Apprentissage par renforcement appliqué au contrôle moteur : reproduction du principe d'isochronie

Didier Marin, Olivier Sigaud

Institut des Systèmes Intelligents et de Robotique
Université Pierre et Marie Curie - Paris 6, CNRS UMR 7222
Pyramide Tour 55, Boîte courrier 173
4 place Jussieu, F75252 Paris Cedex 05
didier.marin@isir.upmc.fr

Résumé : Notre système moteur se caractérise par une redondance qui laisse une infinité de possibilités quant au mouvement que nous mettons en œuvre. Pourtant, l'expérience nous conduit à sélectionner des comportements bien spécifiques dans une situation donnée. Les spécialistes du contrôle moteur ont observé qu'il existait une relation linéaire entre la distance qui nous sépare d'un objectif et la durée du mouvement que nous effectuons pour l'atteindre, que l'on appelle principe d'isochronie. En robotique, les méthodes de commande optimale ne permettent pas d'obtenir des propriétés similaires. Nous proposons ici un modèle d'apprentissage du contrôle moteur basé sur l'utilisation de méthodes d'apprentissage par renforcement appelées Acteur-Critique. Nous illustrons sur une tâche de pointage simple que ce modèle est capable d'apprendre à réaliser celle-ci en vérifiant le principe d'isochronie.
Mots-clés : apprentissage par renforcement, acteur-critique, états et actions continus

1 Introduction

Lorsque nous cherchons à attraper un objet, une infinité de mouvements nous sont permis grâce à la redondance de notre système musculo-squelettique. En principe, nous pouvons donc réaliser cette tâche d'une façon différente à chaque fois, pourtant nous effectuons toujours des mouvements similaires dans un même contexte. Plus généralement, on observe un certain nombre de propriétés invariantes dans toutes les tâches motrices que nous effectuons, que les spécialistes du contrôle moteur capturent par un ensemble de lois telles que le principe d'isochronie (Viviani & Schneider, 1991), qui relie la distance d'un objet et le temps mis pour l'atteindre, ou le fait que les mouvements de "*reaching*" ont un profil de vitesse en cloche (Flash & Hogan, 1985). Ces caractéristiques se retrouvent même chez de jeunes enfants, mais sont améliorées qualitativement jusque à l'âge adulte par le biais de l'apprentissage (Viviani & Schneider, 1991). Il semble donc que nous optimisons avec l'expérience un ensemble de critères intuitifs, liés à notre système moteur et à l'objectif que nous souhaitons atteindre.

A partir de cette hypothèse, de nombreux modèles ont été élaborés pour identifier ces critères, que l'on représente généralement par une fonction de coût à minimiser : la dérivée de la position dans le temps (*minimum-jerk*), les couples articulaires, l'effort musculaire ou l'erreur de position à la fin du mouvement.

Un cadre qui s'est imposé récemment dans la littérature pour rendre compte du contrôle moteur humain est celui de la commande optimale en feedback (Todorov & Jordan, 2002), qui permet d'optimiser le contrôle d'un système suivant un critère de performance donné. Cette performance s'exprime par la minimisation de l'intégrale sur le temps d'une fonction de coût. Cependant, il apparaît que la commande optimale ne permet pas de reproduire toutes les caractéristiques motrices humaines. En particulier, nous ne disposons pas d'une connaissance exacte de notre système moteur et de l'environnement dans lequel nous agissons, mais seulement d'une connaissance imparfaite que nous acquérons au cours du temps.

L'objet de cette contribution est de montrer qu'un autre cadre, celui de l'apprentissage par renforcement, fournit une alternative intéressante à la commande optimale pour modéliser l'apprentissage moteur humain. Pour mettre en œuvre l'apprentissage par renforcement pour le contrôle moteur, il est nécessaire d'avoir recours à des méthodes dites Acteur-Critique que nous présentons dans la section 2. Puis dans la section 3, nous proposons une expérience illustrant la capacité du cadre de l'apprentissage par renforcement à reproduire le principe d'isochronie et le profil de vitesse en cloche sur une tâche de déplacement d'un

point-masse sur une cible. Les résultats obtenus sont présentés dans la section 4 et discutés dans la section 5. Enfin, nous concluons et donnons les perspectives sur lequelles débouchent notre travail préliminaire dans la section 6.

2 Algorithmes Acteur-Critique avec gradient naturel

L'apprentissage par renforcement regroupe une famille de méthodes de résolution de problèmes de décision dans un environnement inconnu et incertain : un agent doit choisir une action connaissant l'état dans lequel il se trouve, de manière à maximiser sur le long terme la récompense reçue en fonction de ses choix. Au fur et à mesure de ces interactions avec l'environnement, l'agent apprend quelle action il doit effectuer dans un état donné. Dans le cadre du contrôle moteur, l'état correspond à un ensemble d'informations sensorielles telles que la configuration des articulations, la position et la vitesse des mains, et l'action correspond à une commande motrice telle que les ordres de contraction envoyés à nos muscles ou bien, en robotique, les couples appliqués par les moteurs. L'état et l'action sont donc continus et potentiellement de grande dimension. La récompense reçue par l'agent est l'équivalent négatif de la fonction de coût en contrôle optimal, et dépend seulement de l'état dans lequel nous nous trouvons et de l'action que nous venons de réaliser.

Dans ce qui suit, nous introduisons le cadre formel de l'apprentissage par renforcement et son application dans le cas des espaces continus que sont les espaces sensori-moteurs (2.1), ce qui nous conduit à adopter une architecture Acteur-Critique (2.2) avec des approximateurs de fonctions continues.

2.1 Cadre de l'Apprentissage par renforcement dans le continu

On suppose que le problème de contrôle étudié est un Processus de Décision Markovien (MDP) (Puterman, 2005), avec $\mathcal{X}$ l'ensemble des états et $\mathcal{U}$ l'ensemble des actions. La récompense immédiate pour un couple état-action est donné par la fonction de récompense $R : \mathcal{X} \times \mathcal{U} \rightarrow \mathbb{R}$. La distribution de probabilité sur les états suivants pour un couple état-action est donnée par la fonction de transition $P : \mathcal{X} \times \mathcal{U} \rightarrow \Pi(\mathcal{X})$. A chaque pas de temps t, l'agent choisit une action u_t connaissant son état x_t et reçoit de son environnement une récompense immédiate r_t et son état suivant x_{t+1}.

