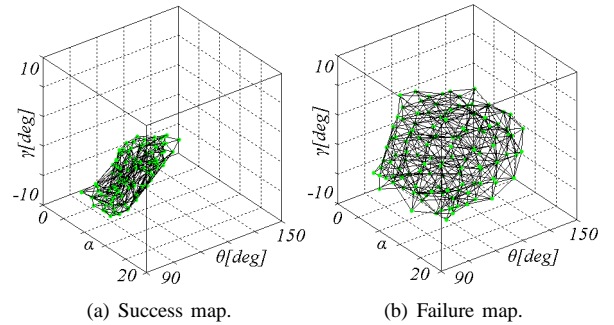


Fig. 11. Success and failure maps after learning on different terrain slopes with vigilance threshold $s_{vig} = -0.2$.

Each map is composed of 125 neurons where each neuron has three weights (w_1, w_2, w_3) that denote a configuration of the low level controller ($w_1 = \alpha_{hip}$, $w_2 = \theta_{hip-max}$) for walking on determined sloped terrain ($w_3 = \gamma$). When $s_{vig} = 0$ there is a success in 86% of trials and a failure in 14%. Success and failure maps are shown in figures 10(a) and 10(b) respectively. For the other value of vigilance, $s_{vig} = -0.2$, there is a success only in 15% of trials and a failure in 85%, as shown in figures 11(a) and 11(b). The space occupied by success map in the second case is bigger than in the first case. This difference is referred to as the difference in the behavior according to vigilance threshold. As the failure rate in the second case is higher than in the first case, the failure map will learn better in the second case.

After learning, each neuron in the success map corresponds to a walking on a particular slope, including gait and speed. To walk on a terrain with a particular slope γ , a calculation occurs between all neurons to find the winner without taking the (w_1, w_2) values for neurons into account. The winner is the neuron whose w_3 is the closest to γ , while other weights of the neuron winner are used to configure the parameters (α_{hip} , $\theta_{hip-max}$) of the low level controller.

Changing the terrain slope during walking causes switching into another neuron in the map, which corresponds to the new slope. This switch can be direct between the neurons or indirect by use of intermediary neurons. Figure 12 shows how the biped can walk on a sloped terrains.

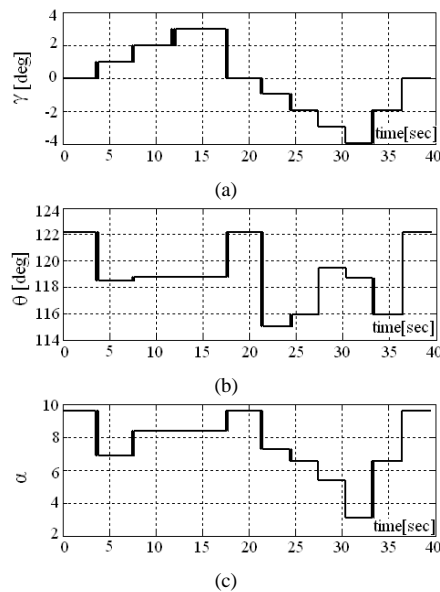


Fig. 12. Switching between the neurons of success map during walking on irregular terrain. (a) represent the terrain slope, which is an input to the learning mechanism. (b) and (c) are the amplitude and the dynamics of the extensor sensor neuron, the outputs of the learning mechanism.

For any slope γ in the domain of viability (success map) there is a corresponding couple $(\alpha_{hip}, \theta_{hip-max})$ that can be applied to the lower level of control to perform the walking.

V. CONCLUSION

In this paper we presented a neurobiological inspired learning algorithm. The objectives of the mechanism were to learn from mistakes and to avoid making them again. This was done by building on experience of past mistakes and successes. We showed how these two experiences could build themselves through the stages of evaluation, decision and then trials. It can be said that the negative reward has an importance as the positive. This mechanism was implemented on a planar biped and allowed the biped to learn to walk without supervision. It added the property of adaptation even to changes of terrain slope. Our future work shall address adaptation to further changes in the environment, as well as changes in the physical parameters of the biped, as an important factor to allow our mechanism to apply to a real robot. An analytical study that will investigate how much benefit the system is able to get from previous experience will also be considered.

REFERENCES

[1] T. Singer, B. Seymour, J. O'Doherty, H. Kaube, R.J. Dolan and C.D. Frith, Empathy for pain involves the affective but not sensory components of pain, *Science*, vol. 303, no. 5661, 2004, pp. 1157-1162.

[2] J.W. Brown and T.S. Braver, A computational model of risk, conflict, and individual difference effects in the anterior cingulate cortex, *Brain Research*, vol. 1202, 2008, pp.99-108.

[3] RB. Mars, MG. Coles, MJ.Grol, CB.Holroyd, S.Nieuwenhuis, W.Hulstijn and I.Toni, Neural dynamics of error processing in medial frontal cortex, *NeuroImage*, vol. 28, no. 4, 2005, pp. 1007-1013.

[4] H. Gemba, K. Sasaki and V.B. Brooks, Error potentials in limbic cortex (anterior cingulate area 24) of monkeys during motor learning. *Neuroscience letters*, 1986, vol. 70, no. 2, pp. 223-227.

[5] W.J. Gehring, M.G.H. Coles, D.E. Meyer and E. Donchin, The error-related negativity: An event-related potential accompanying errors, *Psychophysiology*, 1990, vol. 27, S34.

