

La robotique développementale

ROS QUENTIN
SALINAS ROMAIN
DIOP MAMADOU
FLORIMOND JOUFFROY
LAVAL ARNAUD
FRANCO MAXIME

Définition générale

Avant même de définir la robotique développementale, une définition de la robotique est nécessaire.

1. La robotique

Qu'est-ce donc que la robotique ? La robotique est l'ensemble des techniques permettant la conception, la réalisation de machines automatiques ou d'appareils étant programmés dans le but d'effectuer une tâche précise (Plus communément appelé « **Robot** »).

Venons-en donc à la robotique développementale, quelle est la différence entre la robotique « normale » et la robotique développementale ?

2. De la robotique à la robotique développementale

A la différence de la robotique « normale », la robotique développementale consiste à inclure, dans la programmation du robot, la notion de **correction d'erreur**.

La robotique « normale » est un processus dit de **Markov**, ce qui signifie qu'elle ne considère pas les actions passées afin d'effectuer ses prochaines actions. Elle suit un programme qui ne considère que le temps présent dans ses actions.

La robotique développementale, quant à elle, est un processus dit **non-Markovien**. Elle va donc agir en prenant en compte ses actions passées. Ces robots-là auront des connaissances très maigres à leur sortie d'usine et le but sera donc de les « éduquer », comme nous éduquerions un enfant.

3. Comment en sommes-nous arrivé là ?

L'idée d'apprentissage en robotique est en lien évident avec ce qu'on observe chez les êtres vivants et c'est ce rapprochement qui est à l'origine de ce qu'on appelle la robotique développementale.

L'idée principale de la robotique développementale est celle de mettre au point des mécanismes permettant à des robots d'apprendre des savoir-faire nouveaux pour s'adapter et interagir dans des environnements physiques et sociaux inconnus et changeants.

Spécifications techniques

1. Langage de programmation

On peut choisir de construire un **robot non programmable**, en s'inspirant des BEAMs (un concept robotique, qui cherche à construire des robots avec le minimum de composants électronique). Ainsi on appréhende les principes de sensation/réaction utilisés sur la plupart des robots, en connectant par une électronique simple un capteur à un actionneur.

On peut aussi choisir de programmer un robot **sans langage écrit**, les signes de communication étant alors visuels pour limiter les possibilités grammaticales : au lieu d'avoir une dizaine de mots-clés comme dans un langage informatique classique (if, goto, write, ...), on utilise des blocs graphiques dont les connexions sont limitées. Un autre avantage de ce type de programmation est la lecture d'un seul coup d'œil de l'ensemble du programme (éventuellement avec des moyens d'encapsulation pour créer des blocs personnels eux-mêmes définis par un graphe visuel). En exemple, on peut citer le Lego Mindstorms, il y a également les robots Mavin utilisé.

Il est également possible d'utiliser un langage plus évolué, comme le langage **C** voire même un langage objet, comme le **C++**, le **C#**, le **Java**, etc... Même dans le cas d'un langage graphique, il y a souvent une interprétation dans un langage écrit, comme c'est le cas pour le Mavin dont l'éditeur permet de passer du graphe au programme C.

Mais aujourd'hui avec les besoins de la robotique développementale de nouveaux langages, qui n'existaient pas en informatique conventionnelle ont vu le jour, ils sont pensés uniquement pour la robotique.

C'est le cas des **langages de script**, permettant une reconfiguration rapide d'éléments majeurs d'un programme robotique : gestion des capteurs, réactions aux événements internes et externes, enchainements de mouvements, algorithmes de traitement du signal. Le principal, est **Urbiscript**, créé par la société Gostai qui l'a rendu Open Source. Ce langage écrit s'accompagne d'un environnement graphique, permettant de mêler les deux options évoquées précédemment.

S'il ne s'agit pas d'un langage spécifique, une mention doit être faite au **ROS**, un système d'exploitation pour robots de la société Willow Garage, qui utilise des langages et des concepts informatiques connus (Linux, modules noyau, C++) pour permettre de construire le logiciel d'un robot complexe.