Le choix d'une action en fonction d'un état est déterminé par une densité de probabilité $\pi : \mathcal{X} \times \mathcal{U} \rightarrow [0, 1]$ appelée politique, qui associe à chaque état une distribution sur les actions : $\pi(u|x) = \Pr(u|x, \pi)$.

Soit $J(\pi)$ un critère de performance de la politique π, le problème que l'on cherche à résoudre consiste à trouver la politique optimale π^* telle que $\pi^* = argmax_\pi J(\pi)$. On utilisera ici le critère de récompense actualisée sur un horizon infini :

$$J(\pi) = \lim_{t \rightarrow \infty} E[\sum_{t=0}^{T} \gamma^t r_t | \pi] \tag{1}$$

où $\gamma \in [0, 1]$ est le facteur d'actualisation, qui règle l'importance des récompenses futures. Sutton *et al.* (2000) montrent que l'on peut reformuler le critère de performance par

$$J(\pi) = \int_{\mathcal{X}} d^\pi(x) \int_{\mathcal{U}} \pi(u|x) R(x, u) du dx$$

où d^π la distribution pondérée des états rencontrés en suivant la politique π et sachant la distribution des états initiaux $P_0 : \mathcal{X} \rightarrow [0, 1]$:

$$d^\pi(x) = \sum_{t=0}^{\infty} \gamma^t \Pr(x_t = x | \pi, x_0 \sim P_0)$$

Pour pouvoir évaluer une politique, on définit des fonctions de cette politique appelées fonctions de valeur. La fonction de valeur d'état V^π nous donne, pour un état x, la récompense espérée sur le long terme en suivant la politique π à partir de x :

$$V^\pi(x) = \sum_{t=0}^{\infty} E[\gamma^t r_t | x_0 = x, \pi]$$

La fonction de valeur d'action Q^π nous donne, pour un état x et une action u, la récompense espérée sur le long terme en effectuant l'action u dans l'état x, puis en suivant la politique π :

$$Q^\pi(x,u) = \sum_{t=0}^{\infty} E[\gamma^t r_t | x_0 = x, u_0 = u, \pi]$$

Lorsque les fonctions de transition et de récompense sont inconnues et que les états et les actions sont discrets, des méthodes classiques d'apprentissage par renforcement telles que Q-Learning (Watkins & Dayan, 1992) nous permettent d'estimer Q^π et d'en déduire une politique, en choisissant l'action qui maximise Q^π pour l'état courant. Ces méthodes trouvent leurs limites lorsque les états et/ou les actions sont de grande dimension et/ou continus. En effet, il nous est alors impossible de stocker directement la valeur de tout couple état-action possible. Ce problème peut être résolu par l'utilisation d'approximateurs de fonction de valeur, généralement linéaires, c'est-à-dire sous forme de somme pondérée :

$$f_v(x) = \sum_{i=1}^{n} v_i \phi_i(x) = \phi(x)^\top v$$

où v est un vecteur de paramètres et $\phi = [\phi_1, \phi_2, \ldots, \phi_n]$ un vecteur de fonctions prédéfinies appelées fonctions de base.

Néanmoins, l'utilisation d'approximateurs dans ces méthodes pose des problèmes de convergence et d'instabilité (Gordon, 1995) : lorsque l'on choisit systématiquement l'action qui maximise Q^π (méthode dite *greedy*), de faibles erreurs dans d'approximation peuvent changer radicalement la politique. De plus, ces méthodes ne permettent pas la représentation de politiques stochastiques.

Dans les méthodes de gradient sur la politique (Williams, 1992), on considère une politique paramétrée π_θ avec $\theta \in \mathbb{R}^n$. L'objectif est de trouver les paramètres θ^* qui maximisent un critère de performance $J : \mathbb{R}^n \to \mathbb{R}$. Pour cela, on effectue une descente de gradient sur J afin de converger vers un optimum local[1]. Cette convergence est garantie si le pas d'apprentissage α_t de la descente de gradient vérifie $\lim_{t\to\infty} \alpha_t = 0$ et $\sum_{t=0}^{\infty} \alpha_t = \infty$. Nous simplifierons π_θ par θ dans les formules suivantes.

Sutton *et al.* (2000) montrent que l'on peut écrire le gradient de la performance sous la forme :

$$\begin{aligned} \nabla_\theta J(\theta) &= \int_{\mathcal{X}} d^\theta(x) \int_{\mathcal{U}} \nabla_\theta \pi_\theta(u|x)\, Q^\theta(x,u) dx du \\ &= \int_{\mathcal{X}} d^\theta(x) \int_{\mathcal{U}} \pi_\theta(u|x) \nabla_\theta log \pi_\theta(u|x)\, Q^\theta(x,u) dx du \end{aligned} \quad (2)$$

où ∇_θ est le gradient selon θ : $\nabla_\theta f_\theta(.) = [\frac{\partial f_\theta(.)}{\partial \theta_1}, \frac{d\partial f_\theta(.)}{\partial \theta_2}, ..., \frac{\partial f_\theta(.)}{\partial \theta_n}]$

La distribution des états d^θ est inconnue mais les expériences réalisées nous en fournissent des échantillons x, pour lequel $\nabla_\theta log \pi_\theta(u|x) Q^\theta(x,u)$ est une estimation sans biais de $\nabla_\theta J(\theta)$.