[6] J. Hohsbein, M. Falkenstein and J. Hoonman, Error processing in visual and auditory choice reaction tasks, *Journal of Psychophysiology*, 1989, vol. 3, 32.

[7] J.W. Brown and T.S. Braver, Learned Predictions of Error Likelihood in the Anterior Cingulate Cortex *Science*, 2005, vol. 307, no. 5712, pp. 1118-1121.

[8] L. Van Leijenhorst, P.M. Westenberg and E.A. Crone, A developmental study of risky decisions on the Cake Gambling Task: Age and gender analyses of probability estimation and reward evaluation, *Developmental Neuropsychology*, 2008, vol. 33, no. 2, pp. 179-196.

[9] K. Matsumoto, W. Suzuki and K. Tanaka, Neuronal correlates of goal-based motor selection in the prefrontal cortex, *Science*, 2003, vol. 301, no. 5630, pp. 229-232.

[10] T. Kohonen, Self-Organizing Maps, *Springer Series in Information Sciences*, vol. 30, Springer, Berlin, Heidelberg, New York, 1995, 1997, 2001. Third Extended Edition, 501 pages. ISBN 3-540-67921-9, ISSN 0720-678X

[11] D.E. Rumelhart and D. Zipser, Feature discovery by competitive learning, *Cognitive science*, 1985, vol. 9, no. 1, pp. 75-112.

[12] H. Ahn, and R. Picard, Affective-Cognitive Learning and Decision Making: A Motivational Reward Framework For Affective Agent, *The 1st International Conference on Affective Computing and Intelligent Interaction*. October 22-24, 2005, Beijing, China.

[13] D.J. Kruger, X.T. Wang, and A. Wilke, Towards the development of an evolutionarily valid domain-specific risk-taking scale, *Evolutionary Psychology*, 2007, vol. 5, no. 3, pp. 555-568.

[14] J.-L. van Gelder, R.E. de Vries and J. van der Plicht, Evaluating a dual-process model of risk: affect and cognition as determinants of risky choice, *Journal of Behavioral Decision Making*, 2008, vol. 22, no. 1, pp. 45-61.

[15] B. Pawlowski, R. Atwal and R.I.M. Dunbar, Sex differences in everyday risk-taking behavior in humans, *Evolutionary Psychology*, 2008, vol.6, no.1, pp. 29-42.

[16] P. Horvath and M. Zuckerman, Sensation seeking, risk appraisal, and risky behavior. *Personality and individual differences*, 1993, vol. 14, no. 1, pp. 41-52

[17] T. Geng, B. Porr and F. Wörgötter, Fast Biped Walking with a Sensor-driven Neuronal Controller and Real-time Online Learning, *The International Journal of Robotics Research*, 2006, vol. 25, no. 3, pp. 243-259.

[18] G. Taga, Y. Yamaguchi and H. Shimizu, Self-organized control of bipedal locomotion by neural oscillators in unpredictable environment, *Biological Cybernetics*, 1991, vol. 65, no.3, pp. 147-159.

[19] T. Wadden and O. Ekeberg, A neuro-mechanical model of legged locomotion: single leg control, *Biological Cybernetics*, 1998, vol. 79, no.2, pp. 161-173.

[20] H. Cruse, Ch. Bartling, M. Dreifert, J. Schmitz, D.E. Brunn, J. Dean and T. Kindermann, Walking: a complex behavior controlled by simple systems, *Adaptive Behavior*, 1995, vol. 3, no. 4, pp. 385-418.

[21] D.A. McCrea and I.A. Rybak, Organization of mammalian locomotor rhythm and pattern generation, *Brain Research Reviews*, 2008, vol. 57, no. 1, pp. 134-146.

[22] P.F. Rowat and A.I. Selverston, Learning algorithms for oscillatory networks with gap junctions and membrane currents, *Network*, 1991, vol. 2, no. 1, pp. 17-41.

[23] E. Marder and R.L. Calabrese, Principles of rhythmic motor pattern generation, *Physiological Reviews*, 1996, vol. 76, no. 3, pp. 687-717.

[24] A. Ishiguro, A. Fujii and P.E. Hotz, Neuromodulated Control of Bipedal Locomotion Using a Polymorphic CPG Circuit, *Adaptive Behavior*, 2003, vol. 11, no. 1, pp. 7-17.

[25] 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*, 1992, vol. 4, no. 3, pp. 356-365.

- 9.9 Adaptive Locomotive Behaviors of a Biped Robot : Patterns Generation and Classification. Proceeding of International Conference on Simulation of Adaptive behavior, SAB 2010**

A study of Adaptive Locomotive Behaviors of a Biped Robot: Patterns Generation and Classification

J. Nassour^{1,3}, P. Hénaff², F. B. Ouezdou¹, and G. Cheng³

¹ Versailles Saint Quentin University - France

² University of Cergy Pontoise, ENSEA, CNRS-F95000 Cergy Pontoise

³ Institute for Cognitive Systems, Technical University Munich

{*ouezdou, nassour*}@*lisv.uvsq.fr*, *patrick.henaff@u-cergy.fr*, *gordon@tum.de*

