

Navigation autonome d'un robot Pepper pour la compétition internationale RoboCup@Home 2020 - Bordeaux

Stage Projet de Fin d'Études / Ingénieur

Mots Clés : Navigation, Cartographie, ROS, Python, C++

Rémunération et durée :

525 euros/mois net, 6 mois à partir de février.

Encadrants

Jacques SARAYDARYAN, Benoit RENAULT, Olivier SIMONIN

Contact : jacques.saraydaryan@cpe.fr, benoit.renault@insa-lyon.fr,
olivier.simonin@insa-lyon.fr

Lieu du stage : Laboratoire CITI / INSA Lyon, CPE Lyon

Équipe de recherche : Chroma (<https://team.inria.fr/chroma/en/>)

Contexte du stage

Afin d'évaluer les différentes avancées en recherche et développement robotique dans des environnements réalistes, plusieurs compétitions internationales ont vu le jour. La Robocup est l'une des plus connues et réputées dans le domaine.

Un regroupement de chercheurs et d'ingénieurs de l'INSA de Lyon, CPE Lyon, Université Lyon 1, et du laboratoire CITI (équipe Chroma) ont créé une équipe, LyonTech¹, pour participer à cette compétition dans les ligues Robocup@Home² SSPL (Social Standard Platform League - Pepper) et OPL (Open Platform League - Robot custom).

Les bonnes performances réalisées lors de la compétition de 2018 (5ème place) et 2019 (3ème place) nous ont permis d'être qualifiés pour l'édition 2020³. Dans ce contexte, de nombreuses briques logicielles ont été développées pour le contrôle, la perception et la navigation du robot.

Dans le cadre du PFE proposé, le stagiaire aura à créer des stratégies de navigation spécifiques à des situations particulières, et étendre la stack de navigation standard de ROS ainsi que les surcouches développées par l'équipe avec la prise en compte des capteurs de type sonars, et le framework RTABMAP.

¹ Site web de l'équipe : <https://robocup-lyontech.github.io/>

² Site web de la compétition : <http://www.robocupathome.org/>

³ Vidéo de qualification 2020 : <https://youtu.be/6DnTP3k-580>

Objectif 1 : Stratégies de navigation

Aujourd'hui, nous utilisons la stack de navigation standard de ROS, finement paramétrée pour nos robots, avec une surcouche maison ajoutant le concept de points d'intérêts et la possibilité d'adopter différentes stratégies de navigation.

Néanmoins, une seule stratégie a été mise en place et testée, qui est celle d'aller vers un point d'intérêt en utilisant seulement la stack de navigation de ROS standard avec le Navfn Global Planner⁴ et le DWA Local Planner⁵. Les précédentes compétitions ont montré un besoin de stratégies autres pour la gestion de cas particuliers comme :

- passage de porte en roulant de côté (permet de transporter un objet sous le bras du robot),
- rotation simple et rapide sur lui-même (l'utilisation de DWA peut engendrer des comportements imprévisibles)
- retour arrière en cas de blocage à l'avant,
- suivi de personne, ...

Il s'agira pour le stagiaire de développer, intégrer et tester de nouvelles stratégies afin d'améliorer les performances des robots en compétition.

Objectif 2 : Cartographie

Nous utilisons aujourd'hui comme système de SLAM (Simultaneous Localization And Mapping) la combinaison de costmap_2d⁶ + AMCL⁷ fournie par la stack de navigation standard de ROS. Elle a l'avantage d'être assez simple et efficace mais dans notre usage, souffre de quelques défauts :

- elle n'utilise pas les sonars pour "nettoyer" la carte, nous privant de précieuses informations sur l'état réel de l'environnement,
- elle n'enregistre qu'une information 2D sur une grille, ce qui, malgré son efficacité, est en deçà de l'état de l'art robotique actuel.

Aussi le stagiaire explorera la possibilité d'utiliser :

- d'une part, les données des sonars pour mettre-à-jour la carte,
- le framework SLAM 3D RTABMAP⁸ pour améliorer les connaissances du robot exploitables, ainsi que le framework d'enregistrement de carte OCTOMAP⁹.

Organisation

Le stage se déroulera dans l'équipe CHROMA (INSA/INRIA) du laboratoire CITI. Vous serez intégré à l'équipe LyonTech et vous contribuerez à la mise en oeuvre et à l'amélioration des différentes briques existantes :

- Perception : détection et enregistrement d'obstacles avec RTABMAP + OCTOMAP, et mise à jour de la costmap en prenant en compte les données des sonars,

⁴ Page de documentation officielle de NavFN : <https://wiki.ros.org/navfn>

⁵ Page de documentation officielle de DWA: https://wiki.ros.org/dwa_local_planner

⁶ Page de documentation officielle de costmap_2d : https://wiki.ros.org/costmap_2d

⁷ Page de documentation officielle d'AMCL : <https://wiki.ros.org/amcl>

⁸ Site officiel : <http://introlab.github.io/rtabmap/> , doc officielle ROS : https://wiki.ros.org/rtabmap_ros

⁹ Site officiel : <https://octomap.github.io/>

- Navigation : navigation en milieu dynamique et incertain (global et local planners, costmaps, stratégies de navigation, ...), localisation du robot.

