CentraleSupélec

Etude de cas

pour démarrer

CentraleSupélec

ONERA
THE FRENCH AEROSPACE LAB

RUTGERS

université
PARIS-SACLAY

Etude de cas

Localisation et caractérisation d'une source de polluant

- Une flotte de robots mobile doit intervenir sur un site d'où s'échappe un polluant et doit en déterminer la source, ainsi que la forme du « nuage » qui s'en est échappé (supposé statique). Pour ce faire, chaque robot est muni d'un capteur lui donnant une mesure du niveau/concentration de pollution à l'endroit où il se trouve.

- **Objectifs de l'étude de cas :** proposer et coder des algorithmes de commande permettant de contrôler le déplacement d'une flotte de *N* robots pour :
  - **Mission 1** : localiser la source du polluant (max) et en relever le niveau de pollution,
  - **Mission 2 (facultative)** : déterminer la forme du nuage de polluant.

- Plusieurs scénarios de difficulté croissante sont à considérer (cf. diapos suivantes).

- Plusieurs solutions peuvent être proposées et testées (on essaiera dans la mesure du possible de minimiser la distance totale parcourue par les robots lors de la mission).

Etude de cas

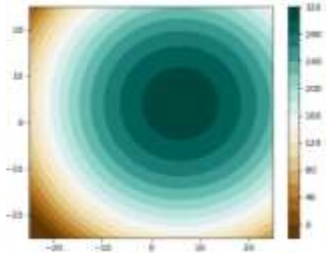
- Une librairie Python vous est fournie pour la génération du nuage de polluant (« potentiel ») et la simulation des trajectoires des robots.
- Le script principal à lancer est : `etude_de_cas.py`
- Vous pouvez utiliser une dynamique simple intégrateur pour simuler le mouvement des robots
- Vous pouvez modifier le nombre de robots
  - Vous pouvez choisir de travailler avec un nombre de robots *N*=3, 4 ou 5, ou bien de proposer un algorithme fonctionnant avec n'importe laquelle de ces trois valeurs possibles.
- Vous pouvez modifier les positions initiales des robots
  - L'algorithme proposé doit pouvoir fonctionner à partir de positions initiales tirées aléatoirement.

- La définition de la fonction potentiel utilisée pour simuler le nuage de polluant est faite dans le fichier `control_algo_potential.py`
- La loi de commande à évaluer par chacun des robots est également à définit dans le fichier `control_algo_potential.py`

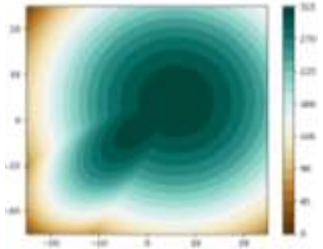
Etude de cas

- Dans le fichier `control_algo_potential.py` vous pouvez définit la difficulté du scénario et sa génération aléatoire (ou non)
- Trois scénarios de difficulté croissante peuvent être traitées (`difficulty` = 1, 2 ou 3).
- Pour le développement et la mise au point des algorithmes, travailler avec un nuage identique à chaque lancement de simulation (`random` = False), puis valider ensuite sur des scénarios aléatoires.

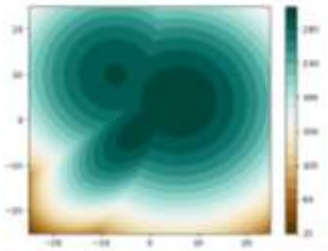
(difficulty=1, random=False)



(difficulty=2, random=False)



(difficulty=3, random=False)



Exemples de scénarios générés pour trois niveaux de difficulté

Exemples de scénarios générés aléatoirement pour le niveau de difficulté 3

(difficulty=3, random=True)