Abstract. Neurobiological studies showed the important role of Central Pattern Generators for spinal cord in the control and sensory feedback of animals' locomotion. In this paper, this role is taken into account in modeling bipedal locomotion of a robot. Indeed, as a rhythm generator, a non-classical model of a neuron that can generate oscillatory as well as diverse motor patterns is presented. This allows different motion patterns on the joints to be generated easily. Complex tasks, like walking, running, and obstacle avoidance require more than just oscillatory movements. Our model provides the ability to switch between intrinsic behaviors, to enable the robot to react against environmental changes quickly. To achieve complex tasks while handling external perturbations, a new space for joints' patterns is introduced. Patterns are generated by our learning mechanism based on success and failure with the concept of vigilance. This allows the robot to be prudent at the beginning and adventurous at the end of the learning process, inducing a more efficient exploration for new patterns. Motion patterns of the joint are classified into classes according to a metric, which reflects the kinetic energy of the limb. Due to the classification metric, high-level control for action learning is introduced. For instance, an adaptive behavior of the rhythm generator neurons in the hip and the knee joints against external perturbation are shown to demonstrate the effectiveness of the proposed learning approach.

1 Introduction

Biological studies of animals suggest that animals' locomotion is mainly generated at the spinal cord, by a combination of a central pattern generator (CPG) and reflexes receiving adjustment signals from a cerebrum, cerebellum and the brain [1], [2], [3]. These studies were taken into account in robot's locomotion gait in order to implement such mechanism, especially on legged robots [4], [5], [6], [7], [8]. Biologically inspired walking mechanism for legged robot does not require a perfect knowledge of the robot's dynamics. Different models of neural oscillators are widely used to generate rhythmic motion [9], [10], [11], [12], [13].

Such oscillations generated by two mutually inhibiting neurons are described in a set of differential equations (e.g. Matsuoka [9]). Rowat and Selverston [14] proposed a new model of rhythmic neuron that can generate different types of patterns such as oscillatory ones. The different behaviors in the activity of these neurons can be used in robot's locomotion to achieve different tasks as well as walking. Complex task, like walking, hopping, running, and obstacle avoidance, require correct synchronization and switching between patterns [15]. In action learning approach, where learning always occurs in the space of parameters, there is a limitation to learn complex tasks, due to the dimension of this space which can drastically increases. This issue can be solved by looking for a new representation of patterns. Instead of learning in the space of parameters, learning can occur inside a new space called patterns' space. (e.g. in case of one dimensional patterns space, patterns will be represented only on one axis). Our work aims to produce a biological inspired neural controller for biped walking, based on CPG with a rhythmic neuron proposed by Rowat and Selverston [14]. According to the environment changes, the adaptation of the neurons behavior will be shown. Therefore, a new space for patterns allowing intrinsic behaviors of a joint motion will be proposed.

This paper is organized as following. Section 2 presents the principles of the neural controller based on the model of rhythmic neurons, which is able to generate CPG-like patterns. The three layers of the CPG used in bipedal control will be presented. A coupling circuitry for walking will be proposed. Next, the walk learning phase based on previous experience with a threshold of vigilance to allow extensive patterns search within a large space of parameters will be detailed in section 3. In the fourth section, a new representation of successful and failure walking patterns is proposed. This approach allows a high level control in space of patterns instead of space of parameters. The effectiveness of our learning scheme, which allows switching between bipedal patterns to achieve different locomotive tasks will be demonstrated. Moreover, an example on the adaptation behavior of the rhythm generator neurons in the hip and the knee joints against external perturbation will be shown. The last section gives a conclusion and details of further developments.

2 Neural Control of Locomotion

Physiological studies suggest that rhythmic movements in animal's locomotion system are produced by a neural network called CPG [16]. It can generate a locomotive rhythmic behaviors with neither sensory nor central inputs [17]. Sensory inputs shape the output of this locomotion system, and allow the animal to adapt its locomotion patterns to external or internal changes. Genetic studies on newborn rat and mice suggest that rhythmic limb movements during locomotion are generated by neuronal networks located within the spinal cord [18]. Matsuoka and McMillen neural oscillators are widely used as mathematical models for non-linear oscillators [9], [10]. These half-centre oscillators consist of two neurons that individually have no rhythmic behavior, but which produce rhythmic

mic outputs when they are reciprocally coupled. This paper present another model of non-linear rhythm generator. This model is based on the fact that one neuron can generate oscillatory as well as different motor patterns [14].

2.1 Cell Model

The cell model introduced by Rowat and Selverston to modulate the gastric mill CPG in the lobster is interesting due to its ability to generate different patterns by controlling only two parameters [14]. Furthermore, such patterns can be generated with only one neuron without need for another coupled neuron as used in classical models [9], [10]. In the adopted model, the membrane currents of the neuron are separated into two classes, fast and slow, according to their time responses. The sum of all fast currents is modeled by a single fast one, and a single slow current is used to model the sum of all slow ones. This model cell has two differential equations, one for membrane potential V , derived from current's conservation, and one for lumped slow current q , derived from current's activation, see eq.(1).

$$\tau_m \cdot \frac{dV}{dt} = -(fast(V, \sigma_f) + q - i_{inj}) \quad \tau_s \cdot \frac{dq}{dt} = -q + q_\infty(V) \quad (1)$$

