

Une arène bas coût pour l'enseignement de la robotique

Alain Godon, Rémy Guyonneau*, Franck Mercier
remy.guyonneau@univ-angers.fr
Adresse : 62 Avenue Notre Dame du Lac, 49 000 Angers

RÉSUMÉ : Ce papier présente une architecture appelée *BotArena* permettant la réalisation d'une plateforme expérimentale pour la robotique en essaim. Cette plateforme répond à trois contraintes : le coût, elle doit être abordable pour les structures d'enseignement ; la modularité, elle doit être déplaçable et doit pouvoir facilement s'adapter à la place disponible ; et la versatilité, elle doit permettre de traiter une multitude de problèmes en robotique mobile. L'objectif de cette architecture open-source et open-hardware est de servir d'outil pédagogique et de vulgarisation.

Mots clés : robotique en essaim, plateforme expérimentale.

1 INTRODUCTION

La robotique utilisée dans un cadre pédagogique apporte plusieurs bénéfices. Entre autres, elle facilite la compréhension de concepts abstraits [1], elle fournit un environnement de travail collaboratif [2] ou encore elle offre l'opportunité de travailler et d'explorer des solutions à des problèmes concrets [3].

La robotique en essaim, initialement inspirée par les comportements d'animaux sociaux, est une branche de la robotique qui s'intéresse au développement de comportements *complexes* à l'aide de robots *simples*. Bien qu'introduit en 1988 [4, 5], c'est un champ de recherche qui a été particulièrement actif ces 20 dernières années [6]. Pour l'instant, les applications de la robotique en essaim restent restreintes aux périmètres des laboratoires de recherche et les applications peinent à arriver dans les milieux professionnels et grand public. Il persiste encore quelques verrous [6] pour que la technologie soit applicable en conditions réelles. Malgré tout, les ambitions de cette approche et les avancées de la recherche suggèrent que les applications de la robotique en essaim sortiront bientôt du cadre du laboratoire pour devenir courantes.

Dans ce contexte, il semble intéressant de pouvoir proposer aux étudiants de travailler sur ces notions de robotique en essaim : bénéficier des avantages pédagogiques de la robotique tout en abordant des notions d'avenir. Cependant, un des freins au développement de la robotique en essaim est le coût que représente la mise en place d'une plateforme expérimentale, ce qui implique aussi la possibilité d'utiliser de tels systèmes dans un cadre pédagogique. Ce papier présente donc une architecture, appelée *BotArena*, inspirée des approches *hardware in the loop* [7, 8] permettant de satisfaire les trois contraintes suivantes :

- Le coût : le système doit être abordable pour les structures d'enseignement. Il existe en effet déjà des arènes de robotique en essaim. Le *Robotarium* [9] par exemple a coûté plus de 2 millions de dollars et n'est disponible que pour les chercheurs, donc pas adapté comme outil pédagogique ;
- La modularité : l'objectif ici est de pouvoir facilement déplacer le système ainsi qu'en mo-

difier sa configuration (nombre de robots, espace de travail, etc) ;

- La versatilité : il faut que le système permette de s'intéresser à une multitude de problématiques abordées en robotique mobile comme la localisation, la planification de trajectoire, la cartographie, les systèmes multi-agents, etc.

La figure 1 présente le principe adopté pour la *BotArena* afin de satisfaire les contraintes listées précédemment : plusieurs robots se déplacent dans une arène. Ils sont localisés par des caméras qui calculent leurs positions dans un référentiel commun. Un superviseur centralise les positions de tous les robots, simule des données capteurs et calcule la commande à appliquer aux moteurs de chacun des robots.

