

# Logique Floue

---

## Problème Posé

Au cœur de ce TP réside le défi de piloter un robot mobile à travers un espace confiné de 20 mètres de côté, semé d'obstacles divers. L'objectif du robot est de localiser et de rejoindre une cible disposée de manière aléatoire, tout en naviguant habilement pour éviter les obstacles présents. La complexité de cette tâche est exacerbée par le fait que le robot doit s'appuyer sur une perception locale et limitée de son environnement, n'ayant pour seules informations que la distance à l'obstacle le plus proche à gauche, droite, devant, et derrière, la distance et la direction de l'objectif, sa vitesse et son angle.

## Méthode Adoptée

Pour aborder ce problème complexe, une méthodologie incrémentale a été choisie, favorisant une approche étape par étape pour développer et affiner les capacités de navigation du robot. Cette approche méthodique permet une compréhension approfondie des principes de la logique floue appliquée à la robotique mobile et offre une opportunité de tester et d'ajuster les règles floues dans des scénarios (cartes) de complexité croissante.

### Étapes de Développement

- 1. Validation des Fonctionnalités de Base** : La première phase se concentre sur la confirmation des capacités motrices et de navigation de base du robot dans un environnement vierge. Cela comprend des tests de déplacement et d'orientation assurant ainsi que le robot est apte à entreprendre des tâches de navigation plus complexes.
- 2. Introduction et Gestion des Obstacles Simples** : Avec les capacités de base validées, le projet passe à l'introduction d'obstacles simples dans l'environnement (map1). Cette étape vise à développer et tester des mécanismes de perception et d'évitement d'obstacles, utilisant des règles floues pour naviguer efficacement autour des obstacles sans assistance.
- 3. Navigation dans des Environnements Complexes** : Finalement, le robot est mis à l'épreuve dans des scénarios de plus en plus complexes (map3), où il doit démontrer sa capacité à intégrer dynamiquement les informations sensorielles pour planifier et ajuster son trajet vers des cibles tout en naviguant autour d'une variété d'obstacles.

## Règle et valeurs linguistiques

Dans ce chapitre, nous explorons avec précision le système de règles basé sur la logique floue conçu pour piloter un robot mobile à travers un environnement non cartographié. Ce système s'appuie sur des variables linguistiques pour interpréter les signaux sensoriels et les états du robot en des termes intuitivement compréhensibles, permettant de définir des comportements complexes via un ensemble cohérent de règles floues.

### Définition des Variables et Valeurs Linguistiques

La construction de notre système de contrôle flou débute par l'établissement des variables linguistiques et de leurs valeurs associées. Ces variables traduisent les entrées — comme la distance et la direction par rapport à un objectif ou à un obstacle — et les sorties — telles que la vitesse de déplacement et l'orientation du robot — du système. Les variables linguistiques définies et leurs valeurs associées sont les suivantes :

- **Distance**

- **tropProche** : Signifie que l'objectif est dangereusement proche, défini par une rampe descendant de 20 à 75 cm.
- **tresProche** : Indique une proximité immédiate de l'objectif, avec une rampe descendant de 130 à 200 cm.
- **proche** : Décrit une distance modérée à l'objectif, représentée par une rampe descendant de 0 à 100 cm.
- **pas proche (loin)** : Représente une distance considérable de l'objectif, illustrée par une rampe montante de 0 à 100 cm.

- **Direction de l'objectif**

- **derriereADroite, aDroite, devant, aGauche, derriereAGauche** : Ces valeurs qualifient la position relative de l'objectif autour du robot, facilitant la prise de décision pour les manœuvres de rotation ou le maintien de la trajectoire.
- **coteDroite, coteGauche** : Ces valeurs qualifient de manière booléenne la direction de l'objectif, à gauche (peu ou très) ou à droite (peu ou très).

- **Vitesse linéaire (Slin) & Vitesse angulaire (Sang)**

- Ces variables déterminent respectivement la rapidité avec laquelle le robot avance ou recule et la vitesse à laquelle il pivote. Les valeurs varient de mouvements avant ou arrière à des ajustements directionnels précis, tels que **enMarcheAvant** ou **tourneADroite**.
- Pour la vitesse angulaire, en entrée, les valeurs **versDroite** et **versGauche** qualifient de manière booléenne la vitesse angulaire du robot, vers la gauche (peu ou très) ou vers la droite (peu ou très).

## Règles de Logique Floue

Le cœur de notre dispositif de contrôle repose sur une série de règles floues élaborées pour décrire la manière dont le robot doit réagir face à diverses configurations environnementales. Ces règles exploitent les valeurs linguistiques attribuées aux variables d'entrée pour influencer les comportements en sortie du système. Exemples de règles pertinentes :

- **Contrôle de la Vitesse (Slin) (Règles 1 à 5) :**

- En absence d'obstacle nécessitant un freinage (**ObstFront est loin**), le robot poursuit sa route (**Slin est enMarcheAvant**), promouvant une avancée fluide vers l'objectif.
- Face à un obstacle imposant un arrêt (**ObstFront est tresProche**) alors que le robot est en mouvement (**InSlin est enMarcheAvant**), une inversion de la marche est exécutée (**Slin est enMarcheArriere**) pour éviter une collision frontale.
- Lorsque robot s'arrête lors d'un équilibre entre la marche avant (**enMarcheAvant**) et la marche arrière (**enMarcheArriere**).