Par ailleurs, on utilise un approximateur linéaire $\hat{Q}^\theta_w$ à la place de Q^θ. Sutton *et al.* (2000) prouvent que pour obtenir une estimation non biaisée du gradient de la performance, cet approximateur doit vérifier la condition de compatibilité suivante :

$$\frac{\partial \hat{Q}^\theta_w(x,u)}{\partial w} = \nabla_\theta \pi_\theta(u|x) \frac{1}{\pi_\theta(u|x)} = \nabla log \pi_\theta(u|x) \quad (3)$$

En pratique, cette estimation souffre d'une forte variance, qui entraîne une convergence très lente. Pour réduire cette variance, deux solutions indépendantes ont été proposées :

Baseline

La première exploite le fait que l'on peut ajouter dans l'équation (2) une fonction quelconque $b : \mathcal{X} \to \mathbb{R}$, appelée *baseline*, à $Q^\theta(x,u)$ sans changer la direction du gradient de la performance :

$$\nabla_\theta J(\theta) = \int_{\mathcal{X}} d^\theta(x) \int_{\mathcal{U}} \nabla_\theta \pi_\theta(u|x) [Q^\theta(x,u) \pm b(x)] dx du$$

[1] On notera que l'optimum global $J(\theta^*)$ ne correspond pas nécessairement à la politique déterministe optimale, puisque les politiques possibles sont limitées par la paramétrisation choisie *a priori*

Bhatnagar *et al.* (2007) montrent que la baseline optimale pour réduire la variance du gradient est en fait V^θ :

$$\begin{aligned}\nabla_\theta J(\theta) &= \int_{\mathcal{X}} d^\theta(x) \int_{\mathcal{U}} \nabla_\theta \pi_\theta(u|x)[Q^\theta(x,u) \pm V^\theta(x)]dxdu \\ &= \int_{\mathcal{X}} d^\theta(x) \int_{\mathcal{U}} \nabla_\theta \pi_\theta(u|x) A^\theta(x,u)dxdu \end{aligned} \quad (4)$$

où $A^\theta(x,u) = Q^\theta(x,u) - V^\theta(x)$ est la fonction avantage de la politique (Baird, 1993).

Gradient naturel

La seconde solution est l'utilisation du gradient naturel de J noté $\hat{\nabla}_\theta J(\theta)$ à la place du gradient classique $\nabla_\theta J(\theta)$. Le gradient naturel, introduit par Amari (1998), se calcule par une transformation linéaire du gradient classique par la matrice inverse d'information de Fisher de θ, notée $F^{-1}(\theta)$:

$$\hat{\nabla}_\theta J(\theta) = F^{-1}(\theta)\nabla_\theta J(\theta)$$

La matrice d'information de Fisher est donnée par :

$$\begin{aligned} F(\theta) &= E_{x\sim d^\theta, u\sim\pi_\theta}[\nabla_\theta \log \pi_\theta(u_t|x_t) \nabla_\theta log \pi_\theta(u_t|x_t)^\top] \\ &= \int_{\mathcal{X}} d^\theta(x) \int_{\mathcal{U}} \pi_\theta(u|x) \nabla_\theta log \pi_\theta(u_t|x_t) \nabla_\theta log \pi_\theta(u_t|x_t)^\top dxdu \end{aligned}$$

Il est donc possible d'estimer $F(\theta)$ à partir d'échantillons puis de l'inverser, ou directement l'inverse en utilisant par exemple le lemme de Sherman-Morrison. Le gradient naturel est intéressant car il pointe plus directement que le gradient classique vers l'optimum local le plus proche, ce qui rend la convergence plus rapide et évite qu'elle soit prématurée. Il hérite des garanties de convergence du gradient classique.

Peters & Schaal (2005) vont plus loin et proposent une méthode pour estimer le gradient naturel sans $F^{-1}(\theta)$. En utilisant $\hat{A}^\theta_w(x,u) = \nabla_\theta \log \pi_\theta(u|x)^\top w$ comme approximation de A^θ, vérifiant bien la condition de compatibilité (3), on peut fait apparaître la matrice de Fisher dans l'estimation du gradient classique (4) :

$$\begin{aligned} \nabla_\theta J(\theta) &= E[\nabla_\theta \log \pi_\theta(u|x) \hat{Q}^\theta_w(x,u)] \\ &= E[\nabla_\theta log \pi_\theta(u|x) \nabla_\theta \log \pi_\theta(u|x)^\top w] \\ &= E[F(\theta)w] \end{aligned}$$

L'estimation du gradient naturel se simplifie alors en :

$$\hat{\nabla}_\theta J(\theta) \approx F^{-1}(\theta)F(\theta)w \approx w$$

Ce résultat nous permet donc d'utiliser le paramètre w comme estimateur du gradient naturel de la performance. La mise à jour de la politique a alors une complexité en $\mathcal{O}(|\theta|)$, au lieu de $\mathcal{O}(|\theta|^2)$ si l'on calcule explicitement $F^{-1}(\theta)$.

2.2 Algorithmes Acteur-Critique avec gradient naturel

Les méthodes de gradient que nous venons de voir ne constituent qu'une partie de la solution, puisqu'elles reposent sur la connaissance d'une approximation de la valeur de la politique π_θ. De plus, après chaque mise à jour des paramètres θ, il nous faut approcher de nouvelles fonctions de valeur. Cette alternance entre évaluation et amélioration de la politique correspond à un cadre appelée Acteur-Critique (Barto *et al.*, 1988) : le Critique évalue la politique et en déduit une amélioration, l'Acteur stocke une politique, agit suivant celle-ci et la modifie suivant les indications données par le Critique. L'idée derrière cette architecture est que l'on résout en parallèle deux sous-problèmes, de prédiction pour le Critique et de contrôle pour l'Acteur. La figure 2 illustre schématiquement leur fonctionnement.

Peters & Schaal (2005) introduit le gradient naturel dans ce cadre avec l' algorithme *Natural Actor-Critic* (NAC). Son Critique approche la fonction avantage et la fonction de valeur par une méthode de moindres carrés sur l'équation de Bellman :

$$A^\theta(x,u) + V^\theta(x) = Q^\theta(x,u) = R(x,u) + \gamma \int_{\mathcal{X}} p(x'|x,u) V^\theta(x')dx'$$

