

Système d'inférence floue : application au contrôle d'un comportement réactif

1 Objectifs

Le but de ce TP est de se familiariser avec la commande floue et en particulier avec la construction de règles décrivant le comportement réactif d'un robot évoluant dans un environnement inconnu qu'il perçoit localement.

Un compte-rendu de 3 à 5 pages devra être fourni pour le A DEFINIR (dernier délai). Il contiendra, entre autre, une explication de la solution proposée, un compte rendu des différentes difficultés rencontrées et, en annexe, le contenu des fichiers de description des variables linguistiques et des règles ainsi que le chemin permettant d'accéder à ces fichiers.

2 Problème posé

Le problème auquel on s'intéresse est celui du pilotage d'un robot mobile dans un environnement inconnu. L'objectif du robot est de rejoindre un ensemble de buts placés aléatoirement tout en évitant les obstacles présents. Comme vous pourrez le constater, cette tâche, que l'on sait accomplir de manière naturelle, est en fait plus compliquée qu'il n'y paraît...

2.1 Description de l'environnement

L'environnement d'évolution du robot est un carré de 20 mètres de côté pouvant contenir des obstacles. L'environnement est décrit par un fichier texte contenant plusieurs lignes décrivant chacune un obstacle. Ces lignes sont de la forme :

longueur largeur posX posY [orientation]

Notons que le centre du carré de 20 mètres de côté se trouve en (0,0).

Vous trouverez quatre fichiers d'exemples : `map0.des`, `map1.des`, `map2.des` et `map3.des`. Ces quatre environnements seront utilisés pour évaluer les règles que vous rendrez. Vous pouvez néanmoins créer vos propres environnements, en vous inspirant de ceux existant, pour la mise au point de votre système. Les dimensions sont exprimées en centimètres.

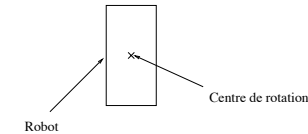


FIG. 1 – Description du robot

2.2 Description du robot

Le robot est un mobile de forme rectangulaire possédant un centre de rotation situé en son centre géométrique (voir figure 1). Ses dimensions sont de 60×40 cm.

Ce mobile est commandé grâce à l'envoi de consignes de vitesse linéaire et angulaire. Ces consignes sont automatiquement fournies au mobile 10 fois par seconde et proviennent de l'évaluation des règles floues que vous fournirez.

Les vitesses angulaires sont exprimées en degrés par seconde, les vitesses linéaires en centimètres par seconde. Comme tout véhicule réel, les changements de vitesses ne peuvent être réalisés instantanément. Ils sont effectués grâce à une accélération linéaire de 70 cm.s^{-2} et angulaire de 70 deg.s^{-2} . La vitesse maximale supportée par le robot est de 150 cm.s^{-1} en linéaire et 100 deg.s^{-1} en angulaire.

L'objectif du robot est de rejoindre un ensemble de buts placés aléatoirement tout en évitant les obstacles présents. Il dispose pour cela de capteurs télémétriques résumés figure 2. Les grandeurs mesurées sont exprimées en cm.

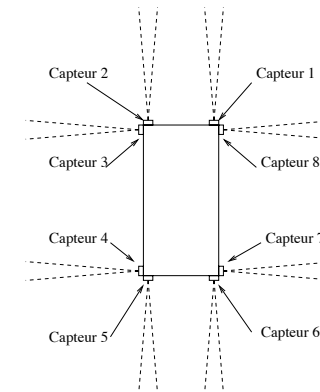


FIG. 2 – Position des capteurs physiques sur le contour du robot

L'accès aux différentes mesures est réalisé au travers de 4 capteurs logiques :

- **Capteur frontal** : mesure = $\min(\text{mesure}(\text{Capteur1}), \text{mesure}(\text{Capteur2}))$.
- **Capteur arrière** : mesure = $\min(\text{mesure}(\text{Capteur5}), \text{mesure}(\text{Capteur6}))$.
- **Capteur gauche** : mesure = $\min(\text{mesure}(\text{Capteur 3}), \text{mesure}(\text{Capteur 4}))$.
- **Capteur droit** : mesure = $\min(\text{mesure}(\text{Capteur 7}), \text{mesure}(\text{Capteur 8}))$.

