

TP3
MA 300 Outil probabilistes pour la robotique
Filtre à particule

Comme les tds précédents, nous considérons ici un robot mobile se déplaçant dans une salle et l'on souhaite suivre sa position (x_t, y_t) à chaque instant t . L'espace est ici considéré comme continu. Nous considérons ici que la salle possède un certain nombre de marqueurs immobiles. Le robot capte la direction de chaque marqueur, connaît leur position absolue dans la salle, et obtient ainsi des informations sur sa propre position.

Le modèle considéré (plus réaliste) est le suivant :

- Le robot est caractérisé par sa position (x_t, y_t) dans la salle ainsi que par son orientation θ_t . Entre chaque mesure, le robot peut effectuer une rotation, et donc changer son orientation, puis se déplacer en ligne droite d'une certaine distance. On suppose que les erreurs d'orientation et de distance parcourue sont gaussiennes, sont centrées sur le déplacement désiré, et ont pour variances respectives les paramètres *turn_noise* et *distance_noise*.
- Le robot reçoit pour chaque marqueur la direction selon laquelle il le perçoit. Il reçoit donc autant de mesures qu'il y a de marqueurs dans la salle. On suppose que le bruit du capteur est gaussien, de variance *measure_noise*, et qu'il s'applique indépendamment à toutes les mesures.
- La position initiale du robot est inconnue. Le robot connaît la position absolue dans la salle de chaque marqueur.
- On ne gère pas les conditions de bord de la salle. Pour cela on supposera qu'elle est cyclique, c'est à dire que si le robot sort par un bord de la salle, il rentre par le bord opposé.

Nous nous proposons dans ce TD, à partir du squelette de code fourni, d'utiliser un filtrage à particule pour évaluer la position et l'orientation du robot au cours du temps.

- 1) Comme nous n'avons aucune information sur la position initiale du robot, nous allons générer les particules de manière aléatoire et uniforme dans toute la salle. Compléter la fonction *generate_particles* en conséquence (ne pas oublier d'initialiser également les paramètres relatifs au bruit). Vous devriez voir la répartition des particules dans la salle en lançant le programme (plus la couleur tend vers le rouge, plus il y a de particules à cet endroit de la salle).
- 2) Compléter la fonction *move(self, motion)*, qui met à jour les caractéristiques de la particule (position et orientation) en fonction du changement d'orientation suivie du déplacement selon la distance demandée, et des bruits gaussien respectifs.
Décommentez la ligne $p = p2$ de la fonction *particle_filter* et lancer le programme. En l'absence de mesure et de rééchantillonnage, la répartition des particules dans la salle varie, mais reste uniforme.
- 3) Compléter la fonction *sense*, en calculant pour chaque marqueur la direction relative selon laquelle il est perçue, c'est à dire la différence entre sa direction absolue et la direction absolue de la particule. Pour chaque mesure, ajouter un bruit gaussien de variance *self.measure_noise*.
Chaque particule calcule ensuite grâce à cette fonction la valeur théorique (c-à-d avec un bruit nul) des mesures qu'elle devrait recevoir à sa position. En comparant avec les mesures effectuées par le robot, on obtient dans la fonction *measurement_prob* la probabilité qu'une particule soit sur la position du robot.
- 4) Il reste maintenant à rééchantillonner les particules suivant cette probabilité. Compléter la fonction *resample(p, w)*, qui tire *len(p)* fois une particule de *p* au prorata de sa probabilité selon une loi uniforme, et renvoi le nouveau vecteur de particules ainsi obtenu.
Vous devriez maintenant voir à chaque itération les particules avant et après le rééchantillonnage, qu'observe-t-on ?
Simuler une seule itération, puis deux. Comment expliquer la différence de répartition à la fin de la première itération ?
- 5) Finalement, compléter également la fonction *get_position(p)*, qui estime la position et l'orientation du robot comme la moyenne des particules sélectionnées par le rééchantillonnage.

- 6) Utilisez des landmarks aléatoires (décommentez les deux lignes à la fin des variables globales), et faites varier le nombre de marqueur, puis le nombre de particules. Que pouvez vous dire de l'importance de ces paramètres ?
Que se passe-t-il s'il n'y a qu'un seul marqueur (utilisez beaucoup de particules), et pourquoi ?
En particulier, est-ce que placer cet unique marqueur au centre de la salle est une bonne idée ?