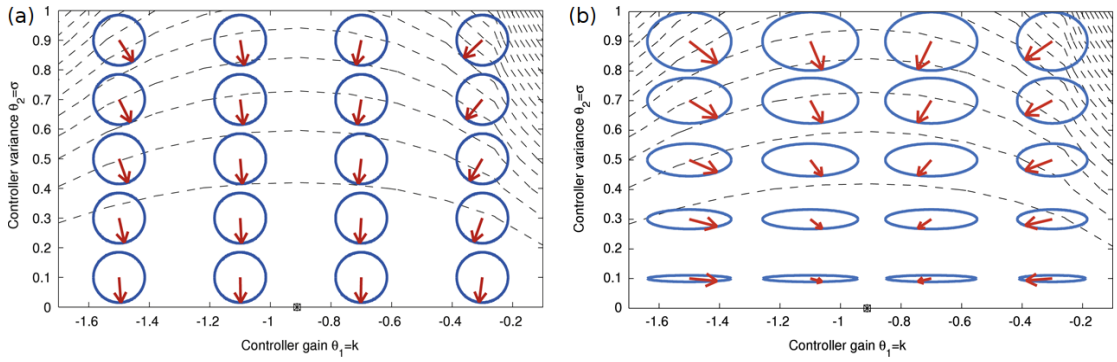

FIG. 1 – Illustration du gradient classique (a) et du gradient naturel (b) sur un problème de régulation linéaire quadratique, empruntée à Peters & Schaal (2005). La politique se compose 2 paramètres : le gain $k = \theta_1$ et l'écart-type d'un bruit gaussien $\sigma = \theta_2$. Le gradient classique va avoir tendance à réduire rapidement σ, ce qui diminue l'exploration, et conduit a une convergence prématurée ($\sigma = 0$ avec un gain k sous-optimal). Le gradient naturel tient compte de l'influence de chaque paramètre, grâce à l'inverse de la matrice d'information de Fisher. En pratique, cela permet de conserver une exploration suffisante pour atteindre le gain optimal.

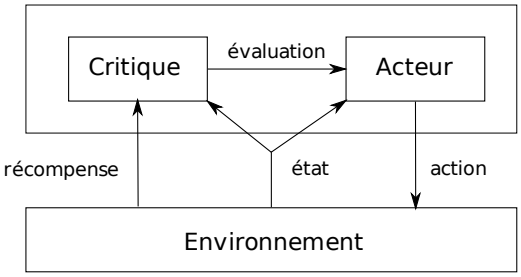

FIG. 2 – Schéma général d'une architecture Acteur-Critique.

En utilisant l'approximateur de la fonction avantage $\hat{A}^\theta_w(x,u) = \psi(x,u)^\top w$ avec $\psi(x,u) \equiv \nabla_\theta \log \pi_\theta(x,u)$, vérifiant la condition de compatibilité (3), on obtient une estimation du gradient naturel, le paramètre w. L'inconvénient majeur de cette approche est la complexité de son Critique, en $\mathcal{O}((n+m)^2)$ où $n = |v|$ et $m = |w|$, qui risque d'être problématique dans les espaces de grande dimension, pour lesquels le nombre de fonctions de base nécessaires est généralement beaucoup plus important.

Il est possible d'obtenir un Acteur-Critique avec gradient naturel avec une complexité en $\mathcal{O}(max(n,m))$. Pour cela, on utilise un Critique qui calcule l'erreur de prédiction de V^θ par la méthode de différence temporelle (TD) (Sutton, 1988). Cette méthode est basée sur la correction de la valeur à partir d'une erreur de prédiction appellée "erreur de différence temporelle" :

$$\delta_t = r_t + \gamma \hat{V}^\theta(x_{t+1}) - \hat{V}^\theta(x_t)$$

L'erreur de différence temporelle a un intérêt tout particulier ici car elle correspond à une estimation de la fonction avantage (Bhatnagar *et al.*, 2007) :

$$E[\delta_t | \pi_\theta] = A^\theta(x_t, u_t)$$

Dans les premiers algorithmes Acteur-Critique, par ex. Sutton *et al.* (2000), elle est utilisée directement comme estimation du gradient (non naturel) de la politique, ce qui donne un algorithme de la forme suivante :

$$\begin{aligned}
\delta_t &\leftarrow r_t + \gamma \phi_{t+1}^\top v_t - \phi_t^\top v_t \\
v_{t+1} &\leftarrow v_{t+1} + \alpha_t \delta_t \phi_t \\
\delta'_t &\leftarrow r_t + \phi_{t+1}^\top v_{t+1} - \phi_t^\top v_{t+1} \\
\theta_{t+1} &\leftarrow \theta_t + \beta_t \delta'_t \psi_t
\end{aligned}$$

où α_t et β_t sont des pas d'apprentissage. Pour que la convergence de l'approximateur de la fonction valeur et de la politique soit garantie, β_t doit tendre vers 0 plus vite que α_t, quand $t \to \infty$. Notons également δ'_t, l'erreur de différence temporelle réestimée après mise à jour de la valeur.

Pour calculer le gradient naturel, nous pouvons optimiser le paramètre w de notre approximateur de l'avantage par une descente de gradient sur l'erreur quadratique $\varepsilon^{\pi_\theta}(w) = E[\hat{A}^\theta(x_t, u_t) - \delta_t | \pi_\theta]$. Nous obtenons alors un algorithme Acteur-Critique appellé *Temporal Difference Natural Actor-Critic* (TDNAC)[2] (Morimura *et al.*, 2005), que nous détaillons dans le cadre Algorithme 1.

Algorithme 1 TDNAC

Initialisation : θ_0, v_0, nombre d'itérations T, pas d'apprentissage $\{\alpha_t\}_{t=0}^T$ et $\{\beta_t\}_{t=0}^T$, facteur d'oubli $\kappa \in [0, 1]$.

$w_0 \leftarrow [0, 0, \ldots, 0]^\top$
$t \leftarrow 0$
tantque $t < T$ **faire**
  Tirer une action $u_t \sim \pi_{\theta_t}(x_t, .)$ et récupérer r_t et x_{t+1}
  Calcul de l'erreur de différence temporelle
    $\delta_t \leftarrow r_t + \gamma \phi_{t+1}^\top v_t - \phi_t^\top v_t$
  Mise à jour de la valeur approchée
    $v_{t+1} \leftarrow v_{t+1} + \alpha_t \delta_t \phi_t$
  Calcul de la nouvelle erreur de différence temporelle
    $\delta'_t \leftarrow r_t + \phi_{t+1}^\top v_{t+1} - \phi_t^\top v_{t+1}$
  Mise à jour de l'avantage / gradient naturel
    $w_{t+1} \leftarrow w_t + \alpha_t \psi_t (\delta'_t - w_t^\top \psi_t)$
  Mise à jour de la politique
    $\theta_{t+1} \leftarrow \theta_t + \beta_t w_{t+1}$
  Oubli partiel du gradient
    $w_{t+1} \leftarrow \kappa w_{t+1}$
  $t \leftarrow t + 1$
fin tantque

Notons l'utilisation d'un facteur d'oubli κ comme dans NAC. D'après Morimura *et al.* (2005), le gradient est biaisé pour $\kappa < 1$ mais, en pratique, cela permet d'éliminer l'information relative aux couples état-action marginaux. La complexité de TDNAC est en $O(n)$ pour l'approximation de V, et en $O(m)$ pour celle de A (et donc du gradient naturel) et la mise à jour de la politique, avec $n = |v|$ et $m = |\theta|$.