L'utilisation de 4 capteurs logiques au lieu de 8 capteurs réels a pour but de simplifier l'écriture des règles, le nombre total de règles possibles étant exponentiel de la dimension de l'espace d'entrée (nombre de capteurs).

La position du robot par rapport au but est déterminée grâce à deux autres capteurs logiques :

- **Distance au but** : cette distance d est fournie en centimètres
- **Orientation du but** : ce capteur fournit l'angle θ entre le but et l'axe du robot (voir figure 3). Cette mesure est comprise entre -180 et 180 degrés.

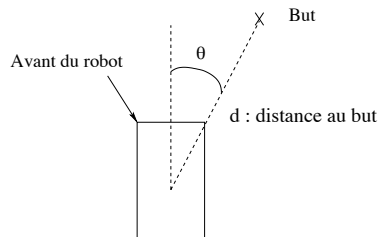


FIG. 3 – Mesure de la distance et de l'orientation au but

3 Variables floues associées au robot

Les différents capteurs et effecteurs décrits dans la section précédente sont connectés à des variables floues que vous pourrez donc utiliser dans vos règles :

– Capteurs :

InSlin : vitesse linéaire du robot.

InSang : vitesse angulaire.

DistGoal : distance au but.

DirecGoal : orientation vers le but.

ObstFront : distance à un obstacle frontal,

ObstBack : distance à un obstacle vers l'arrière.

ObstLeft : distance à un obstacle vers la gauche.

ObstRight : distance à un obstacle vers la droite.

– Effecteurs :

Slin : vitesse linéaire.

Sang : vitesse angulaire.

La syntaxe du langage utilisé pour l'écriture des règles est semblable à celle utilisée pour le TP précédent. Voici quelques exemples de règles :

si X est A alors Z est B

si X est pas A alors Y est B

si X est A alors Y est B et Z est C

si X est A ou Y est B alors Z est C

si X est A et Y est B alors Z est C

si X est A ou Y est B et Z est C alors T est D

4 Lancement de la simulation

La simulation doit être réalisée avec un environnement graphique permettant de voir le robot, ses capteurs, son évolution, etc. Pour la lancer il faut saisir la commande :

```
./robot [mapFile] fuzzyFile
```

5 Conseil de mise au point des règles

Afin de ne pas compliquer immédiatement le problème, il est fortement conseillé d'utiliser au préalable un environnement simple comportant peu ou pas d'obstacles. On cherchera à le complexifier au fur et à mesure.

De manière à faciliter la mise au point, il est possible de visualiser une partie de la surface de contrôle associée à vos règles avec l'outil `commande`.

```
commande [-v] configFile fuzzyFile
```

Les règles floues utilisées dans ce projet réalisent une fonction de $\mathbb{R}^8 \mapsto \mathbb{R}^2$. Afin de pouvoir visualiser facilement le résultat, il est nécessaire de réaliser une projection afin d'obtenir un sous-ensemble $\mathbb{R}^2 \mapsto \mathbb{R}$. Pour ce faire, on sélectionne :

- La variable de sortie que l'on désire étudier (Slin ou Sang).
- Deux variables d'entrée dont on souhaite voir varier les valeurs entre deux intervalles donnés.
- Les valeurs des autres variables d'entrée qui resteront fixe durant la projection.

La projection est réalisée grâce à l'utilitaire `commande`. `commande` admet comme paramètre d'entrée un fichier de configuration (indiquant les choix réalisés pour la projection) et un de description des règles.

Exemple : Fichier de configuration pour l'analyse de la vitesse angulaire demandée en fonction de la distance au but (entre 0 et 2m.) et l'orientation par rapport au but (entre -180° et 180°). Le paramètre **Resolution** indique le nombre de points de discrétisation pour chacun des deux intervalles.

Sortie	Sang
DistGoal	0 200
DirecGoal	-180 180
ObstBack	1000
ObstFront	1100
ObstLeft	1200
ObstRight	1300
InSlin	50
InSang	60
Resolution	50