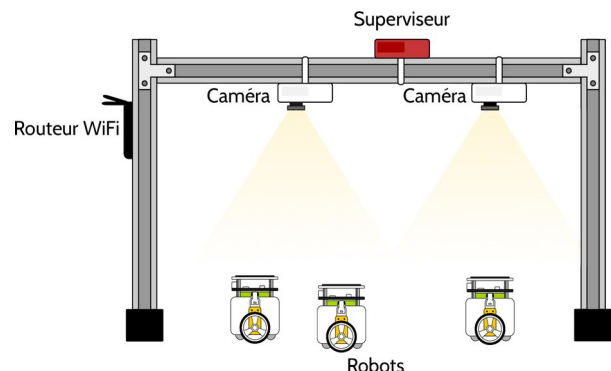


Figure 1: Principe de la BotArena

L'architecture logicielle de la *BotArena* est quant à elle présentée sur la figure 2.

La suite de l'article détaille les trois éléments de cette architecture : les robots, les caméras connectées et le superviseur. Enfin, la conclusion présente en détail l'utilisation actuelle, les tâches restantes à accomplir et les limites identifiées de cette approche.

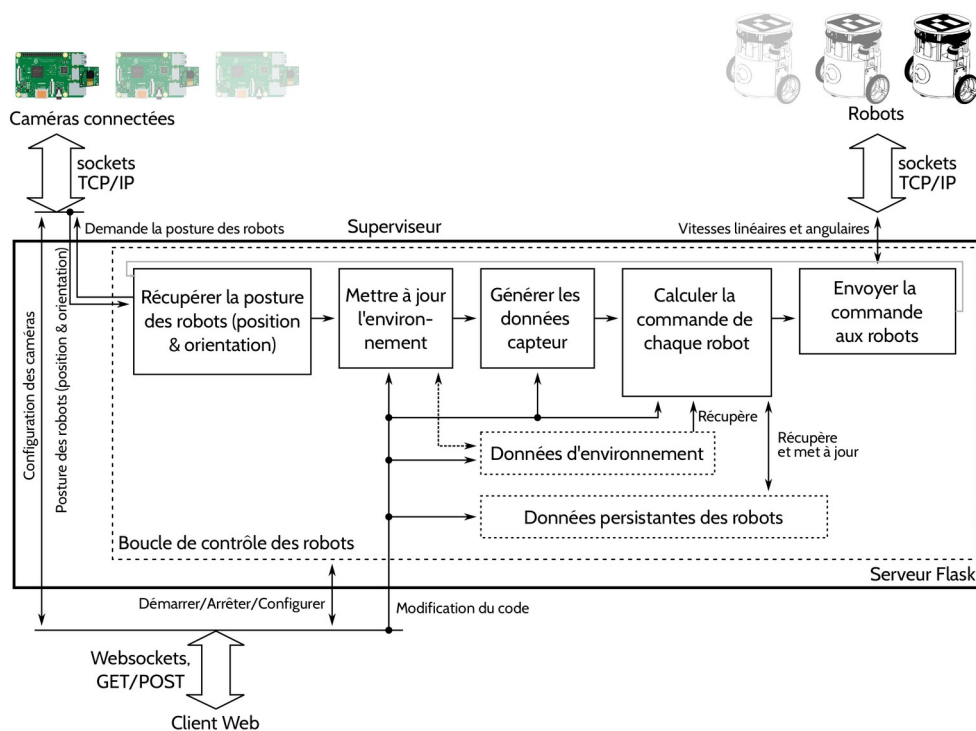


Figure 2: Architecture logicielle de la BotArena

2 LES ROBOTS

Afin de pouvoir adapter les robots aux besoins de l'arène, il a été décidé de réaliser un nouveau robot au lieu d'utiliser un robot du commerce. Ce nouveau robot répond aux besoins suivants :

- Faible coût : pour la réalisation d'un essaim, le prix unitaire d'un seul robot est un élément critique ;
- Petite taille : comme pour le point précédent, considérer une multitude de robots se déplaçant au même endroit génère des contraintes de place ;
- Architecture différentielle pour permettre un maximum de mobilité dans un minimum de place ;
- Communication WiFi pour la communication entre les robots et le superviseur ;
- Positionnement d'un marqueur au sommet du robot afin de pouvoir être détecté par les caméras.