On trouvera une autre variante sous le nom de *Incremental Natural Actor-Critic* (iNAC) dans Bhatnagar *et al.* (2007) Alg. 3, mais dans laquelle le facteur d'oubli est omis.

2.3 Features dans des espaces continus

Pour pouvoir appliquer ces méthodes Acteur-Critique à un problème de contrôle, il nous faut encore choisir les fonctions de base ϕ et ψ pour l'approximation des fonctions de valeurs et la représentation de la politique. Ce choix est loin d'être trivial et influe fortement sur la qualité des solutions obtenues : des fonctions de base trop peu nombreuses ou linéairement dépendantes peuvent entraîner la dégénérescence du gradient de la politique, alors qu'un trop grand nombre ralentit fortement la convergence de celle-ci (Rohanimanesh *et al.*, 2007).

Une famille de fonctions de base relativement simples consiste à couvrir les espaces d'état et d'action avec un ensemble de fonctions de base. Dans des méthodes telles que le *Tile Coding* et le *Coarse Coding* (Sutton & Barto, 1998), leur sortie vaut 1 si l'entrée se trouve dans une certaine région de l'espace, et 0 sinon. Etant donné que nous cherchons à approcher des fonctions dérivables, nous nous tournons vers une famille de fonctions de base de forme gaussienne appartenant à la classe des *Radial Basis Functions* (*RBF*) (Park & Sandberg, 1991), très utilisées pour l'approximation de fonctions non-linéaires. Nous utiliserons ici abusivement le terme général *RBF* pour désigner les *RBF* gaussiennes, bien qu'il en existe d'autres formes (hyperboliques *etc*).

[2]Ce nom est dû à Matthieu Geist

Une ***RBF*** ϕ est une fonction de $\mathbb{R}^n$ dans $[0,1]$, définie par :

$$\phi(x) = \exp(-\frac{|c-x|^2}{\sigma^2})$$

où c est un paramètre de centre et σ un paramètre de largeur.

Pour approcher la fonction de valeur d'état, nous pouvons donc utiliser un ensemble de ***RBF***, dont les centres et les largeurs auront été choisis au préalable.

Pour représenter une politique déterministe $\mu_\theta(x)$, nous pouvons utiliser un ensemble de n ***RBF*** normalisées

$$\mu_\theta(x) = \sum_{i=1}^{n} \frac{\phi(x)^\top \theta}{\sum_{j=1}^{n} \phi_j(x)}$$

Afin d'améliorer notre politique, il nous faut introduire une stratégie d'exploration, c'est-à-dire une probabilité non nulle d'essayer d'autres actions que celle que nous indique $\mu_\theta(x)$. L'approche la plus simple consiste à ajouter un bruit gaussien de variance σ_{explo} à l'action moyenne $\mu_\theta(x)$:

$$\pi_\theta(u|x) = \frac{1}{\sqrt{2\pi}\sigma_{explo}} \exp(-\frac{\|u-\mu_\theta(x)\|^2}{2\sigma_{explo}^2})$$

Les fonctions de base compatibles pour l'approximation de l'avantage sont :

$$\frac{\partial \log \pi_\theta(u|x)}{\partial \theta_i} = \frac{\|u-\mu_\theta(x)\|}{\sigma_{explo}^2} \frac{\phi_i(x)}{\sum_{j=1}^{k} \phi_j(x)}$$

3 Modèle élémentaire du mouvement

Nous proposons maintenant une application de l'apprentissage par renforcement sur une tâche de contrôle simple destinée à illustrer ses propriétés. Le système est constitué d'un point-masse de $1\ kg$ contrôlé en accélération sur un axe et qui doit être amené dans une région cible avec une vitesse suffisamment faible pour obtenir une récompense. L'état est constitué de la position et de la vitesse du point, bornées respectivement sur $[0,1]\ m$ et $[-1,1]\ m.s^{-1}$, et l'action est l'accélération appliquée bornée sur $[-1,1]$. La fonction de récompense est $r(x,u) = r_g(x) + r_c(u)$ avec $r_g(x) = 1$ quand la région cible est atteinte, -1 quand si on atteint les bornes de l'espace des états, 0 sinon, et $r_c(u) = c.u^2$ avec $c = 0.01$ un facteur constant pondérant le coût énergétique. Le facteur d'actualisation γ est réglé à 0.99. Quand la cible est atteinte ou quand on atteint les bornes de l'espace des états, la valeur de l'état suivant est considérée comme nulle dans les mises à jour, toute récompense au-delà de la tâche étant considérée comme nulle, et l'épisode se termine. Le nombre de pas au cours d'un épisode est limité à $T = 2000$. L'état initial x_0 est fixé à la position 0.25 avec une vitesse nulle, et l'état cible x_{cible} à la position 0.75 avec une vitesse nulle. On considère que la cible est atteinte quand la distance à l'état cible est inférieure à 0.01 pour la position et pour la vitesse. Le système est simulé par la méthode de Runge-Kutta, avec un pas de temps de $0.01s$.

Nous utilisons pour l'approximation de la valeur et la représentation de politique un même ensemble de 21×21 ***RBF*** équi-réparties dans l'espace des états, c'est-à-dire de coordonnées $\{0.0, 0.05, \ldots, 0.95, 1.0\} \times \{-1.0, -0.9, \ldots, 0.9, 1.0\}$. Pour chaque dimension, la largeur des ***RBF*** est réglée avec un même facteur de densité $d = 1.0$, suivant la largeur de l'espace et leur nombre : $\sigma = d[\frac{1.0-0.0}{21-1}, \frac{1.0-(-1.0)}{21-1}] = [0.05, 0.1]$.