While the fast current is supposed to activate immediately, the membrane time constant τ_m is assumed to be smaller than the slow current's time constant for activation τ_s . We have taken the ratio of τ_s to τ_m to be about 20 as in [14], $\tau_m = 0.05$, and $\tau_s = 1$ for all rhythmic neurons. The injected current is i_{inj} . An idealized current-voltage curve for the lumped fast current is given by: $fast(V, \sigma_f) = V - A_f \cdot tanh((\sigma_f/A_f)V)$. The fast current can represent the sum of a leak current and an inward Ca^{++} . The dimensionless shape parameter for current-voltage curve is given by: $\sigma_f = \frac{g_{Ca}}{g_L}$. Where g_L is a leak conductance and g_{Ca} is the calcium conductance. $q_\infty(V)$ is the steady state value of the lumped slow current, which is given by: $q_\infty(V) = \sigma_s(V - E_s)$. $q_\infty(V)$ is linear in V with a reversal potential E_s . σ_s is the potassium conductance g_K normalized to g_L . σ_s is given by: $\sigma_s = \frac{g_K}{g_L}$. q and i_{inj} have the dimension of an electrical potential. A true current is obtained by multiplying the model current by a leak conductance g_L . V , E_s , i_{inj} , and q are given in millivolts while τ_s and τ_f are expressed in milliseconds. With different values of the cell parameters, different intrinsic behaviors can be achieved : quiescence (Q), almost an oscillator (A), endogenous oscillator (O), depolarization (D), hyperpolarization (H), and plateau (P), as shown in Fig.1. In bio-inspired locomotion, a pair of neurons

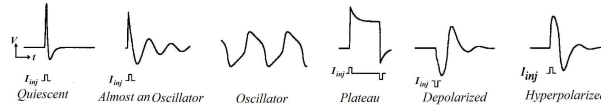


Fig. 1. The six intrinsic behaviors of the cell's model, Rowat and Selverston [14].

with mutual inhibition can be used to generate rhythmic motion in extension and flexion. A bio-inspired model for locomotion is proposed in the next section.

2.2 Locomotion Model

Studies of rhythmic movement in the animal show that local circuits in the spinal cord are able to control the timing and coordination of complex motion patterns [19]. The locomotion and rhythmic movements in mammals are organized by oscillatory spinal cord circuits called CPGs. Experimental studies show that the rhythmic patterns in cat limbs can be generated in the absence of descending control from higher centers and sensory feedback [3]. Each joint appears to have its own CPG, which can be coupled to the CPG of another joint in order to achieve complex movements such as walking, running, swimming, flying, etc. These CPGs controlling such behaviors in animals locomotion can be responsible of rhythmic movements in human locomotion [20]. Several schemes for the spinal CPG have been proposed to generate rhythmic movements: "half-center CPG" proposed by Brown [21], "half-center CPG" with more complex patterns of motoneuron activity introduced by Perret et al. [22] and "half-center CPG" with sensory input proposed by Orlovsky et al. [1]. One drawback of these models is the direct excitatory connection between the rhythm generator interneurons and motoneurons. Any change in the interneurons layer will affect simultaneously the motoneurons layer. A more complicated architecture is required to face the adaptation with the environment changes. Two and three levels CPGs with rhythm generation and pattern formation circuitry have been proposed by [2] and [23]. This model separates cycle timing and motoneurons activation. In order to achieve a rhythmic movement such as walking, the CPG model was implemented on a simulated biped robot using MATLAB software. Fig.2(a) shows the wiring diagram for one biped robot's joint. It can be separated into three layers: Rhythm Generation neurons (RG), Pattern Formation neurons (PF) and MotorNeurons (MN). Sensory feedback shapes the activity of these neurons. This paper focuses on the effect of descending control on the rhythm generators neurons in order to control the behavior of these neurons when external perturbation occurs during walking. In the analytical study, after observing the phase diagram of a joint and changing σ_s and σ_f in the rhythm generators neurons, different motion behaviors were observed on the joint. Fig.2(b) shows the distribution of motion patterns in space of σ_s and σ_f . Varying σ_s and σ_f in RG of a joint will change its motion pattern. The four detected basic motion patterns can lead the robot to achieve some complex tasks like walking, running, and jumping depending on synaptic circuits between joint CPGs.

2.3 Control architecture for a biped robot

Previously, the basic motion patterns obtained for one joint was shown. To achieve a complex movement like walking, synchronization between joints is needed. The complex patterns like walking and running are always composed of synchronized basic patterns. The synchronization between patterns is ensured

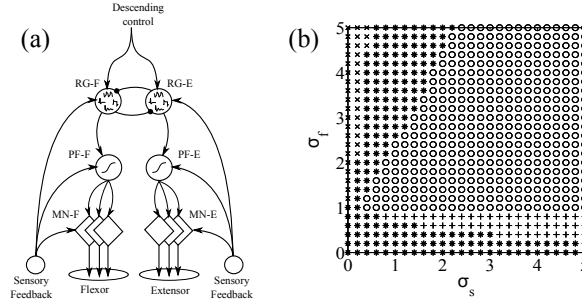


Fig. 2. Model of one joint controller and its motion patterns. (a)The model's scheme, CPG with three levels: Rhythm Generator, Pattern Formation, and MotorNeuron level. (b)The different behaviours observed on the joint for the same injected current. (x): Plateau , (*): Quiescence; (+): Almost an oscillator, and (o): Oscillatory behavior.

