Extension d’une librairie logicielle pour le contrôle sensorimoteur à différents robots simulés et physiques

Projet proposé par Clément Moulin-Frier[1] (Inria-Flowers[2]) et Michael Garcia-Ortiz[3] (City University, Londres, UK)
Contacts : clement.moulin-frier@inria.fr ; Michael.Garcia-Ortiz@city.ac.uk

Ce projet vise à étendre une librairie de programmation de comportements réactifs sur des robots simulés. Cette libraire (décrite ci-dessous) ne permet pour l’instant que le contrôle de robots spécifiques dans un simulateur 3D. L’objectif est de pouvoir également l’interfacer avec un simulateur 2D ainsi qu’un robot mobile physique.

Mots clés : simulation robotique, robotique mobile, systèmes multi-agents, behavior-based robotics, supports pédagogiques, jupyter notebooks

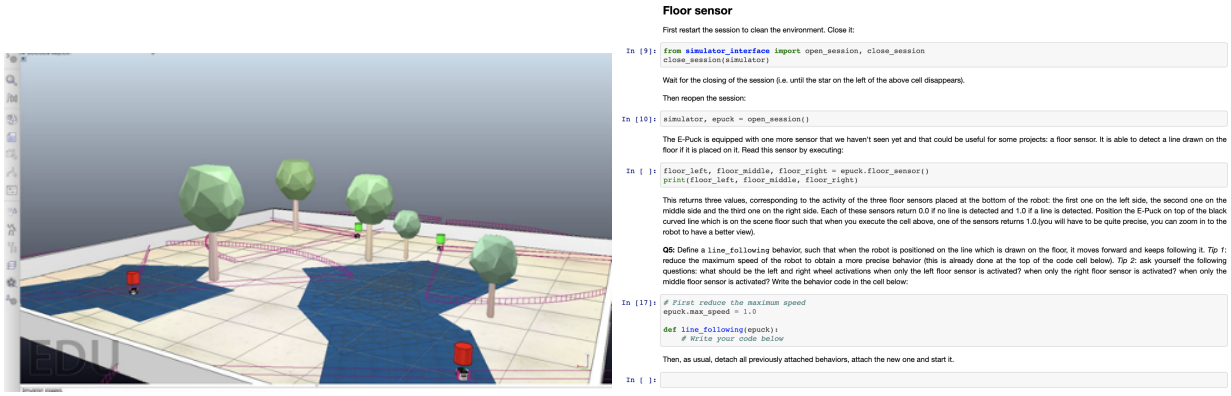

Figure 1. La librairie pyvrep-epuck fournit une plateforme éducative pour aborder des problèmes de robotique, d’apprentissage machine et de simulation multi-agent de façon interactive et intuitive. Elle repose actuellement sur le couplage d’un simulateur robotique 3D (à gauche) et de Jupyter Notebook (à droite). Ces derniers permettent de réaliser des supports de travaux pratiques interactifs, sous la forme de pages web (contenant du texte, des images, des liens …) dans lesquelles les étudiants peuvent exécuter du code Python. La librairie pyvrep-epuck permet ainsi de contrôler le simulateur directement depuis ces pages web (lecture des capteurs des robots, envoi de commandes motrices, contrôle d’éléments de l’environnement). Elle permet de définir des boucles de contrôle de manière simple, sous la forme de fonctions prenant en argument les valeurs des capteurs et retournant des activations sur les moteurs. La capture d’écran du simulateur provient du projet d’un groupe d’étudiants : deux robots « prédateurs » (en rouge) implémentent des boucles de contrôle pour chasser deux robots « proies » (en vert clair) dans un environnement reproduisant un cycle de saisons été-hiver.