Les paramètres v et θ, de l'approximation de la valeur et de la politique respectivement, sont initialisés à 0. Sans bruit d'exploration, la politique initiale consiste donc à rester sur place, ce qui correspond à l'approximation initiale de la valeur, qui vaut 0 quel que soit l'état. Pour pouvoir effectivement apprendre, nous utilisons un bruit d'exploration constant $\sigma_{explo} = 0.3$, suffisant pour que l'agent trouve rapidement la cible grâce à une accélération aléatoire. L'apprentissage est réalisé par TDNAC, avec les pas α_t et β_t pour le Critique et de l'Acteur, respectivement et sont réglés suivant $\alpha_t = \frac{\alpha_0 \alpha_c}{\alpha_c + t^{2/3}}$ et $\beta_t = \frac{\beta_0 \beta_c}{\beta_c + t}$ avec $\alpha_0 = 0.01$, $\alpha_c = 10^7$, $\beta_0 = 0.001$ et $\beta_c = 10^7$. Nous avons donc bien β_t qui converge vers 0 plus rapidement que α_t.

Nous avons constaté qu'un facteur d'oubli κ inférieur à 1 s'avérait nécessaire pour que TDNAC puisse converger vers une bonne solution : une fois la cible trouvée, l'estimation du gradient des couples états-actions éloignés de la cible doit être oubliée afin d'éviter que la politique ne continue d'évoluer dans ces

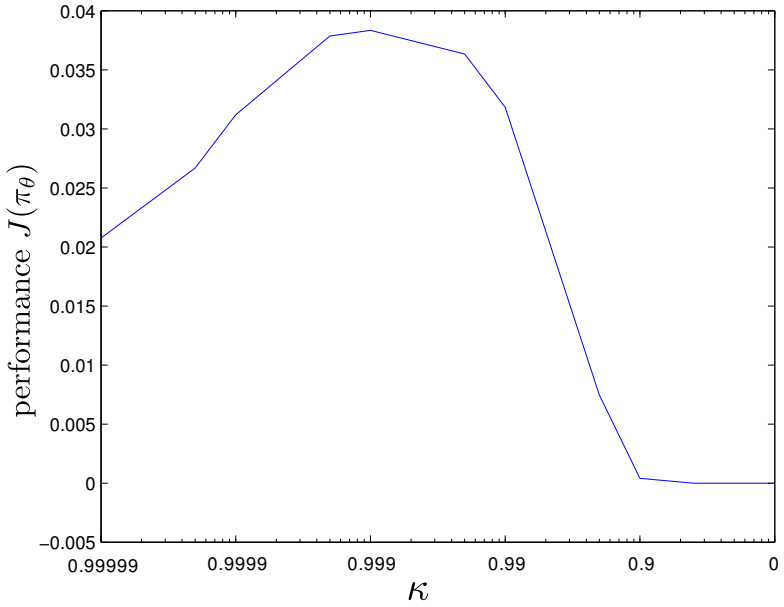

FIG. 3 – Performance de la politique optimisée avec TDNAC (Eq. 1) en fonction du facteur d'oubli κ, après 2×10^4 épisodes. La cible n'est effectivement atteinte que pour $\kappa \geq 0.9$.

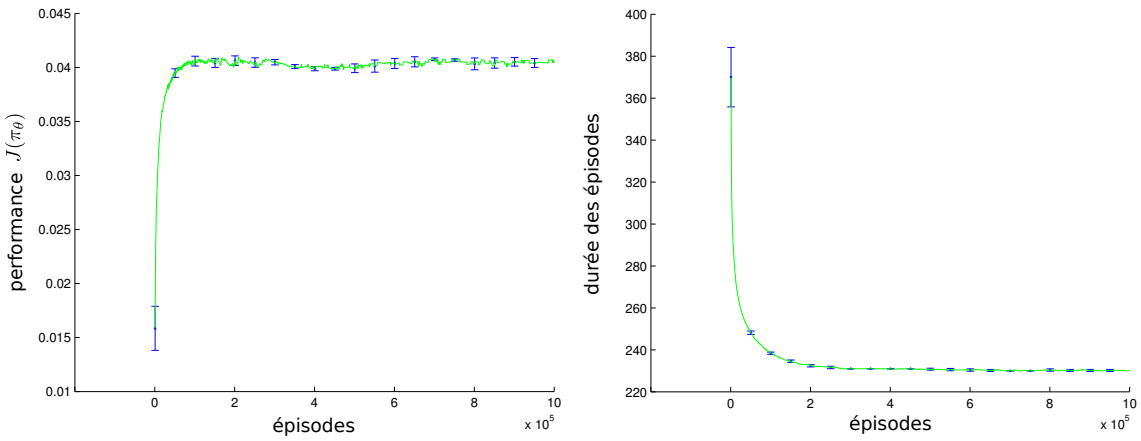

FIG. 4 – Performance de la politique optimisée avec TDNAC (Eq. 1) et durée des épisodes, en fonction du nombre d'épisodes d'apprentissage

régions, qui ne sont que très rarement revisitées, alors que son évaluation par le Critique n'y est plus mise à jour. Après une étude de sensibilité (Fig. 3), nous avons choisi $\kappa = 0.999$.

Pour évaluer notre contrôleur, nous mesurons la performance sur un épisode de la politique déterministe μ_θ.

4 Expérience

Nous commençons par tester la capacité de notre modèle à apprendre la tâche : la Fig. 4 montre la performance en fonction du nombre d'épisodes de TDNAC, moyennée sur 10 apprentissages indépendants, et la Fig. 5 l'approximation de la fonction de valeur après 10^6 épisodes. On observe que le système apprend très vite à atteindre la cible et que la valeur le dirige de façon robuste vers une bonne trajectoire.