2. Fonctionnement

Avant même de parler du fonctionnement du robot une fois « lâché » dans son environnement, il faut préciser les quelques préparations nécessaires à l'évolution de celui-ci.

Dans un premier temps, il est indispensable de fournir au robot des **schémas de base**. Ces schémas sont là afin d'indiquer au robot les actions qu'il a à effectuer durant le début de son existence. Ces schémas contiennent :

- Les actions **principales** à effectuer, celles qui sont le plus optimales dans le milieu dans lequel le robot évolue,
- Les actions « **secondaires** », celles qui ne sont à considérer qu'en cas d'échec du robot lors de l'exécution d'une action principale.

Une fois les conditions initiales renseignées, on peut « lâcher » le robot dans son environnement. C'est maintenant que le processus démarre.

Premièrement, le robot va appliquer le schéma initial à l'environnement. En cas de réussite, il continuera d'appliquer ce schéma jusqu'à un échec ou jusqu'à la réussite de sa tâche. En cas d'échec, il **mettra à jour le schéma** afin qu'il n'applique plus cette action à ce moment donné.

Ensuite, le robot appliquera le schéma mis à jour à l'environnement et effectuera la première étape avec le schéma mis à jour en boucle jusqu'à tomber sur le **schéma « parfait »**.

Afin que le robot sache quelles actions il doit conserver et quelles actions il ne doit plus faire, on associe chacune d'entre elles à un facteur, à une **variable « réussite »**. Si l'action effectuée par le robot est « **productive** » (Si elle permet de réduire le temps / le nombre d'actions pour effectuer la tâche) alors cette variable « réussite » sera incrémentée. Dans le cas contraire, elle sera décrémentée. L'objectif étant d'obtenir un schéma final avec lequel cette variable réussite ne diminue plus : Le robot n'effectuera que les actions à valeur nulle ou positives.

Dans le but d'expliquer ce comportement de manière plus clair, voici un exemple tiré du projet Ernest :

On va mettre un robot dans un environnement avec un chemin au milieu de murs (Voir figure 1). L'objectif de ce robot : Parcourir le chemin sans se cogner aux murs et en le moins de temps possible.

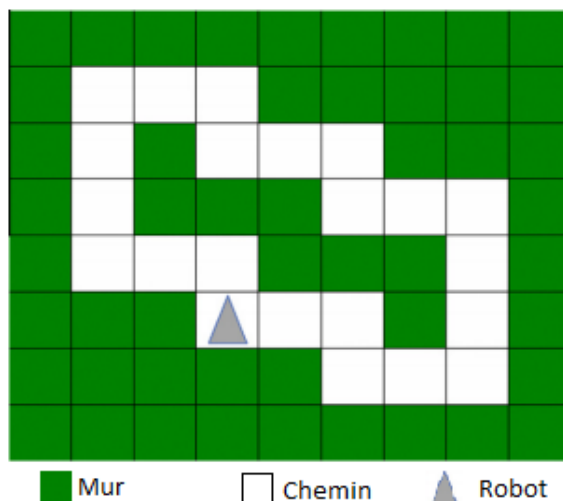


Figure 1 : Robot devant trouver son chemin

Dans ce cas, le schéma initial sera « Faire avancer le robot ». Il pourra effectuer cette action une fois avant d'échouer (Rencontre d'un mur), moment auquel il devra mettre à jour son schéma initial. A chacune de ces actions est associée une valeur de **productivité** (Voir figure 2)

Interaction	Description	Proclivity
→	Move forward	10
→	Bump wall	-10
↘	Turn to the right toward an empty square	0
↘	Turn to the right toward a wall	-5
↙	Turn to the left toward an empty square	0
↙	Turn to the left toward a wall	-5
-	Touch a wall ahead	0
=	Touch an empty square ahead	-1
\	Touch a wall on the right	0
\	Touch an empty square on the right	-1
/	Touch a wall on the left	0
/	Touch an empty square on the left	-1