La partie contrôle du robot est basée sur un microcontrôleur ESP32. Ce dernier a plusieurs avantages : il est peu cher, il intègre directement le WiFi (et du Bluetooth si besoin), et, couplé à un programmeur FT232, il permet de programmer directement le robot depuis l'interface Arduino, déjà populaire en pédagogie.

La partie actionneur est composée de deux moteurs à courant continu équipés de codeurs afin de pouvoir réguler la vitesse.

La figure 3 présente une vue éclatée du robot développé pour l'arène.

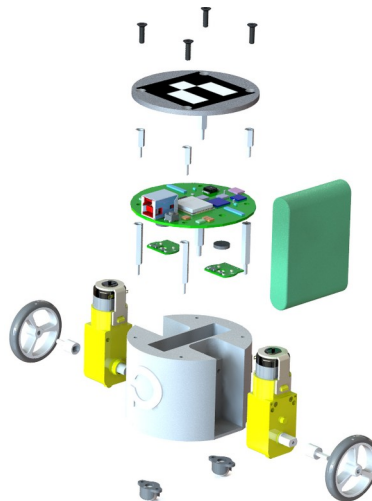


Figure 3: Le robot réalisé pour la BotArena

Un dépôt Gitlab¹ est accessible, contenant toutes les informations relatives au robot (routage de la carte électronique, modèles 3D des parties mécaniques, nomenclature...). Le code source du BIOS du robot est aussi disponible².

1 https://gitlab.u-angers.fr/botarena/ba_conception

2 https://gitlab.u-angers.fr/botarena/ba_bios

3 LES CAMÉRAS

Afin de satisfaire les contraintes de coût et de modularité, il a été décidé de réaliser un système multi-caméras basés sur des caméras peu chères. Ainsi, ajouter ou supprimer une caméra permet de facilement modifier la zone de travail considérée pour l'arène. Ces caméras connectées sont basées sur des cartes Raspberry Pi 4 équipées de modules PiCamV2, le tout installé dans un boîtier fabriqué en impression 3D. Elles sont placées en hauteur afin de pouvoir détecter les robots, plus spécifiquement les marqueurs installés sur le dessus des robots.

Pour la partie traitement d'image, tout est basé sur *OpenCV*. Les marqueurs utilisés sont des marqueurs *ArUco* [10] ; ils sont utilisés pour localiser et identifier les robots. Ces derniers ont l'avantage d'être faciles à mettre en place et sont nativement utilisables avec *OpenCV*. Les caméras sont aussi équipées d'un serveur TCP/IP : sur une demande, les positions et orientations calculées des robots sont envoyées au format JSON.

L'utilisation des caméras passe par trois étapes. La première est le calcul de la matrice intrinsèque. Ce calcul correspond à la calibration de la caméra, il n'est à faire qu'une seule fois (il dépend uniquement de l'optique de la caméra). Il faut ensuite calculer la matrice extrinsèque. Cette dernière dépend de la position de la caméra par rapport à un *repère monde*. Cette étape est à faire à chaque fois que la caméra est déplacée par rapport au repère de référence. Pour calculer cette matrice une mire est utilisée (Figure 4) : sur cette mire la position des marqueurs dans le repère monde est connu. Finalement, une fois les deux matrices calculées, il est possible de calculer la position et l'orientation de tous les marqueurs *ArUco* (et donc des robots) dans le repère monde.

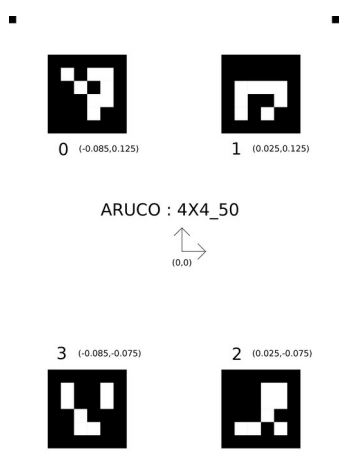


Figure 4: Exemple de mire pour le calcul de la matrice extrinsèque

Un dépôt Gitlab³ est accessible avec le code source du serveur TCP/IP ainsi qu'une documentation sur les maths pour le calcul des matrices et les images des marqueurs utilisés.