Nous comparons ensuite les contrôleurs optimisés pour des cibles dont la position va de $x_{cible} = 0.35$

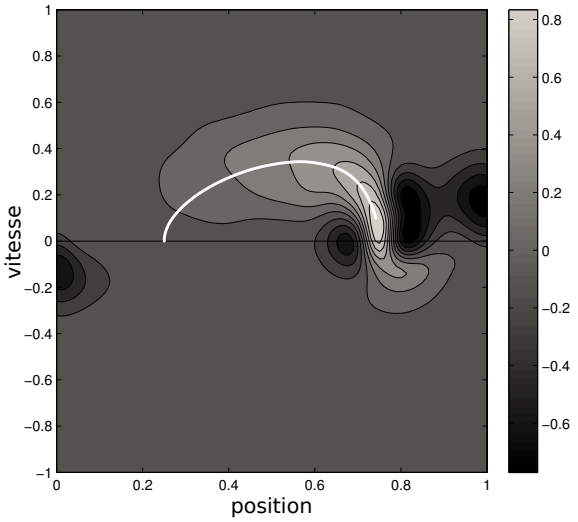

FIG. 5 – Approximation de la fonction de valeur d'état à la fin de l'apprentissage de TDNAC. La trajectoire de la politique déterministe dans l'espace des états est représentée en blanc.

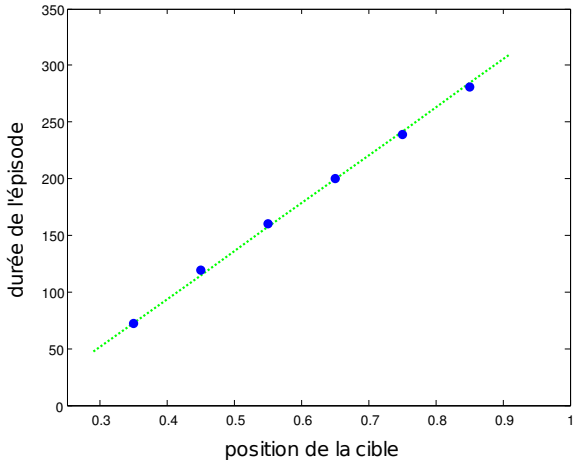

FIG. 6 – Temps mis pour atteindre la cible en fonction de la position de celle-ci. La ligne en pointillés montre la linéarité de la relation entre distance et temps.

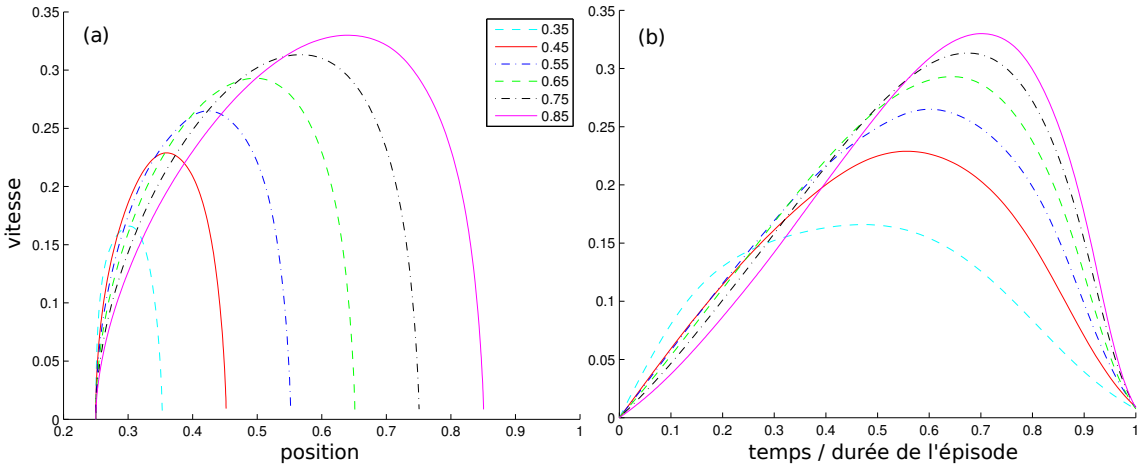

FIG. 7 – Trajectoires dans l'espace des états (a) et profils de vitesse en fonction du temps normalisé pour chaque cible (b).

à 0.85 par pas de 0.1 (l'apprentissage se fait de manière indépendante pour chacune). On observe une relation linéaire entre la distance du point de départ à la cible et le temps mis par le point-masse pour l'atteindre (Fig. 6), ce qui est exactement le résultat que l'on obtient quand on enregistre des mouvements humains, conformément au principe d'isochronie. Les trajectoires dans l'espace des états pour chaque cible sont présentées sur la Fig 7a. Nous comparons les profils de vitesse correspondants sur la Fig. 7b. Ils se caractérisent par une forme en cloche avec une phase d'accélération plus longue que la phase de décélération et un pic de vitesse d'amplitude croissante à mesure qu'on éloigne la cible, ce qui correspond aussi aux attentes.

5 Discussion

La performance du contrôleur appris par notre modèle converge dans l'ensemble des essais vers une valeur positive, ce qui indique que la tâche de pointage est correctement réalisée, d'après la fonction de

récompense choisie. De plus, la relation linéaire entre distance à la cible et temps de mouvement (Fig. 6) montre que notre modèle respecte bien le principe d'isochronie expliqué en introduction. L'approximation de la fonction (Fig. 5) nous montre que la récompense obtenue en atteignant la cible est propagée suivant la trajectoire de la politique, et les puits situés aux bornes des positions admissibles nous indiquent que la punition liée au fait de sortir de cet espace de travail a bien été prise en compte. La déformation des courbes de vitesses (Fig. 7b) peut être imputée à la pondération de la récompense par le facteur d'actualisation γ, qui d'une part n'est pas linéaire en fonction du temps et d'autre part pondère autant la récompense de la cible que les coûts énergétiques : une même accélération sera plus fortement pénalisée dans la performance si elle a lieu plus tôt. Nous observons également que le profil de vitesse pour la cible la plus proche ($x_{cible} = 0.35$) présente une forme un peu différente des autres, ce qui s'expliquer par la granularité insuffisante de la politique représentée par les fonctions de bases.

Plusieurs travaux d'application de l'apprentissage par renforcement à une tâche de pointage ont déjà été entrepris, notamment dans le cas du contrôle d'un bras à 2 degrés de liberté, beaucoup plus représentatifs de nos caractéristiques motrices qu'un simple point-masse (Nagengast *et al.*, 2009). Certains restent dans un cadre purement robotique (Shibata *et al.*, 2000), mais d'autres au contraire tâchent d'imiter le système moteur humain en y intégrant un modèle musculaire (Izawa *et al.*, 2004; Kambara *et al.*, 2009).