by coupling the CPGs for the joints. Fig.3 shows the proposed coupling circuits between the rhythm generator neurons for the hip, the knee, and the ankle joints of a simulated biped robot. Each joint is driven by a simulated servo motor. With such simple coupling, the robot can carry out walking task from basic oscillatory patterns. With different coupling circuits, another task can be achieved. In some complex circuits, the robot can walk with different gaits. A desired task can be accomplished by defining basic patterns and special coupling circuit. The principle of our proposed circuit for walking (see fig.3) is described by the activity between the CPGs which is regulated by excitatory synaptic connections. For inter-limb circuitry, rhythm generator neuron extensor in the left hip (RG-E-hipL) excites rhythm generator neuron flexor in the right hip (RG-F-hipR). Rhythm generator neuron flexor in the left hip (RG-F-hipL) excites rhythm generator neuron extensor in the right hip (RG-E-hipR). The same synaptic excitation is proposed from the right hip to the left hip. For one leg, rhythm generator extensor neuron in the hip (RG-E-hip) excites rhythm generator extensor neuron in the knee (RG-E-knee) and rhythm generator extensor neuron in the ankle (RG-E-ankle) of the same leg. Rhythm generator flexor neuron in the hip joint (RG-F-hip) excites rhythm generator flexor neuron in the knee one (RG-F-knee) and rhythm generator flexor neuron in the ankle joint (RG-F-ankle) of the same leg. As described before, the locomotion is the interaction between CPG, sensory feedback, and descending control. Sensory information is used to shape the motion and manage some perturbations and balance control [24]. Thanks to the interaction with sensory feedback, the robot can walk without a perfect knowledge of its dynamics. A static model of sensory neuron proposed by Ekberg [25] is described in eq.(2). ρ_i is the activity of sensory neuron, α is a positive constant that denotes the dynamics of the neuron, θ is the amplitude and ϕ is the input on the neuron. ϕ can be an angular position, or a contact force [26].

$$\rho_i = (1 + e^{\alpha(\theta - \phi)})^{-1} \quad (2)$$

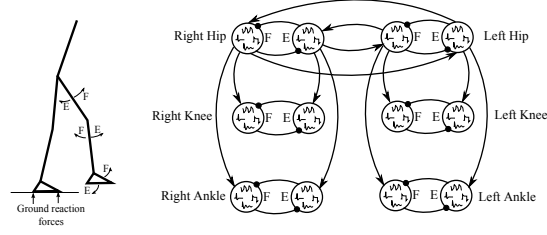


Fig. 3. Planar Biped model and proposed coupling circuitry between rhythm generator neurons for all joints. E and F are extension and flexion.

The extension and flexion sensory neurons in each joint inhibit the corresponding motorneuron for this joint. This circuitry is referred as articular reflex. Equilibrium control is achieved by the difference between the center of pressure and the projection of the center of mass. In our model, the parameter of equilibrium used as input of two neurons: falling forward and falling backward neurons. The activity of both neurons is injected in pattern formation layer at the ankle CPG. If the robot may fall forward, the corresponding neuron becomes active to excite the pattern formation neuron extensor for the ankle of stance leg. The flexor pattern formation neuron will be excited if the falling backward neuron becomes active. Once the control architecture was proposed and the model of rhythmic neurons is determined, it is time to show how the simulated biped is learning to walk on a flat terrain. As the desired task is the walking and the coupling circuit is already defined, the biped will learn basic patterns, in space of σ_s and σ_f , that lead to successful walking.

3 Success and Failure Learning

The objectives of the learning mechanism is to detect in the space of σ_s and σ_f the basic patterns which lead to successful walk. Our previous work in experience-based learning mechanism with the vigilance concept has been used here to detect successful and failure walking patterns, see [27] for more details. Walking trial occurs inside a time window of ten seconds. Successful walking is defined when the simulated biped did not fall during the time window and achieved two steps at least.

This mechanism is composed of two phases: evaluation and decision, see Fig.4. In the evaluation phase, two independent neural networks based on well-known Self Organizing Maps, proposed by Kohonen [28], are used to represent the knowledge in success and in failure. Success map learns in case of success trials, and failure map learns in case of failure ones. During learning, the two maps will be self-organized in the space of parameters that will be therefore divided into three zones: a zone of success represented by success map, a zone of failure represented by failure map, and a zone of conflict that corresponds to the interference between the two maps. The evaluation of any vector \vec{v} from space

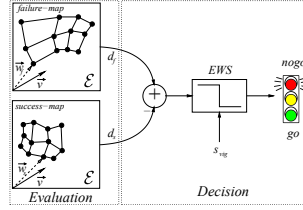


Fig. 4. Learning mechanism with evaluation and decision phases.

\mathcal{E} belonging to success or failure is defined by the distance between \vec{v} and each map. The distance of a vector with a map is the distance between this vector and the closest neuron in the state space (the winner neuron). For each \vec{v} , two distances therefore exist: one to success map called d_s , and another to failure map called d_f . In the decision phase, the comparison between the distance with success map d_s and the one with failure map d_f leads to an expected result in the case where the vector \vec{v} is applied on the controller (trial). According to expected result, if it may lead to failure, then an Early Warning Signal (*EWS*) becomes active to avoid the trial, and the decision will be “nogo”. When *EWS* is inactive, the decision called “go” is taken. The decision mechanism is affected by the threshold of vigilance s_{vig} , which represents the tolerance to risk. The vigilance is related to human learning approaches and decision making [29].

