

Cartographie et localisation simultanées par filtrage de Kalman étendu

David FILLIAT

16 avril 2019

1 Introduction

Dans ce TP, nous allons étudier une méthode de cartographie et de localisation simultanées (SLAM) dans un environnement inconnu utilisant un filtre de Kalman étendu. Pour cela, nous utiliserons le code MATLAB réalisé par [Tim Bailey](#) légèrement modifié, que vous téléchargerez [sur ma page](#).

Ces programmes permettent de simuler un robot se déplaçant sur une trajectoire donnée dans un environnement constitué d'amers ponctuels. Ils implémentent également une méthode de filtrage de Kalman étendu simple utilisant la perception de la direction et de la distance de ces amers ponctuels. Le vecteur d'état (variable globale `XX` dans le code) est de la forme :

$$X = \begin{bmatrix} x \\ y \\ \theta \\ x_{a1} \\ y_{a1} \\ \vdots \\ x_{aN} \\ y_{aN} \end{bmatrix}$$

Sa matrice de covariance associée est stockée dans la variable `PX`. La plupart des paramètres à modifier dans le code se trouvent dans le fichier `configfile.m`

2 Prise en main

Chargez les vecteurs décrivant la trajectoire du robot (`wp`) et les amers de l'environnement (`lm`) stockés dans le fichier `TD-map.mat`. Lancez ensuite la simulation de cartographie (`ekfslam_sim(lm, wp)`). Par défaut, l'association de données est supposée connue, c'est à dire que pour chaque amer perçu, l'amer correspondant de la carte est identifié sans ambiguïtés. Cela permet notamment de gérer convenablement les fermetures de boucle, même lorsque l'erreur dans la carte est très forte. Observez en particulier le comportement de l'algorithme lors de la fermeture de boucle (en particulier l'évolution des ellipses de covariance).

Utilisez l'utilitaire `frontend`¹ pour créer des cartes et des trajectoires. Étudiez le comportement de l'algorithme de SLAM avec différents types de cartes et de trajectoires :

- des cartes denses avec une trajectoire réalisant des boucles courtes
 - des cartes denses avec une trajectoire réalisant une boucles très longue
 - des carte avec très peu d'amer réalisant une boucles très longue
- Que constatez-vous à chaque fois pour la précision de l'estimation de la carte ?

1. Attention : pour ajoutez des points, cliquez sur 'waypoint' avec le bouton gauche de la souris, puis ajoutez des points avec le bouton gauche et quand vous avez fini, **cliquez sur le bouton droit avant de passer aux landmarks**. Si vous avez des problèmes avec `frontend`, reprenez le code d'origine dans le fichier zip.

3 Modèles probabilistes

Faites varier les niveaux de bruit dans les processus de commande et d'observation utilisés pour la simulation (`sigmaV`, `sigmaG`, `sigmaR`, `sigmaB` dans le fichier `configfile.m`). Etudiez l'influence de chaque paramètre. Que se passe-t-il lorsque les bruits augmentent fortement ?

Modifiez la structure du bruit dans le fichier `add_observation_noise.m` afin de le rendre non gaussien, par exemple en ajoutant un fort biais positif. Que se passe-t-il pendant la cartographie ?

Quel est le rôle du paramètre `SWITCH_INFLATE_NOISE` ? Permet-il de compenser les bruits non gaussiens ?

4 Association de données

Modifiez le paramètre `SWITCH_ASSOCIATION_KNOWN` pour utiliser des amers non identifiables. Ecrivez le code du fichier `data_associate.m` pour réaliser l'association de données entre un amer perçu et l'amer de la carte le plus proche en distance euclidienne.

Pour chaque amer perçu, contenu dans la variable `z`, vous devez :

- Calculer sa position absolue en prenant la position du robot au début du vecteur d'état,
- Trouver l'amer i de la carte qui est le plus proche de cet amer perçu,
- Si la distance à l'amer i est inférieure à `gate1`, ajouter cet amer dans la variable des amers connus `zf` et l'index i de cet amer dans la variable `idf`,
- Si la distance à l'amer i est supérieure à `gate2`, ajouter cet amer dans la variable des amers nouveaux `zn`

Étudiez comment l'algorithme se comporte en fonction des bruits des différents modèles, de la densité des amers et de la taille des boucles de la trajectoire.

Bonus : distance de Mahalanobis

Modifiez l'association de donnée pour utiliser la distance de Mahalanobis en remplacement de la distance euclidienne :

$$d^2 = (zp - z)^T (H * P * H' + R)^{-1} (zp - z)$$

où z est l'observation, $zp = Hx$ l'observation prédite, H la matrice d'observation, P la matrice de covariance de l'état et R la matrice de covariance des observations.