Un aspect que nous avons mis de côté ici est la méthode d'exploration, pour laquelle nous avons utilisé un simple bruit gaussien, or il est évident que notre stratégie face à un environnement inconnu et imprévisible dépend fortement de notre estimation de la récompense espérée. En restant dans le cadre d'un bruit gaussien, une solution simple est de considérer σ_{explo} comme un paramètre de la politique (Peters & Schaal, 2005; Ruckstiess *et al.*, 2008). De cette manière, l'exploration est naturellement incluse dans la performance, puisqu'elle est optimisée au même titre que les autres paramètres. L'approximation de la valeur peut être exploitée en choisissant σ_{explo} d'après l'évolution de celle-ci (Shibata *et al.*, 2000). Notons la solution originale adoptée par Izawa *et al.* (2004) qui utilise la raideur des muscles, dépendant d'une commande de co-contraction, comme source d'exploration.

6 Conclusion et perspectives

Dans cette contribution, nous avons montré que l'apprentissage par renforcement est un cadre potentiellement adapté à la modélisation du contrôle moteur humain, en reproduisant quelques invariants simples de ce contrôle. En particulier, nous avons illustré sur une tâche de contrôle simple que le principe d'isochronie, reliant la distance à une cible au temps de mouvement mis pour l'atteindre, et la forme en cloche des profils de vitesse étaient respectés par le critère de performance choisi.

Le cadre de ce travail préliminaire sera par la suite étendu au contrôle d'un bras à 2 degrés de liberté. Il serait alors intéressant de reproduire les expériences de perturbation de trajectoire par un champ de force et de comparer les trajectoires réalisées avec celles de sujets humains (Shadmehr *et al.*, 2005). Nous souhaitons de plus intégrer un modèle des transitions et de la récompense, dont les prédictions permettrait de compenser le délai de la perception sensori-motrice, mais permettrait également de mieux anticiper les changements dans l'environnement (perturbations, mobilité de la cible...) par l'application de techniques de mise à jour indirecte de la politique. Grâce à la possibilité de simulation "dans la tête" de l'agent (Sutton, 1990), nous pourrions également proposer un modèle capable de reproduire la capacité de notre cerveau à améliorer notre contrôle même entre les expériences (Shadmehr *et al.*, 2005).

Références

AMARI S. (1998). Natural gradient works efficently in learning. *MIT Press*, **10**(2), 251–276.

BAIRD L. C. (1993). *Advantage updating*. Rapport interne, Wright Laboratory.

BARTO A. G., SUTTON R. S. & ANDERSON C. W. (1988). Neuronlike adaptive elements that can solve difficult learning control problems. p. 535–549.

BHATNAGAR S., SUTTON R. S., GHAVAMZADEH M. & LEE M. (2007). Natural actor-critic algorithms. In *Twenty First Annual Conference on Neural Information Processing Systems*, p. 105–112.

FLASH T. & HOGAN N. (1985). The coordination of arm movements : An experimentally confirmed mathematical model. *The Journal of Neuroscience*, **5**(7), 1688–1703.

GORDON G. J. (1995). Stable function approximation in dynamic programming.

IZAWA J., KONDO T. & ITO K. (2004). Biological arm motion through reinforcement learning. *Biol Cybern.*, **91**(1), 10–22.

KAMBARA H., KIM K., SHIN D., SATO M. & KOIKE Y. (2009). Learning and generation of goal-directed arm reaching from scratch. *Neural Netw.*, **22**(4), 348–61.

MORIMURA T., UCHIBE E. & KENJI D. (2005). Utilizing the natural gradient in temporal difference reinforcement learning with eligibility traces. In *2nd International Symposium on Information Geometry and its Applications*, Tokyo, Japan.

NAGENGAST A. J., BRAUN D. A. & WOLPERT D. M. (2009). Optimal control predicts human performance on objects with internal degrees of freedom. *PLoS Comput Biol*, **5**(6).

PARK J. & SANDBERG I. W. (1991). Universal approximation using radial-basis-function. *Neural Computation*, **3**, 246–257.

PETERS J. & SCHAAL S. (2005). Natural actor-critic. In *Proceedings of the Sixteenth European Conference on Machine Learning*, p. 280–291 : Springer.

PUTERMAN M. L. (2005). *Markov decision processes : Discrete stochastic dynamic programming*. Wiley & Sons, Inc.

ROHANIMANESH K., ROY N. & TEDRAKE R. (2007). *Towards Feature Selection In Actor-Critic Algorithms*. Rapport interne, MIT.

RUCKSTIESS T., FELDER M. & SCHMIDHUBER J. (2008). State-dependent exploration for policy gradient methods. In *ECML PKDD 2008, Part II, LNAI 5212*, p. 234–249 : Springer.

SHADMEHR R., DONCHIN O., HWANG E.-J., HEMMINGER S. E. & RAO A. (2005). *Motor Cortex in Voluntary Movements : A distributed system for distributed function*, chapter Learning Dynamics of Reaching, p. 297–328. CRC Press.

SHIBATA K., SUGISAKA M. & ITO K. (2000). Hand reaching movement acquired through reinforcement learning.

SUTTON R. & BARTO A. (1998). *Reinforcement Learning : An Introduction*. MIT Press, Cambridge, MA.

SUTTON R. S. (1988). Learning to predict by the method of temporal differences. *Machine Learning*, **3**, 9–44.

SUTTON R. S. (1990). Integrated architectures for learning, planning, and reacting based on approximating dynamic programming. In *Proceedings of the Seventh International Conference on Machine Learning*, p. 216–224.

SUTTON R. S., MCALLESTER D., SINGH S. & MANSOUR Y. (2000). Policy gradient methods for reinforcement learning with function approximation. *Advances in Neural Information Processing Systems*, (12), 1057–1063.

TODOROV E. & JORDAN M. I. (2002). Optimal feedback control as a theory of motor coordination. *Nature neuroscience*, **5**(11), 1226–1235.

VIVIANI P. & SCHNEIDER R. (1991). A developmental study of the relationship between geometry and kinematics in drawing movements. *Journal of Experimental Psychology : Human Perception and Performance*, **17**(1), 198–218.

WATKINS C. & DAYAN P. (1992). Q-learning. *Machine Learning*, **8**(3), 279–292.

WILLIAMS R. J. (1992). Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine Learning*, **8**, 229–256.