In order to increase the reflectivity of the vigilance threshold model proposed in our previous work [27], a modulation of the above mentioned threshold s_{vig} is introduced. This lead to get different values of it for each trial. Hence, this model increases the learning mechanism efficiency by extending the learning process to sectors of space of parameters. As an important issue, the risk behavior will change from prudence at the beginning of learning to adventure at the end. An example of vigilance threshold modulation is given as following (see Fig.5(c)):

$$y_1 \leq s_{vig} \leq y_2 \quad \begin{cases} y_1 = a_1 - b_1 * \log((x + c_1)^2) \\ y_2 = a_2 - b_2 * \log((x + c_2)^2) \end{cases} \quad (3)$$

The coefficients values are ($a_1 = 0.9, a_2 = 1.47, b_1 = b_2 = 0.15, c_1 = c_2 = 20$) and were chosen after several attempts. y_1 and y_2 chosen curves ensure smooth change between the prudence and adventure above mentioned behaviors. Walking patterns are presented by success map and falling patterns are presented by failure one. With such learning mechanism, learning failure map is as important as learning success map, since falling patterns stored in failure map can be used in an adaptation approach where walking patterns are limited (ex: in case of external perturbation). Fig.5 shows success and failure maps after learning 200 trials based on the new model of the vigilance threshold. The state space is normalized between 0 and 1 and each map has 25 neurons. Weights of neuron (w_1, w_2) denote the parameters of the rhythmic neuron ($w_1 = \sigma_s, w_2 = \sigma_f$). Therefore, there are 25 different configurations in each map that match 25 successful walking gaits stored in success map, and 25 unsuccessful walking patterns

stored in failure map. Because of the topological properties of the Self Organizing Maps, three neurons in failure map are situated in the success zone and show oscillatory behaviors $((0.39, 0.57), (0.46, 0.33), (0.17, 0.23))$, see Fig.5(a). As these neurons did not represent any failure pattern, they are eliminated from the failure map.

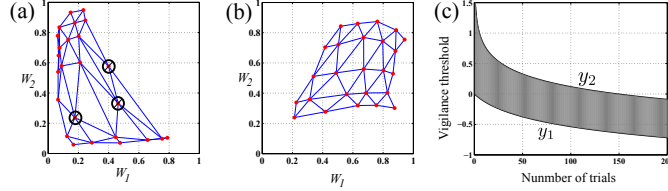


Fig. 5. Success and failure maps after learning walk on flat terrain. (a) Failure map after learning unsuccessful walking patterns. Three neurons was eliminated from the map, because they did not represent any input vector. (b) Success map after learning walking patterns. (c) New vigilance Model related to learning iterations, $y_1 \leq s_{vig} \leq y_2$. The risk behavior will change from prudence at the beginning of learning to adventure at the end.

4 Adaptive behavior for perturbation

As shown in the previous section, the walking task was achieved in the success map zone for the proposed coupling circuits. Because of the synaptic connection between rhythmic generator neurons for all joints, patterns cannot be independent. Then, the same pattern in all joints exist whenever the coupling circuitry is active. To have different patterns on different joints at a time, the synaptic connection between the CPGs must be inhibited. By having independent patterns in the hip, the knee and the ankle joints, the biped can achieve some complex behaviors. In this section, how the robot reacts to an external perturbation force is detailed.

As switching between success map neurons during walking will change the walking pattern and thus walking gait, it can also be interesting to switch between these neurons against external perturbation. The limitation of this algorithm will appear for a large perturbation force. This can be solved by switching toward failure patterns stored in failure map neurons. Inhibit the synaptic connection between CPGs is necessary to get different patterns in different joints.

The space of parameters in such case will be augmented, with a pair (σ_f, σ_s) for each joint. It increases from 2 dimensions in case of existing of coupling circuitry to 12 dimensions in case of independent patterns. To reduce dimensionality, we propose to represent all the patterns of a joint in one axis only. This will reduce the dimension by two and facilitates classification and visualization of high-dimensional data. To do so, a metric \mathcal{E} which reflects the kinetic

energy of one limb is introduced (eq. 4). Based on this metric, an energy based classification of the patterns can be carried out.

$$\mathcal{E} = \int_{t_0}^{t_f} \dot{\theta}^2 dt \quad (4)$$

Fig.6(a) shows the logarithmic scale of the energy based metric for all the motion patterns of Fig.2. Fig.6(b) shows the logarithmic scale of the energy based metric of all neurons of failure and success maps given in Fig.5. First 25 neurons belong to failure map, and last 25 neurons belong to success map. The different behaviors are separated according to the energy based metric of motion patterns. Two neurons with Plateau have the lower values for the energy based metric, then 16 neurons with Quiescent behaviors, then four neurons with Almost an oscillator, then all the neurons of success map according to the Oscillation frequency. Patterns can be classified on a new axis according to the logarithmic scale of the

