

Projet ANDROIDE : cahier des charges

Parham SHAMS, Tanguy SOTO

UPMC 2017

1 Contexte, définition du problème et objectifs

1.1 Contexte

Le projet se déroule à l'ISIR (Institut des Systèmes Intelligents et de Robotique) à l'Université Pierre et Marie Curie. Nous disposons d'une arène de 2.5m sur 2.5m, dans laquelle peuvent se déplacer une trentaine de robots à roue Thymio.



FIGURE 1 – Un robot thymio

Ces robots font 12cm de côté, et sont dotés de deux roues motrices. Ils

perçoivent leur environnement par (entre autres) des senseurs de proximité (5 à l'avant et 2 à l'arrière) et à l'aide d'une caméra. Chaque robot est contrôlé par une carte Raspberry PI 3.

L'interaction entre l'ordinateur central et les différents robots se fait à l'aide de la plate-forme OctoPY développée en Python par Arthur Bernard (ancien doctorant à l'ISIR sous la supervision de Nicolas Bredeche, ISIR - UPMC, et de Jean-Baptiste André, ISEM - CNRS). Le code relatif à cette plate-forme se situe sur le git publique [1].

1.2 Définition du problème

La robotique devient de plus en plus omniprésente de nos jours et les comportements des robots font l'objet de beaucoup de recherches. Ainsi, une des pistes principales pour définir un comportement est l'apprentissage qui, s'il est correctement mis en place, permet à un robot d'appréhender des situations inconnues et variées.

Dans ce sens, beaucoup de travaux sont menés afin d'implémenter ce type de comportements via l'apprentissage embarqué et en ligne, c'est à dire depuis le robot lui même. La difficulté majeure rencontrée dans ces travaux réside dans l'aspect empirique des paramètres à fixer lors des tests menés. De plus, beaucoup de travaux et d'articles sur le sujet se basent sur des tests via simulateurs. Ainsi, transposer ces travaux sur des robots réels peut potentiellement amener beaucoup de problèmes liés aux différences entre la théorie pure et la pratique.

1.3 Objectifs

Le premier objectif de ce projet est de faire évoluer un essaim de robots via un algorithme d'apprentissage en ligne et distribué sur cet essaim comme décrit dans les articles [2] et [3]. Les comportements souhaités sont le suivi de lumière et pourquoi pas un comportement de transport collectif. De plus, il s'agira de mettre en place l'expérience témoin d'un apprentissage en ligne d'un seul robot comme décrit dans l'article [4].

L'autre objectif tout aussi essentiel, voire indispensable, de ce projet est l'amélioration de la plate-forme OctoPy. En effet, elle a été mise en place très rapidement et sans beaucoup de tests. Ainsi, il s'agira de la rendre le plus robuste possible, tout en améliorant ses performances et en agrandissant son champ d'actions.

Ce deuxième aspect du projet n'est pas à négliger car la bonne fonctionnalité de la plate-forme et sa robustesse sont indispensables à la mise en place des différents algorithmes attendus et permettront de travailler plus sereinement.

De plus, on pensera aux futures personnes travaillant sur OctoPY en leur permettant de travailler dans les meilleures conditions et en leur donnant l'opportunité de la prendre en main de la manière la plus intuitive possible.

2 Contraintes et exigences

2.1 Contraintes géographiques

Comme énoncé précédemment, le projet se déroule à l'ISIR à l'UPMC. Ainsi, les tests réels sur les robots ne pourront se faire qu'à l'arène située à cet endroit. Ainsi, il faudra se déplacer pour travailler même s'il sera possible d'emprunter un robot pour faire des tests sur un seul robot en dehors de l'arène. Cependant, la plate-forme est prévue pour fonctionner seulement à l'arène. Sa fonctionnalité ailleurs que dans l'arène sera donc à notre charge. De plus, dans les buts d'appréhender et d'améliorer la plate-forme OctoPY, nous serons amenés à prendre contact avec son concepteur Arthur Bernard. Cependant, celui-ci ne sera présent que les deux premières semaines (pendant lesquelles il pourra nous faire une démonstration de la plate-forme) avant de se rendre en Lausanne, rendant plus difficiles les interactions avec lui et les questions éventuelles que nous aurions à lui poser.

2.2 Contraintes matérielles

Comme évoqué précédemment, nous aurons à disposition une trentaine de robots (Thymio+Raspberry PI 3+caméra). Cependant, nous n'avons que 10 robots à disposition pour le moment et nous recevrons pendant le projet une vingtaine de robots supplémentaires que nous devrons préparer et rendre apte à tourner avec la plate-forme.

2.3 Exigences

Il est attendu que notre travail soit régulièrement mis à jour sur le git afin de pouvoir suivre l'avancée et pouvoir être guidé en fonction de ce qui a été fait. De plus, il est essentiel de faire un suivi du travail pour les mêmes raisons que précédemment. Enfin, il pourrait s'avérer utile d'avoir un fichier rassemblant les objectifs fixés avec les délais correspondants.

3 Résultats attendus

Comme développé précédemment, le projet comporte deux objectifs principaux qui vont ensemble. Ainsi, nous sommes libres dans le choix de l'ordre dans lequel nous traiterons ces objectifs même si une plate-forme robuste et fonctionnelle nous permettrait d'avancer plus rapidement sur l'aspect algorithmique et purement théorique de ce projet.

De plus, en ce qui concerne la tâche de suivi de lumière, il sera apprécié de collaborer avec un étudiant du Master Imagerie. En effet, il pourra nous aider dans les algorithmes de capture d'images et de calcul sur les images. En retour, nous pourrions lui apporter notre expérience sur les algorithmes comportementaux. Enfin, l'algorithme décrit en [2] et [3] suppose que les robots ne communiquent qu'avec leurs voisins. Or, actuellement, un robot ne peut avoir connaissance de ses voisins et nous serons (dans l'état actuel des choses) contraints de faire communiquer chaque robot avec tous les autres. Ainsi, cette collaboration pourrait nous permettre de mettre en place un système de reconnaissance des robots alentour via la caméra.

Références

- [1] N. Bredeche, “thymiopypi,” Janvier 2015. <https://github.com/nekonaute/thymioPYPI>.
- [2] J.-M. Montanier, S. Carrignon, and N. Bredeche, “Behavioural specialisation in embodied evolutionary robotics : Why so difficult?,” *Frontiers in Robotics and AI*, vol. 3, no. 38, 2016.
- [3] R. A. Watson, S. G. Ficici, and J. B. Pollack, “Embodied evolution : Distributing an evolutionary algorithm in a population of robots,” *Robotics and Autonomous Systems*, vol. 38, pp. 1–18, 2002.
- [4] N. Bredeche, E. Haasdijk, and A. E. Eiben, “On-line, on-board evolution of robot controllers,” *Lecture Notes on Computer Science*, vol. 5975, pp. 110–121, 2010.