- **Ajustement Directionnel (Sang) (Règles 6 à 10) :**
  - Le robot se met en direction de l'objectif s'il n'a pas d'obstacle devant lui, dans le cas où il en a, les règles de la partie "blocage" prennent le relais.
- **Stratégie en Cas de Blocage (Règles 11 à 16) :**
  - On identifie le robot comme étant bloqué, si un obstacle est proche devant lui, et qu'il est à l'arrêt (sachant que la présence d'un obstacle l'arrête).
  - Si le robot est bloqué et qu'un obstacle est à sa droite, il tournera vers sa gauche puis commencera à longer l'obstacle (anciennement frontal qui est maintenant à sa droite), et vice-versa.
  - **Méthode du balancement (Règles 13 à 16) :**
    - Lorsque le robot détecte un obstacle à devant lui, sans en avoir un à droite ou à gauche, il commence à tourner alternativement à droite et à gauche (se balancer), en commençant par le côté duquel se situe l'objectif, augmentant progressivement l'angle de rotation à chaque balancement. Ce processus continue jusqu'à ce que l'obstacle qui était devant lui soit à sa droite ou à sa gauche, afin de commencer à le longer et de reprendre sa progression. Le balancement offre une solution simple mais efficace pour surmonter les blocages frontaux.
- **Contraintes de sécurité (Règles 17 et 18) :**
  - L'un des problèmes lors de ce robot est la présence d'angles morts, ces-derniers font que, lorsque le robot tourne et avance en même temps, le capteur avant peu capter un obstacle extrêmement proche, ne lui laissant pas le temps de s'arrêter.
  - Pour diminuer la fréquence de se problème, le robot ralenti lorsqu'il tourne.

Ces règles illustrent comment la logique floue permet au robot de naviguer de manière adaptative et intelligente dans un environnement rempli d'incertitudes, en prenant des décisions basées sur une interprétation nuancée des données sensorielles. Cette méthodologie offre au robot les outils nécessaires pour ajuster sa trajectoire de manière dynamique, garantissant une navigation

## Résultats et Analyse

Après l'implémentation des règles de logique floue et la définition précise des variables linguistiques pour le pilotage du robot mobile, nous avons procédé à une série de tests dans divers environnements simulés. Ces tests visaient à évaluer l'efficacité de notre système de contrôle flou dans des scénarios variés, allant d'espaces ouverts simples à des labyrinthes complexes avec de multiples obstacles. Voici une synthèse des résultats obtenus, illustrant la performance du robot dans ces différents contextes.

### Environnement Simple (Map0 et Map1)

Dans les environnements les plus simples, sans obstacle ou avec un seul obstacle, le robot a démontré une excellente capacité à naviguer vers ses objectifs. La logique floue a permis une adaptation fluide à l'environnement, avec des ajustements précis de la vitesse et de la direction en fonction de la position relative

de l'objectif. Le robot a systématiquement atteint ses cibles sans aucune collision, validant l'efficacité des règles de base de la logique floue dans des conditions optimales.

### Environnement Moyennement Complexe (Map2)

L'introduction d'un nombre modéré d'obstacles a mis en évidence la robustesse du système de navigation flou. Le robot a réussi à contourner efficacement les obstacles tout en maintenant sa trajectoire vers l'objectif. Cependant, nous avons observé une légère augmentation du temps nécessaire pour atteindre l'objectif, en raison des détours nécessaires. Cette observation souligne l'importance de l'équilibrage entre l'évitement d'obstacles et l'optimisation du chemin.

### Environnement Complexe (Map3)

Dans les scénarios les plus complexes, comportant plusieurs obstacles placés de manière stratégique, le robot a pu démontrer toute la flexibilité et l'adaptabilité de la logique floue. Malgré quelques hésitations initiales dans des situations de proximité élevée avec plusieurs obstacles, le robot a utilisé la technique du balancement pour trouver des passages viables, évitant ainsi les impasses et progressant vers ses objectifs. Ces tests ont révélé les limites de notre système dans des environnements extrêmement denses, mais ont également souligné sa capacité à gérer des situations délicates grâce à une stratégie d'évitement dynamique.

### ***Analyse des problèmes***

Ces analyses sur les différentes cartes ont misent en évidence un problème, déjà cité plus haut : lorsqu'un obstacle se situe dans un des angles morts avant du robot, et que ce dernier avance en tournant (même lentement ou à l'arrêt), le robot s'écrasera contre cet obstacle. A l'arrêt, ce la peut s'expliquer par le fait que le robot ne soit pas carré ou circulaire, mais rectangle, ce qui créer un porte-à-faux au niveau de ces angles (car son centre de rotation est au centre).

## **Conclusion**

Les résultats obtenus à travers ces tests confirment l'efficacité de l'approche de notre ensemble de règles floues dans la navigation autonome du robot. Bien que des défis demeurent dans la gestion des environnements hautement complexes, les fondements établis par ce système de contrôle offrent une base solide pour de futures améliorations et adaptations. Une piste d'amélioration possible est d'obtenir l'accès à des données plus précises (exemple : chaque capteur indépendamment, pouvoir traiter séparément les capteurs avants, droite et gauche, et pareil pour les autres côtés).