4 LE SUPERVISEUR

Le superviseur est l'élément central de la BotArena, il permet :

- L'administration des caméras ;
- La configuration de l'environnement ;
- La simulation des capteurs et l'implémentation du comportement des robots.

Basé sur un serveur Flask (serveur Web en Python), il est directement accessible depuis un navigateur client. Il propose différentes fonctionnalités explicitées ci-dessous.

Comme indiqué à la Section 3, les caméras nécessitent des calibrations avant de pouvoir être utilisées. Le superviseur permet de lancer directement ces calibrations sans avoir à se connecter manuellement sur chacune des caméras. Ce qui facilite le changement de la configuration de l'arène et donc de répondre à la contrainte de modularité.

Le superviseur permet de facilement ajouter ou supprimer un robot. Il permet aussi de définir quelles sont les données que partagent les robots concernant l'environnement (la présence d'obstacles, de points d'intérêts...). Cette approche permet de modifier « virtuellement » l'environnement et donc de s'intéresser à une multitude de configurations différentes.

Enfin le superviseur permet de simuler les capteurs et de programmer les robots. La simulation des capteurs se fait pour chaque robot en fonction de sa posture dans l'environnement. Pour les premières expérimentations, des capteurs LiDAR ainsi que des capteurs de phéromones ont été simulés. Cette simulation permet d'aborder une grande variété de problématiques, ce qui répond à la contrainte de versatilité. Concernant le code du robot, il n'est pas envoyé sur les robots. Le superviseur, en fonction du code saisi sur l'interface, des données capteurs et de l'environnement, calcule indépendamment les commandes à envoyer à chacun des robots. Ce mode de fonctionnement permet de rapidement pouvoir reprogrammer le comportement de tous les robots.

Pendant le fonctionnement de l'arène, le superviseur effectue les tâches suivantes (Figure 2) :

- Ouverture d'une socket TCP/IP pour chacune des caméras ;
- Ouverture d'une socket TCP/IP pour chacun des robots ;
- Boucle de fonctionnement :
 - Récupération de la position/orientation de chacun des robots (depuis les caméras) ;
 - Générations des données capteurs ;

3 https://gitlab.u-angers.fr/botarena/ba_picam

- Calcul de la commande pour chacun des robots en fonction des données capteurs mais aussi des données robots et de l'environnement ;
- Envoi de la commande aux robots ;

Le code source du superviseur est disponible sur un dépôt Gitlab⁴.

5 CONCLUSION

Dans cet article une architecture pour la réalisation d'une plateforme expérimentale pour la robotique en essaim a été présentée. Afin de pouvoir être exploitable pédagogiquement, elle répond à trois critères : un coût abordable pour les structures d'enseignement, une modularité sur la taille et le nombre de robots, une versatilité sur les problématiques abordables.

Deux configurations de cette arène ont déjà été présentées : dans le cadre des 6^e journées des Démonstrateurs en Automatique en juin 2022 (Figure 5) et lors de la fête de la science en octobre 2022 (Figure 6). Le système c'est montré facilement reconfigurable et adaptable en fonction des besoins. Ceci étant, il reste encore à tester les limites de l'architecture en termes de nombre de caméras et de robots. Pour l'instant les tests ont été réalisés avec 3 caméras et une dizaine de robots sans noter de ralentissements.