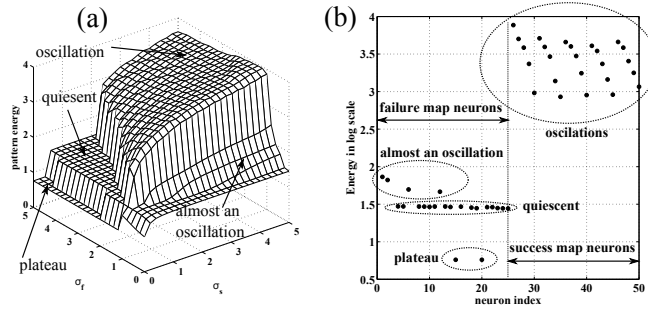


Fig. 6. The energy based metric patterns for the space of σ_s and σ_f (a), and for success and failure neurons represented on the horizontal axis (b). Neurons of success map represent oscillatory patterns with different frequency. Each neuron represent a pattern, but neurons are separated into four classes of patterns according to the energy based metric.

energy based metric. As shown in Fig.6(b) patterns can be positioned on this axis in the following order: Plateau, Quiescent, Almost an oscillator, and Oscillatory patterns from low to high oscillation frequency. All neurons in success and failure maps can be placed on the new axis according to their rhythm. Therefore, two dimensional space (σ_s, σ_f) can be represented in only one dimension axis. One axis is obviously needed for each joint. In the first step of the study, only synapses between CPGs of the hip and the knee joints are inhibited. While the connection between CPGs of the ankle and the hip joints are kept. Fig.7 shows two dimensional space of patterns for the hip and the knee joints. Walking zone in Fig.7(b) corresponds to oscillatory patterns in the hip and the knee joints. In case of external perturbation force, pattern manipulation is necessary to avoid falling. The figure shows the group of patterns in the hip and the knee joints by

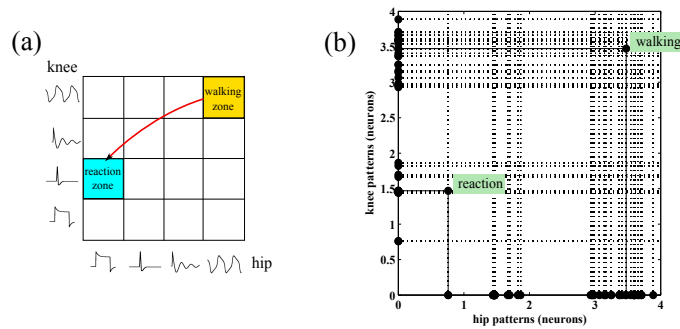


Fig. 7. The space of patterns is for hip and knee joints, with an example of switching against perturbation. (a) Patterns switch from walking by oscillatory patterns to quiescent pattern for knee and plateau for hip. (b) Neurons switch from walking zone to other neurons that represent quiescent pattern for knee and plateau for hip. Each neuron represents one pattern.

which the robot can react against the perturbation. An example for walking and reaction phases is shown in Fig.8. First, it presents the normal walking on a flat terrain without any perturbation. Next, it illustrates the fall because of external perturbation force of $45N$ applied on the back of the robot (the simulated robot mass is about 22 kg and the walking speed is almost 0.2 m/s). Fig.8(c) shows how the biped robot react correctly against the external force by adapting the behavior of the rhythm generators neurons.

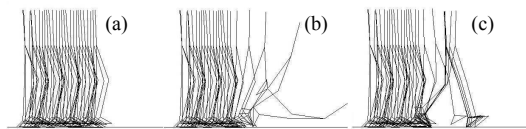


Fig. 8. Effects of adaptation mechanism on the biped to avoid falling. (a) Walking without perturbation. (b) Falling due to the perturbation. (c) Successful walking with adaptation to the perturbation.

5 Conclusion

In this paper a neurobiological inspired controller for biped walking is presented. We showed how the behavior in rhythm generator neurons brings adaptation to face external perturbations. The switching between patterns was simplified by using a simple method to classify success and failure. Moreover, a technique for dimensionality reduction depending on the energy based metric patterns leads

greater benefits, since the classification can be carried out over one axis only in relation to the motion patterns. Hence, establishing a space of patterns for the hip, and the knee joints. This space allows high-level control for goal directed action, thus, learning to achieve more complicated reactions. It also permits other rhythmic movement, where learning patterns replaces learning parameters. This was done by our experience based learning mechanism with this new model for vigilance threshold; we are able to explore in more efficient manner the space of parameters for new motion patterns. This mechanism was implemented on a simulated planar biped and allowed the robot to learn to walk and to react to perturbation without supervision. Our future work shall address goal directed action learning and adaptation to further changes in the environment, as well as changes in the physical parameters of the biped. This important issue will be addressed in order to apply the proposed adaptation mechanism to a humanoid prototype under development.