[1] http://clement-moulin-frier.github.io/
[2] https://flowers.inria.fr/
[3] https://www.city.ac.uk/about/people/academics/michael-garcia-ortiz

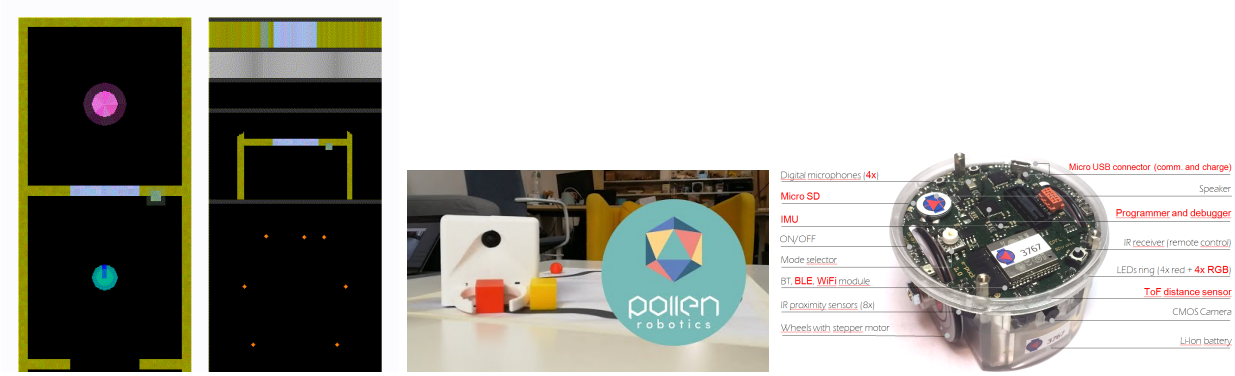

Figure 2. L'objectif du projet est d'étendre la librairie pyvrep-epuck pour pouvoir l'interfacer avec d'autres simulateurs (par ex. Simple-playground à gauche[4]) ainsi qu'un robot mobile physique (soit le robot Rosa au milieu, soit le robot Epuck à droite).

Contexte

Depuis 2015, le cours "*System design integration and control (SDIC)*"[5] de l'Université Pompeu Fabra à Barcelone (conçu par Clément Moulin-Frier, co-encadrant du projet) propose à des étudiants de Master d'étudier des mécanismes de contrôle sensorimoteur et d'interaction multi-agent par une initiation pratique au contrôle réactif de robots mobiles. Les cursus de ces étudiants sont très variés : beaucoup sont informaticiens mais un certain nombre viennent de cursus de sciences sociales, de design ou d'art digital. Le cours doit donc être compréhensible par l'ensemble de ces profils.

La librairie pyvrep-epuck[6] a été développée dans le but de permettre à des étudiants aux profils variés de s'initier au contrôle robotique et à l'interaction multi-robot. Elle permet de programmer des boucles sensorimotrices bas niveau par des fonctions Python, de les combiner pour créer des comportements plus complexes, et de lancer ces boucles de contrôle sur une population de robots mobiles dans un simulateur. La définition des comportements est réalisée selon une approche "bottom-up" (du plus simple au plus complexe), dans l'esprit de la "behavior-based robotics" proposée par Brooks[7] ; et plus précisément le cadre formel des "véhicules de Braitenberg"[8].

La libraire est interfacée avec le simulateur 3D CoppeliaSim[9] et permet de contrôler des robots mobiles simulés Epucks. Ces robots sont constitués d'une base circulaire, de deux roues, de

---

[4] Voir une version animée de cette image sur https://github.com/mgarciaortiz/simple-playgrounds/
[5] https://www.upf.edu/web/csim/syllabus#:~:text=System%20design%20integration%20and%20control%20-%2030863
[6] https://github.com/clement-moulin-frier/pyvrep_epuck
[7] Brooks R. A. (1991), *Intelligence without representation*, Artificial Intelligence, vol. 47, no. 1-3
[8] Braitenberg, V. (1984). *Vehicles: Experiments in synthetic psychology*. Cambridge, MA: MIT Press
[9] https://www.coppeliarobotics.com/

capteurs de proximités, de capteurs au sols, ainsi que d'une caméra (voir Fig. 2 à droite pour une description du robot physique).

Les sessions pratiques du cours sont réalisées dans des Jupyter Notebooks[10], qui contiennent à la fois les instructions de la session ainsi que du code Python. La librairie permet de se connecter au simulateur puis de programmer des comportements de façon très intuitive et interactive (voir Fig. 1). Par exemple, exécuter une cellule du notebook contenant le code `robot.left_speed = 1` a un effet immédiat dans le simulateur : la roue gauche va se mettre en rotation et le robot va donc tourner sur lui-même.

