Dyna-Q parallèle

Guillaume J. Laurent, Emmanuel Piat
Laboratoire d'Automatique de Besançon UMR CNRS 6596
25, rue Alain Savary, 25000 Besançon, France
E-mail :glaurent@ens2m.fr, epiat@ens2m.fr, web : lab.ens2m.fr

Mot-clés : programmation dynamique, apprentissage par renforcement, Q-Learning, Dyna-Q, architecture comportementale, microrobotique, micromanipulation

1 Introduction

Les phénomènes physiques qui se révèlent aux micro-échelles sont complexes et leur modélisation est difficile. Par conséquent, dans certains cas de figure, il est particulièrement délicat de prévoir la dynamique d'interaction entre un micromanipulateur ou un microrobot et son environnement micrométrique ou millimétrique. Dans un tel contexte, l'apprentissage par renforcement offre un cadre formel intéressant pour réaliser la synthèse de contrôleurs sans connaissance *a priori*.

Lorsqu'on considère la complexité des systèmes étudiés en microrobotique[1], les algorithmes classiques d'apprentissage par renforcement ne sont généralement pas utilisables car ils sont limités par leur vitesse d'apprentissage à des systèmes dont l'espace d'états est de faible dimension. Il est cependant possible de diviser le problème global de commande en de multiples sous-problèmes plus simples dont l'apprentissage par une méthode classique ne présente pas de difficulté. Notre approche s'inspire donc à la fois de l'apprentissage par renforcement et des architectures multi-comportements (ou architectures comportementales) qui sont capables de produire des comportements complexes à partir de comportements plus élémentaires.

2 Objectifs

Notre premier objectif était d'implémenter par apprentissage des stratégies de commande complexes à partir d'une approche comportementale. Schématiquement, l'avantage des architectures comportementales est qu'elles permettent de réaliser la commande de systèmes complexes à l'aide de modules élémentaires plus simples à concevoir. L'inconvénient est que les comportements sont généralement spécifiés de manière *ad hoc* et leur coordination peut parfois conduire à des cycles oscillants dus à la présence de maxima locaux. Dans de nombreux cas de figures, la coordination conduit donc à un blocage du système.

1. Typiquement, le cardinal de l'espace d'états considéré est supérieur à 10^{48}.

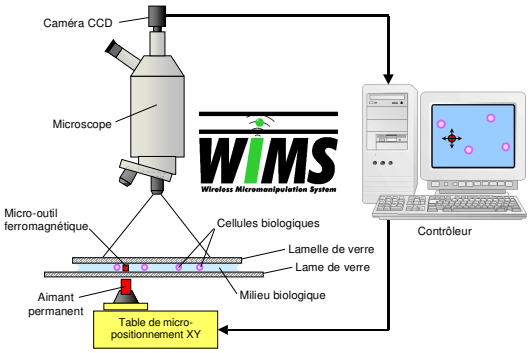


FIG. 1 – *Le micromanipulateur de cellules WIMS (WIreless Micromanipulation System).*

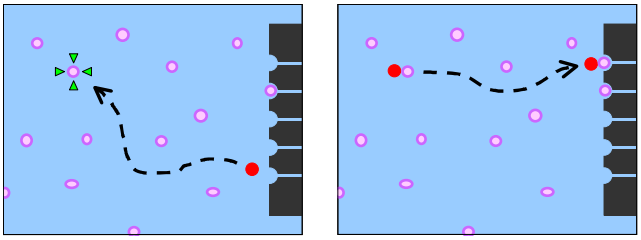


FIG. 2 – *Objectif applicatif du contrôleur : commande du micromanipulateur de cellules WIMS (WIreless Micromanipulation System).*

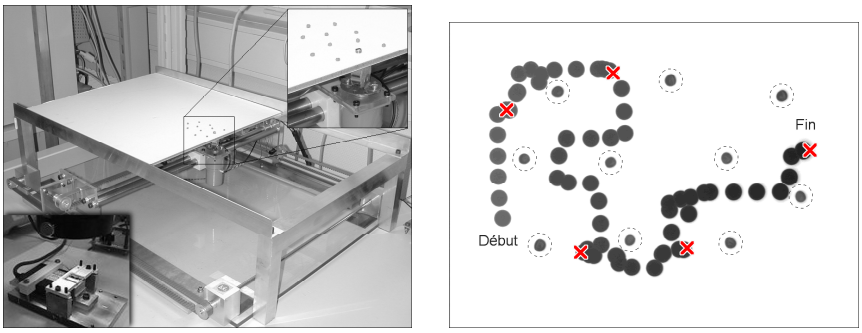
Notre deuxième objectif était d'obtenir des durées d'apprentissage très courtes, de l'ordre de quelques dizaines de minutes. On visait en effet la commande d'une application réelle de manipulation de cellules biologiques : le convoyage vers un banc de test de cellules sélectionnées dans une population donnée (*cf.* figures 1 et 2).