Figure 6: Fête de la science - Angers 2022

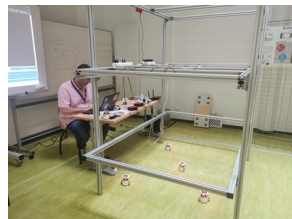


Figure 5: 6èmes Journées des Démonstrateurs en Automatique - Angers 2022

Cet outil dont la première version a été terminée pour les journées des démonstrateurs n'a pas encore été utilisé devant un public étudiant. Quatre axes d'utilisation sont considérés :

- L'utilisation des robots seuls : Étant construits autour d'un ESP32 il est envisagé d'utiliser ces plateformes pour les cours de microcontrôleur et comme illustration pour la mise en place d'une régulation (PID sur les moteurs). De plus, les robots pouvant directement être programmés avec l'environnement Arduino, ils peuvent être utilisés par des jeunes novices en programmation.
- L'utilisation des caméras : les caméras représentent une bonne illustration de techniques pouvant être utilisées en traitement d'images. Les calculs des matrices intrinsèque et extrin-

sèque sont une application directe de notions étudiées en vision industrielle par exemple.

- En robotique mobile : cette arène a pour vocation à être utilisée dans un module de robotique mobile. Cette plateforme sera employée pour mettre en œuvre des algorithmes de robotique en essaim, mais pas uniquement. La simulation des capteurs et de l'environnement offre la possibilité de travailler dans de nombreux contextes différents, allant de la sortie de labyrinthe à la cartographie.
- Amélioration du système : cet outil encore en développement est un terrain de jeu intéressant pour les projets étudiants. Parmi les prochains projets il y aura la réalisation d'un simulateur pour pouvoir développer les algorithmes sans avoir à utiliser l'arène (pour le prototypage par exemple) ainsi que le développement d'une interface de programmation simplifiée pour le superviseur afin de pouvoir l'utiliser lors des événements de vulgarisation comme la fête de la science.

Tout le projet est en open-source et open-hardware.

Bibliographie

- [1] Eguchi, Amy. "Educational robotics theories and practice: Tips for how to do it right." *Robots in K-12 education: A new technology for learning*. IGI Global, 2012. 1-30.
- [2] Eguchi, Amy, and Lisbeth Uribe. "Is Educational Robotics for Everyone?-A case study of a 4th grade educational robotics unit." *Society for information technology & teacher education international conference*. Association for the Advancement of Computing in Education (AACE), 2012.
- [3] Miller, David P., and Illah Nourbakhsh. "Robotics for education." *Springer handbook of robotics*. Cham: Springer International Publishing, 2016. 2115-2134.
- [4] Beni, Gerardo. "The concept of cellular robotic system." *Proceedings 1988 IEEE International Symposium on Intelligent Control*. IEEE Computer Society, 1988.
- [5] Fukuda, Toshio, and Seiya Nakagawa. "Approach to the dynamically reconfigurable robotic system." *Journal of Intelligent and Robotic Systems 1 (1988)*: 55-72.
- [6] Dorigo, Marco, Guy Theraulaz, and Vito Trianni. "Swarm robotics: Past, present, and future [point of view]." *Proceedings of the IEEE 109.7 (2021)*: 1152-1165.
- [7] Bullock, Darcy, et al. "Hardware-in-the-loop simulation." *Transportation Research Part C: Emerging Technologies 12.1 (2004)*: 73-89.
- [8] Isma, Akli, Abdous Kamel, and Sebaa Abderrahmane. "Hardware in The Loop Simulation for robot Navigation with RFID." *2022 7th International Conference on Image and Signal Processing and their Applications (ISPA)*. IEEE, 2022.
- [9] Wilson, Sean, et al. "The robotarium: Globally impactful opportunities, challenges, and lessons learned in remote-access, distributed control of multirobot systems." *IEEE Control Systems Magazine 40.1 (2020)*: 26-44.
- [10] Garrido-Jurado, Sergio, et al. "Automatic generation and detection of highly reliable fiducial markers under occlusion." *Pattern Recognition 47.6 (2014)*: 2280-2292.

4 https://gitlab.u-angers.fr/botarena/ba_superviser