Projet

Si le simulateur CoppeliaSim offre de nombreuses possibilités intéressantes (par ex. la possibilité de concevoir des scènes en 3D assez riches), il a également certains inconvénients. En particulier, il est assez gourmand en ressources, on ne peut ainsi simuler que 3 ou 4 robots interagissant dans la même scène (sur une machine standard). Ceci est problématique car l'un des objectifs du cours est justement d'étudier des interactions entre différents robots, voire entre différents groupes de robots. Plus généralement, le fait que la librairie soit pour l'instant spécifique au contrôle de robots Epuck dans CoppeliaSim est également problématique. On aimerait pouvoir l'utiliser dans différents environnements, que ce soit des simulateurs 2D ou des robots physiques (voir Fig. 2).

Du côté des simulateurs, nous sommes particulièrement intéressés par le simulateur simple-playgrounds[11] (ex-Flatland[12]), dont le développement a commencé à l'ENSTA-Paristech et qui se poursuit actuellement par Michael Gracia Ortiz de la City University à Londres (qui co-encadrera ce projet). Il s'agit d'un simulateur rapide en 2D à la physique réaliste, permettant d'intégrer de nombreux robots (jusqu'à une quinzaine sur une machine standard) ainsi que des objets à la dynamique variée (voir Fig. 2, à gauche).

Du côté des robots réels, la décision reste à prendre mais les deux candidats sont le robot ROSA[13] développé par Pollen Robotics à Bordeaux (Fig. 2, au centre), ou le robot Epuck[14] développé par l'EPFL en Suisse (Fig. 2, à droite).

Les objectifs du projet sont les suivants :

- Familiarisation avec les principes de la "behavior-based robotics" par la lecture de quelques articles scientifiques (voir les notes de bas de page ci-dessus)

[10] https://jupyter.org/
[11] https://github.com/mgarciaortiz/simple-playgrounds/
[12] https://arxiv.org/abs/1809.00510
[13] https://www.youtube.com/watch?v=BjSzU6EHspg
[14] https://www.gctronic.com/doc/index.php/e-puck2

- Prise en main de la librairie pyvrep-epuck et des jupyter notebooks associés[15]
- Prise en main de l'environnement de simulation simple-playgrounds[16]
- Prise en main du robot physique (soit Rosa, soit Epuck)
- Abstraction des structures de données de la librairie pour pouvoir l'interfacer avec différents simulateurs ou robots physiques
- Intégration de simple-playgrounds dans pyvrep-epuck
- Intégration d'un robot physique dans pyvrep-epuck
- Adaptation des jupyter-notebook utilisés dans le cours (VREP -> simple-playgrounds)

Les pré-requis pour le groupe d'étudiants sont les suivants :

- Compétences solides en programmation orientée-objet, notamment en Python
- Lecture et écriture en anglais (les jupyter notebooks sont en anglais). L'anglais parlé est recommandé.
- Une expérience dans l'utilisation de robots physiques est recommandée, bien que pas forcément nécessaire
- Intérêt pour les systèmes autonomes et l'interaction multi-agent, en intelligence artificielle et en biologie

Note : les étudiants étant à la recherche d'un stage de Master et fortement intéressés par l'émergence de comportements dans les systèmes multi-agents sont invités à prendre contact avec Clément Moulin-Frier (voir email en début de document). Ces stages s'inscriront dans les cadres conceptuels décrits dans ces deux articles :

- Grounding Artificial Intelligence in the Origins of Human Behavior https://arxiv.org/abs/2012.08564
- Multi-Agent Reinforcement Learning as a Computational Tool for Language Evolution Research: Historical Context and Future Challenges https://arxiv.org/abs/2002.08878

[15] Toutes les informations nécessaires à l'installation sont disponibles sur https://github.com/clement-moulin-frier/pyvrep_epuck
[16] Toutes les informations nécessaires à l'installation sont disponibles sur https://github.com/mgarciaortiz/simple-playgrounds/