References

1. Orlovsky, G.N., Deliagina, T., Grillner, S.: Neuronal control of locomotion: from mollusc to man, Anonymous. Oxford University Press, New York(1999)
2. McCrea,D.A., Rybak,I.A.: Organization of mammalian locomotor rhythm and pattern generation. *Brain Research Reviews*. 57(1), 134-146 (2008)
3. Graham-Brown,T.: The intrinsic factors in the act of progression in the mammal. *Proc R Soc Lond B Biol Sci*. 84, 308-319 (1911).
4. Ijspeert,A.J.: Central pattern generators for locomotion control in animals and robots: a review. *Neural Networks*, 21(4), 642-653 (2008).
5. Taga, G., Yamaguchi, Y., Shimizu, H.: Self-organized control of bipedal locomotion by neural oscillators. *Biological Cybernetic*. 65, 147159 (1991).
6. Kimura, H., Akiyama, S., Sakurama, K.: Realization of dynamic walking and running of the quadruped using neural oscillator. *Auton. Robots*. 7, 247258 (1999).
7. Endo, J., Morimoto, J., Matsubara, T., Nakanishi, J., Cheng, G.: Learning CPG-based Biped Locomotion with a Policy Gradient Method: Application to a Humanoid Robot. *The International Journal of Robotics Research, Special Issue on Machine Learning in Robotics*. 27(2), 213-228 (2008).
8. Morimoto, J., Endo, J., Nakanishi, J., Cheng, G.: A Biologically Inspired Biped Locomotion Strategy for Humanoid Robots: Modulation of Sinusoidal Patterns by a Coupled Oscillator Model. *IEEE Transaction on Robotics*. 24(1), 185-191 (2008).
9. Matsuoka, K.: Sustained oscillations generated by mutually inhibiting neurons with adaptation. *Biological cybernetics*. 52(6), 367-376 (1985).
10. McMillen D. R., Gabriele M. T. DEleuterio . Simple Central Pattern Generator model using phasic analog neurons. *Physical review*. 59(6), 6994-6999 (1999).
11. Righetti, L., Buchli, J., Ijspeert, A. J.: Adaptive Frequency Oscillators and Applications. *The Open Cybernetics and Systemics Journal*. 3, 64-69 (2009).
12. Righetti, L., Buchli, J., Ijspeert, A. J.: Dynamic hebbian learning in adaptive frequency oscillators. *Physica D*. 216(2), 269-281 (2006).
13. Nakanishi, J., Morimoto, J., Endo, G., Cheng, G., Schaal, S., Kawato, M.: Learning from demonstration and adaptation of biped locomotion, *Robotics and Autonomous Systems Journal*. 47(2-3), 79-91 (2004).
14. Rowat, P.F., Selverston, A.I.: Learning algorithms for oscillatory networks with gap junctions and membrane currents. *Network*. 2(1), 17-41 (1991).
15. Ivanenko, Y.P., Cappellini, G, Dominici, N., oppele ,R. E., Lacquaniti, F.: Modular Control of Limb Movements during Human Locomotion. *The Journal of Neuroscience*. 27(41), 11149-11161 (2007).
16. Marder , E., Calabrese, R.L.: Principles of rhythmic motor pattern generation. *Physiol Rev*. 76, 687717 (1996).
17. Kuo, A.D.: The relative roles of feedforward and feedback in the control of rhythmic movements. *Motor Control*. 6, 129-145 (2002).
18. Kiehn, O., Butt, S.J.: Physiological, anatomical and genetic identification of CPG neurons in the developing mammalian spinal cord. *Prog Neurobiol* 70, 347-361 (2003).
19. Williams, S. M., Purves, D.: *Neuroscience* (2nd edition). Publisher: Sinauer Associates. January 15, (2001).

20. Taga, G.: Nonlinear Dynamics of Human Locomotion: from Real-Time Adaptation to Development. Book Chapter: Adaptive Motion of Animals and Machines. Springer Tokyo. 189-204 (2006).
21. Graham-Brown, T.G.: On the fundamental activity of the nervous centres: together with an analysis of the conditioning of rhythmic activity in progression, and a theory of the evolution of function in the nervous system. *J. Physiol.* 48, 1841 (1914).
22. Perret, C., Cabelguen, J.M., Orsal, D.: Analysis of the pattern of activity in knee flexor motoneurons during locomotion in the cat. In: Gurfinkle, V.S., Ioffe, M.E., Massion, J., Roll, J.P.(Eds.), *Stance and Motion: Facts and Concepts*. Plenum Press, New York. 133141 (1988).
23. Koshland, G.F., Smith, J.L.: Mutable and immutable features of paw-shake responses after hindlimb deafferentation in the cat. *J. Neurophysiol.* 62, 162173 (1989).
24. Taga, G.: A model of the neuro- musculo-skeletal system for anticipatory adjustment of human locomotion during obstacle avoidance. *Biological Cybernetics.* 78(1):9-17 (1998).
25. Wadden , T., Ekeberg, O.: A neuro-mechanical model of legged locomotion: single leg control. *Biological Cybernetics.* 79(2), 161-173 (1998).
26. Geng, T., Porr, B., Wörgötter, F.: Fast Biped Walking with a Sensor-driven Neuronal Controller and Real-time Online Learning. *The International Journal of Robotics Research.* 25(3), 243-259 (2006).
27. Nassour, J., Hénaff, P., Ben Oueddou, F., Cheng, G.: Experience-based learning mechanism for neural controller adaptation: Application to walking biped robots. *The IEEE/RSJ International Conference on Intelligent RObots and Systems.* St. Louis, MO, USA. 2616-2621 (2009).
28. Kohonen, T.: *Self-Organizing Maps.* Springer Series in Information Sciences. Springer, Berlin, Heidelberg, New York, 1995, 1997, 2001. Third Extended Edition, 501 pages. ISBN 3-540-67921-9, ISSN 0720-678X.
29. Ahn, H., Picard, R.: Affective-Cognitive Learning and Decision Making: A Motivational Reward Framework For Affective Agent: The 1st International Conference on Affective Computing and Intelligent Interaction. , Beijing, China. 22-24, (2005).