Figure 2 : Valeur des actions effectuées

Ici, on voit que l'action « Avancer » retourne une productivité élevée, alors que le fait de se « Cogner à un mur » est **anti-productif**. D'après ces chiffres, le robot va voir la productivité de ses actions varier et va mettre à jour son schéma jusqu'à ce que sa productivité ne baisse plus (Lorsque qu'il saura dans quel ordre il faut répéter les actions « Avancer », « Se tourner à droite vers un bloc vide » et « Se tourner à gauche vers un bloc vide »).

Milieu d'activité

1. Dans quel milieu les trouve-t-on ?

Les robots sont déjà chez nous, des robots qui font le ménage comme, par exemple, les aspirateurs autonome. Mais aussi les jouets pour enfants ou adulte.

La robotique développementale est, quant à elle, moins présente pour le moment. Elle est et sera présente dans différents domaines comme :

- **Dans nos foyers**, avec des robots qui nous assisteront dans la vie de tous les jours (Aller nous préparer un café et nous l'amener par exemple ou bien les robots pour enfant. Mais tous ces robots sont, à l'heure actuelle, en phase de développement. On peut déjà parler de « Icube » le robot-enfant qui apprend comme un jeune humain. Ou bien on commence à avoir des jouets comme par exemple un chien-robot qui va chercher une en essayant d'optimiser le plus possible le temps qui lui est nécessaire ;
- **Dans le domaine aéronautique**, avec, par exemple, Kirobo, le robot envoyé dans la station ISS pour étudier l'apport des robots dans le cadre des missions spatiales
- **Dans l'armée.**

Comme nous l'avons déjà précisé, ces robots ont besoin d'apprentissage dans la majorité des domaines « importants », comme par exemple **la santé, l'espace, l'armée** ... C'est pour cela que de tels projets ne verront le jour que dans quelques années et que nous voyons, donc, très peu de robots développementaux.

2. Les projets existants

Les deux principaux projets trouvés sont les projets **Ernest** et **Ideal**. Ces deux projets sont tout de même liés du fait que le projet Ideal est une sorte de suite au projet Ernest.

Le projet Ernest, qui a débuté en Septembre 2008 et a duré cinq ans et demi, consistait à créer des intelligences artificielles dotées d'un algorithme pour la robotique développementale afin de tester l'évolution du comportement de ces agents.

Suite à cela, l'un des chercheurs du projet Ernest s'est lancé sur le projet Ideal consistant à intégrer les mécanismes d'apprentissage dans un environnement de simulation. Le principe est très similaire aux travaux du projet Ernest et est prévu pour durer une quinzaine d'années

Sources

<http://www.syrobo.org/la-robotique-de-services/>

<http://liris.cnrs.fr/ideal/>

<http://liris.cnrs.fr/ideal/doc/GeorgeonO2011-emergent-cognition.pdf>

<http://www.futura-sciences.com/magazines/high-tech/infos/dossiers/d/robotique-icub-robots-service-1143/>

<http://depinfo.u-cergy.fr/~pl/docs/IAE07Oudeyer.pdf>

Fiche descriptive

Titre	La Robotique développementale
Tuteur	Amélie Cordier
Membres du groupe	Ros Quentin Salinas Romain Franco Maxime Diop Mamadou Jouffroy Florimond Laval Arnaud
Résumé du projet	Développement d'un robot utilisant la technologie développementale
Technologies envisagées pour la réalisation du projet	Langage de programmation avancée (C++, C#, Java, etc ...)
Méthodes de gestion de projet	
Probabilité de poursuivre ce projet au S3	Nulle. Désintéressement de la majeure partie du groupe.
Résumé des difficultés rencontrées lors de la réalisation du projet	Recherche de documents (Très peu de projets à l'heure actuelle), Simplification du jargon scientifique, Illustration de nos propos