

Robotique et comportements réactifs

Véhicule de Braitenberg, Architecture de subsomption, Boïds

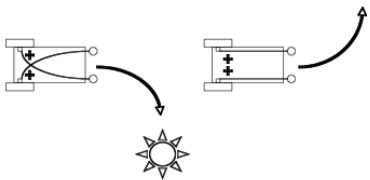
Mise à jour : Mars 2018

Rendez-vous sur la page <http://pages.isir.upmc.fr/~bredeche/> - onglet 3i025 pour télécharger le code source Python nécessaire pour faire les exercices ci-dessous (dépendances: Pygame, Matplotlib). Dans l'archive que vous récupérerez, seul le fichier *multirobots.py* vous intéresse. Pour toutes les questions qui suivent, vous devrez faire une copie de ce fichier avant de commencer. Il existe une aide sommaire dans les commentaires au début du fichier, et le fichier contient tout ce qu'il faut pour vous aider à utiliser les fonctions utiles à la réalisation des questions ci-dessous. Passez un peu de temps pour vous familiariser avec la fonction `step(.)`.

Pour l'évaluation, vous devez conserver un fichier prêt à l'emploi pour chaque question de chaque exercice.

Véhicules de Braitenberg

[Braitenberg, 1984]



L'image ci-contre montre deux exemples de véhicules de Braitenberg. Dans ces exemples, les connexions reliant les entrées sensorielles (ici: deux photorécepteurs) et les sorties motrices (ici: les deux moteurs) sont soit excitatrices, soit inhibitrices.

En s'inspirant de ces deux exemples, on vous demande de programmer quatre comportements:

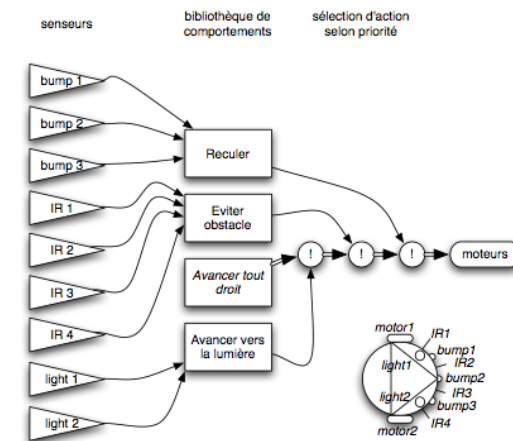
- (1) comportement "aller vers les murs et *ignorer* les robots", (fichier *braitenberg_loveWall.py*, à créer)
- (2) comportement "éviter les murs et *ignorer* les robots", (fichier *braitenberg_hateWall.py*, à créer)
- (3) comportement "aller vers les robots et *ignorer* les murs" (fichier *braitenberg_loveBot.py*, à créer)
- (4) comportement "éviter les autres robots et *ignorer* les murs" (fichier *braitenberg_hateBot.py*, à créer).
- (5) comportement "éviter les obstacles (murs et robots) (fichier *braitenberg_avoider.py*, à créer).

Par défaut (i.e. en l'absence d'obstacles), les comportements commandent un déplacement en ligne droite.

Testez avec 1 seul agent, puis avec 10 agents (variable *nbAgents*) afin de valider vos comportements.

Architecture de subsomption

[Brooks, 1986]



La figure ci-dessus donne un exemple d'architecture de subsomption, composée de comportements et d'un processus de sélection d'action qui précise l'ordre de priorité des comportements en fonction de leur activation possible.

Dans l'exercice précédent, les comportements (3) et (4) ne permettent pas de gérer l'évitement d'obstacle. Pour remédier à cela, créez un fichier *subsumption.py* afin d'implémenter pour chaque agent une architecture de Subsumption disposant de trois blocs de comportement.

Avant de commencer, modifier les comportements 2, 3 et 4 afin qu'en l'absence de stimulation sensorielle (i.e. le robot ne voit pas de murs ou d'autres robots), le robot n'avance pas. Implémentez aussi un nouveau comportement "aller tout droit" qui n'utilise pas les senseurs.

Stratégie 1: solo (comportement d'exploration)

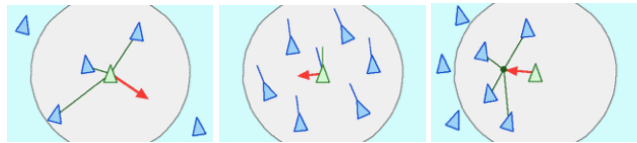
- (a) Aller tout droit
- (b) Eviter les obstacles (comportement no.5)

Stratégie 2: bot lover (comportement d'agrégation)

- (a) Aller tout droit (nouveau comportement)
- (b) Eviter les murs (comportement no.2)
- (c) Aller vers les robots (comportement no.3)

Vous devrez fixer la stratégie en terme de sélection d'action (i.e. l'ordre de priorité des comportements).

Testez votre programme en utilisant 1 robot, puis 10 robots, pour chaque stratégie. Puis mettez simultanément 4 robots qui suivent la stratégie 1 et un robot qui suit la stratégie 2.



Le comportement des "Boids" est contrôlé par trois règles: attraction, répulsion et orientation. Chaque règle s'applique en fonction de la distance aux voisins (distance au barycentre ou au plus proche -- au choix). Pour implémenter ces comportements, chaque agent doit pouvoir estimer la distance à ses voisins, ainsi que sa différence d'orientation.

Créez un fichier *boids.py*, et implémentez le comportement des boids pour 40 agents. Vous pouvez supprimer les murs. A vous de régler les conditions de déclenchement pour chaque règle.

Afin de gérer le cas particulier des murs, il est conseillé d'ajouter les comportements boids dans une architecture de subsomption dont le comportement de plus haute priorité est l'évitement de murs.