

TD: Robotics Toolbox [initiaux]

Le but de ce TD est d'utiliser la boîte à outils *Robotics Toolbox*, créée par Peter Corke (Queensland University of Technology, Australia), dans des problèmes de localisation et navigation de robots mobiles: <http://petercorke.com/wordpress/toolboxes/robotics-toolbox>

Introduction

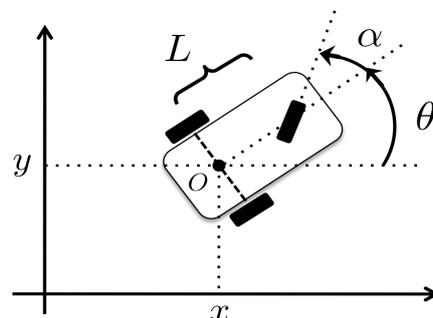
Robotics Toolbox est une boîte à outils Matlab qui fournit de nombreuses fonctions qui sont utiles pour l'étude et la simulation de robots industriels (modèle géométrique, cinématique et dynamique, génération de trajectoires, etc.) avec une méthode de représentation très générale. La boîte à outils supporte aussi les *robots mobiles* avec des algorithmes de planification de chemin, localisation (EKF, filtre particulaire), création de cartes (EKF), localisation et cartographie simultanée ou SLAM (EKF), avec plusieurs modèles Simulink de véhicules (robot car-like, véhicule de Braitenberg, quadricopter).

Les différentes fonctions de la boîte à outils sont présentées dans le fichier « robot.pdf » (rvctools/robot/robot.pdf) et sur le lien: www.petercorke.com/RTB/r9/html/index_alpha.html

Pour utiliser la boîte à outils, téléchargez le dossier « robot-9.10.zip » (RTB 9.10) sur le site web, copiez le répertoire « rvctools » dans votre espace de travail et lancez le fichier de démarrage « startup_rvc.m » qui placera les bons répertoires dans votre chemin Matlab. La démo « rtbdemo » montre les fonctionnalités principales de la boîte à outils.

Exercice 1 : Créer et déplacer un robot mobile

Utiliser la classe « Vehicle », qui simule un robot car-like (tricycle), pour créer un robot, le déplacer suivant une trajectoire aléatoire et récupérer sa position instantanée.



- Créer un robot « veh » avec une matrice de covariance odométrique (rappel la matrice Σ_{Δ}) et afficher son état initial.
- Utiliser la fonction « update » pour appliquer une vitesse de 0.2 m/s avec une orientation de 0.1 rad pour un pas de temps. Afficher la nouvelle position du robot.

- Attacher un pilote (« driver ») au robot pour le faire déplacer suivant une trajectoire aléatoire dans une région $-10 < x < 10$ mètres, $-10 < y < 10$ mètres.
- Visualiser le mouvement du robot pour $N = 1000$ instants, en utilisant l'option « run », et récupérer l'état $[x, y, \theta]^T$ du robot.

Exercice 2 : Localisation Monte Carlo

Le but de ce deuxième exercice est d'estimer la pose d'un robot avec le filtre particulaire en utilisant l'odométrie et les observations de points d'intérêt connus (marqueurs).

- Utiliser la fonction « map » pour la création d'une carte de 20 marqueurs.
- Créer un robot « veh » avec une matrice de covariance odométrique et un pilote.
- Créer un capteur « scanner » en utilisant la fonction « RangeBearingSensor »
- Pour le filtre particulaire, nous avons besoin de définir deux matrices de covariance:
 - La première, $\mathbf{Q} \in \mathbb{R}^{3 \times 3}$, est la covariance du bruit gaussien ajouté aux états des particules à chaque itération pour représenter l'incertitude en configuration.
 - La deuxième, $\mathbf{M} \in \mathbb{R}^{2 \times 2}$, est la covariance de la fonction de vraisemblance appliquée à l'innovation.
- Utiliser la fonction « ParticleFilter » pour construire le filtre particulaire qui est configuré avec 1000 particules. Ces particules sont initialement uniformément distribuées dans l'espace de configuration à trois dimensions.
- Exécuter la simulation pour $N = 800$ instants, tracer la carte et le chemin du véhicule.
- Superposer la moyenne du nuage de particules.

Exercice 3 : Utilisation du filtre de Kalman étendu pour la localisation et la cartographie

Cet exercice utilise la classe « EKF » qui peut être exploitée pour la localisation basée sur une carte, pour la cartographie et pour la localisation et cartographie simultanées (SLAM).

a) Estimation de la position d'un robot

- Créer un véhicule « veh » avec une matrice de covariance odométrique \mathbf{S} .
- Ajouter un pilote avec une trajectoire aléatoire.
- Créer un filtre de Kalman avec une covariance estimée \mathbf{V}_{est} et une covariance initiale de l'état \mathbf{P}_0 .
- Exécuter le filtre pour N pas de temps.
- Afficher le chemin du véhicule, superposer le chemin estimé, et ajouter des ellipses d'incertitude tous les 10 pas de temps.

b) Localisation basée carte

- Créer un véhicule « veh » avec une matrice de covariance odométrique \mathbf{S} .
- Ajouter un pilote avec une trajectoire aléatoire.
- Créer une carte avec 20 marqueurs.
- Créer un capteur qui utilise la carte et l'état du véhicule pour estimer les primitives du capteur avec une covariance \mathbf{W} .
- Définir le filtre de Kalman avec les covariances estimées \mathbf{V}_{est} et \mathbf{W}_{est} et la covariance initiale de l'état du véhicule \mathbf{P}_0 , puis exécuter le filtre pour N pas de temps.
- Tracer la carte, afficher le chemin du véhicule, superposer le chemin estimé, et ajouter des ellipses d'incertitude tous les 10 pas de temps.

c) Création de cartes par un robot

- Créer un véhicule « veh » avec une matrice de covariance odométrique \mathbf{S} .
- Ajouter un pilote avec une trajectoire aléatoire.
- Créer un capteur qui utilise la carte et l'état du véhicule pour estimer les primitives du capteur avec une covariance \mathbf{W} .
- Définir le filtre de Kalman avec la covariance estimée \mathbf{W}_{est} et un véhicule « parfait » (utiliser une matrice de covariance "[]"), puis exécuter le filtre pour N pas de temps.
- Tracer la carte et superposer les ellipses d'incertitude 3-sigma estimées.

d) Localisation et cartographie simultanées (SLAM)

- Créer un véhicule « veh » avec une matrice de covariance odométrique \mathbf{S} .
- Ajouter un pilote avec une trajectoire aléatoire.
- Créer une carte avec 20 marqueurs.
- Créer un capteur qui utilise la carte et l'état du véhicule pour estimer les primitives du capteur avec une covariance \mathbf{W} .
- Définir le filtre de Kalman avec les covariances estimées \mathbf{V}_{est} et \mathbf{W}_{est} et la covariance initiale de l'état du véhicule \mathbf{P}_0 , puis exécuter le filtre pour N pas de temps pour estimer l'état du véhicule à chaque pas de temps et la carte.
- Tracer la carte, afficher le chemin du véhicule, superposer le chemin estimé et ajouter des ellipses d'incertitude tous les 10 pas de temps.

Exercice 4 : Simulation d'un véhicule de Braitenberg (version 4a)

- Ouvrir le modèle Simulink « sl_braitenberg » et analyser sa structure.
- Simuler le mouvement du véhicule vers la source à (60, 90) avec des conditions initiales différentes (voir le bloc « Bicycle »). Afficher les chemins du véhicule.