3 Dyna-Q parallèle

Après avoir étudié les différents algorithmes d'apprentissage par renforcement et les architectures comportementales, nous nous sommes orientés vers une architecture basée sur la parallélisation de l'algorithme du Dyna-Q (Sutton, 1990).

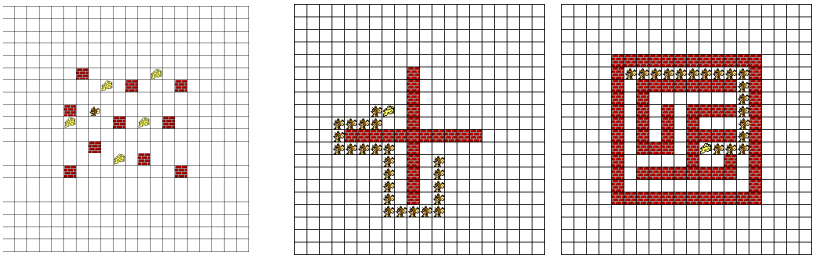
Cette nouvelle architecture de contrôle par apprentissage est adaptée à la commande de systèmes dont l'état est de grande dimension et peut se décomposer en une *situation*. Une situation est définie par le produit cartésien de plusieurs variables markoviennes appelées *perceptions* appartenant à un unique ensemble $\mathcal{X}$. On définit alors un signal de récompenses multiples dont chaque récompense scalaire caractérise l'évolution d'une des perceptions de la situation courante.

Le fonctionnement de l'architecture parallèle consiste à réaliser un apprentissage type Dyna-Q pour chaque perception en utilisant toujours une même



(a) Manipulateur WIMS (b) Trajectoire obtenue

FIG. 3 – *Exemple de trajectoire obtenue avec le Dyna-Q parallèle sur le manipulateur WIMS en une heure d'apprentissage.*



(a) Labyrinthe d'entraînement (b) Labyrinthes de test

FIG. 4 – *Exemples de trajectoires obtenues sans réapprentissage avec un contrôleur à fusion de modèles sur l'exemple classique du labyrinthe discret (grid-world).*

fonction d'utilité et un même modèle de transition définis sur l'ensemble $\mathcal{X}$. Cette architecture, baptisée *Dyna-Q parallèle*, permet de réduire la complexité du système et de réaliser une amélioration importante de la vitesse de convergence. De plus, nous avons adapté l'algorithme du Dyna-Q à la commande de systèmes non déterministes en ajoutant un historique des dernières transitions.

Pour choisir une commande, il est nécessaire de regrouper les espérances de gain données par la fonction d'utilité pour chaque perception. Une première solution consiste à utiliser une fonction de fusion des espérances de gain de chaque perception. Nous avons ainsi étudié l'opérateur de fusion « somme ».

4 Résultats expérimentaux

Les expérimentations montrent que l'apprentissage est possible directement sur le système en temps réel et sans utiliser de simulation (*cf.* figure 3).

La fonction de fusion initiale par somme des espérances de gain fonctionne bien si les obstacles sont relativement éloignés les uns des autres. Si ce n'est pas le cas, elle peut créer des maxima locaux qui entraînent le système dans un

cycle répétitif. Pour pallier ce problème, nous proposons une autre fonction de fusion qui, cette fois, synthétise un modèle plus global du système à partir du modèle de transition des perceptions généré par le Dyna-Q. A chaque période d'échantillonnage, la commande choisie est la première commande optimale proposée par l'étape de planification effectuée sur le modèle global. Cette nouvelle fonction de fusion permet de sortir de ces maxima locaux (*cf.* figure 4). L'application en temps réel est possible car la planification nécessite peu de cycles de programmation dynamique, le modèle global évoluant peu entre deux périodes d'échantillonnage.