Après vous être familiarisé avec l'éco-système (robot Pepper, framework ROS, framework NaoQi), vous devrez refaire les épreuves pertinentes de la robocup de 2019 avec les solutions existantes. Une fois cette mise en oeuvre réalisée vous participerez au développement / à l'amélioration des briques logicielles sur la perception, la navigation et la décision.

Une phase de durcissement du code et de la procédure sera à prévoir avant la compétition. Elle sera fondée sur un ensemble d'expérimentations dans un environnement test de type pseudo-appartement dont nous disposons.

Vous irez avec le reste de l'équipe à la Robocup 2020 à Bordeaux (22-28 juin 2020) afin de mettre à l'épreuve en personne vos contributions !

Profil recherché

- Curieux, persévérant et autonome
- A la recherche d'une expérience de pointe dans le domaine de la robotique
- Appétence pour la programmation et l'architecture d'applications
- Connaissance et mise en oeuvre des langages Python et C++
- Connaissance de l'écosystème linux (notamment à l'aise avec bash et git)

Des connaissances de ROS (Robot Operating System), Framework NaoQi ou Docker seraient un plus.

Annexes



Photo de groupe des participants à la compétition @Home Sydney 2019



L'équipe en pleine compétition à Montréal en 2018

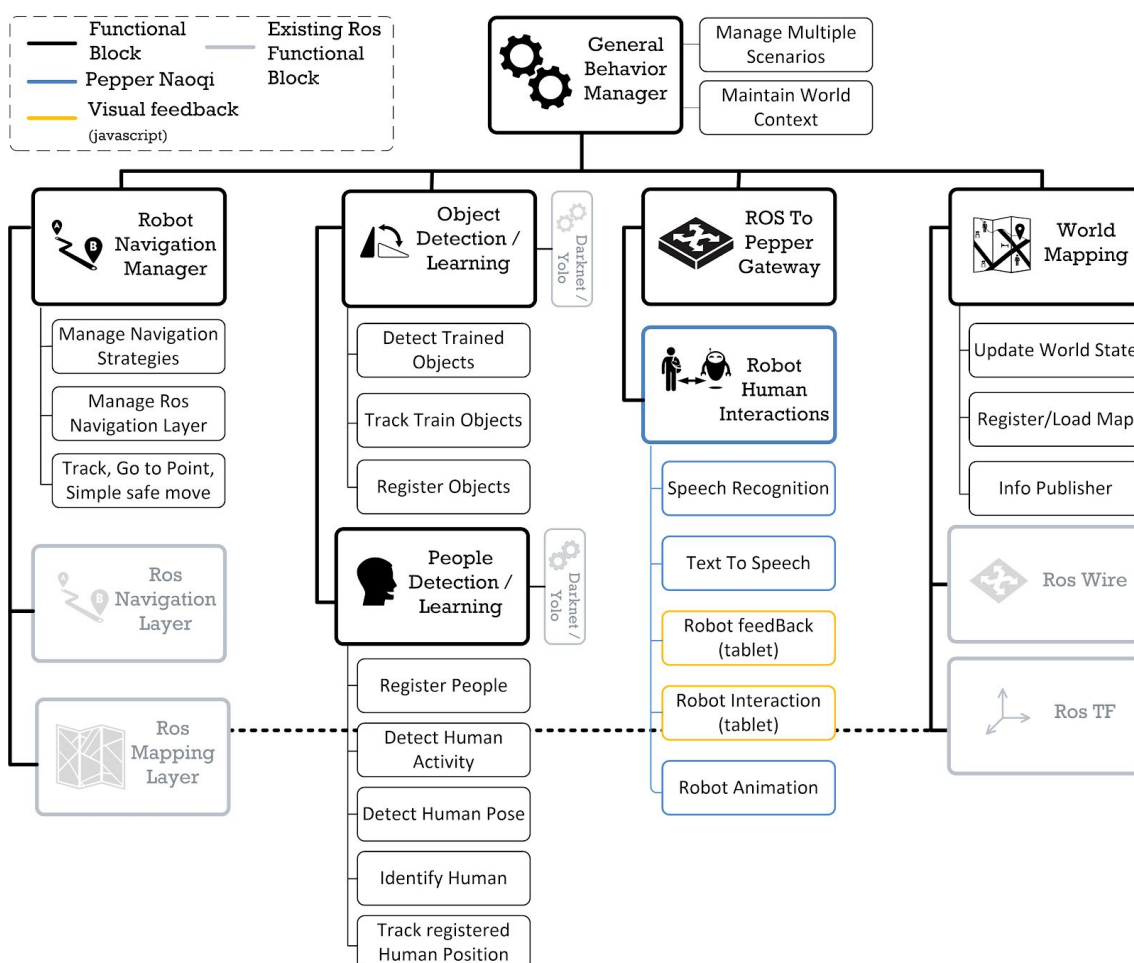


Schéma de notre architecture logicielle 2019 sur Pepper