

Objectifs généraux

Rain of Music se déroule depuis plusieurs années entre le SCRIME¹, l'Université de Bordeaux, l'ESTIA, et l'Université des Pays-Basques à Bilbao.

Il s'agit un projet arts-science regroupant les problématiques de la robotique, des drones, de la reconnaissance gestuelle et du temps-réel dans les médias interactifs. L'équipe est constitué d'ingénieurs aux spécialités variées, ainsi que d'artistes compositeurs et improvisateurs. Le projet se présente sous la forme d'un opéra où la technologie réagit au spectacle vivant, produisant une chorégraphie de drones et de Metabots sous l'influence de la performance musicale.

Cette chorégraphie plus ou moins autonome naît de traitements en temps-réel guidés par un scénario interactif écrit dans le séquenceur libre *ossia score*² développé au SCRIME.

Les technologies employées se basent en grande partie sur des systèmes d'apprentissage, pour la reconnaissance des gestes du chef d'orchestre et la génération de mouvement des robots. Le traitement d'image tient aussi une part importante, notamment pour la gestion des trajectoires des drones et la prévention de collisions.

Petrichord est un DIMR (démonstrateur interactif de médiation robotique) prévu pour installation lors de la Robocup 2020, qui aura lieu en juin prochain à Bordeaux.

Ce DIMR permet à un enfant visiteur de l'exposition de chorégrapier à l'aide de gestes un opéra mêlant des sons, des drones, des robots.

Le développement de ce système sera réalisé sur la plateforme logicielle *ossia score*, écrite en C++ avec Qt, et bénéficiera du concours de la société Celtera.

Projet 1: évitement de collisions en temps réel par apprentissage automatique

Objectif : Développer un contrôleur de robots qui possède en entrée les positions et trajectoires actuelles des robots et drones, et décide d'une action en cas de risque de collision, qui en priorité continuera le mouvement entamé dans une autre direction, ou bien mettra le robot ou drone en sécurité.

Niveau conseillé : Bac+5

Estimation : 20 jours-homme

Compétences requises : C++, robotique, apprentissage automatique

Références

Two-layer obstacle collision avoidance with machine learning for more energy-efficient unmanned aircraft trajectories
Choi, Jimenez, Mavris

Deep Learning for Autonomous Collision Avoidance
Strömberg

Collision Avoidance for Indoor Service Robots through Multimodal Deep Reinforcement Learning
Leiva, Tsunekawa, Ruiz-del-Solar

¹<https://scrimelabri.fr>

²<https://ossia.io>

Projet 2 : protocole de communication bas-niveau pour drone

Objectifs : Implémenter un protocole dans ossia score permettant d'envoyer des contrôles aux drones, en utilisant le SDK Parrot. Une fois cela fait, optimiser la consommation d'énergie : utiliser la connaissance des commandes dans score pour altérer les trajectoires de manière à maximiser le temps de vol de chaque drone.

Niveau conseillé : Bac+5

Estimation : 20 jours-homme

Compétences requises : C++, robotique

Références

SDK des drones Parrot :

<https://github.com/Parrot-Developers/arsdk-ng>

Projet 3 : trajectoires créatives pour drones

Objectif : on cherche à améliorer la manière dont les trajectoires des drones et robots sont écrites. Actuellement, le logiciel de scénarisation permet d'écrire des splines en 2D, ce qui est suffisant pour les Metabots³, qui sont des robots quadrupèdes mais pas pour les drones.

On cherche à introduire deux avancées :

- Une bibliothèque de trajectoires autres que les splines :
 - Cercles, rectangles, et autres formes géométriques élémentaires
 - Spirales
 - Scannages
 - D'autres idées de trajectoires sont possibles, à la discrétion des étudiants - on pourra par exemple s'inspirer de Zirkonium ou bien de IanniX, voire générer des trajectoires par diverses méthodes d'apprentissage.
- Un passage à la 3D, afin d'écrire des trajectoires que les drones suivront en temps-réel.
 - Au minimum permettre de positionner les formes géométriques simples et en réaliser rotation, scale, translation.
 - Selon le temps disponible, implémenter une spline 3D modifiable.

Le logiciel est écrit en C++ avec la bibliothèque Qt. La partie 3D pourra utiliser :

- soit Qt 3D qui est un moteur 3D de haut-niveau fourni avec Qt.
- soit OpenGL directement.

Niveau conseillé : Bac+4

Estimation : 12 jours-homme

Compétences requises : C++, Qt, 3D

³<http://www.metabot.fr/>