

Cours ISC - 2A

TD et Correction - Commande d'un robot suiveur de ligne

Thao Dang

CNRS-VERIMAG, 2, av. de Vignate, 38610 Gieres, France

1 Atteindre un point cible

Dans le TD précédent, nous avons calculé deux contrôleurs qui réalisent les trajectoires du robot en fonction de la différence entre les coordonnées courantes du robot et celles du point cible (x_{final}, y_{final}) .

Questions 1.

- Tester le contrôleur du modèle fourni. Justifier les valeurs des paramètres de ce contrôleur. Pourrait-on les améliorer? **Réponse :** *Ces valeurs des paramètres garantissent la stabilité. On pourrait les améliorer pour d'autres critères de performance.*
- Pour implanter le contrôleur du modèle fourni, il faut le discrétiser. Si on choisit une période d'échantillonnage T et une approximation du premier ordre, on a

$$s \approx \frac{z - 1}{zT}.$$

Trouver une période d'échantillonnage T appropriée et la valider en utilisant par simulation.

Réponse : *En plus du critère de Shannon, il est important de garantir que la boucle fermée discrétisée soit stable. Pour ceci, par exemple, quand on considère le contrôleur d'angle, il faut discrétiser la dynamique de l'orientation θ et puis calculer la fonction de transfert en Z de la boucle fermée.*

2 Suivre une ligne noire

Nous allons maintenant considérer le problème de faire suivre au robot une ligne en utilisant des capteurs de lumière.

Le robot possède deux capteurs de lumière G (gauche) et D (droite) positionnés sur le devant et qui pointent vers le sol afin de détecter une ligne noire. Le robot possède deux moteurs connectés à ses deux roues. On doit connaître les valeurs que les capteurs retournent quand ils voient une « couleur » entre le noir et le blanc. Autour de chaque capteur, il y a une petite zone qu'il peut voir.

Hypothèse : Supposons que les capteurs soient calibrés sur une échelle de 0 (noir) à 100 (blanc) (il faut vérifier cette hypothèse et ajuster le calibrage si besoin). Si la zone est complètement noire, le capteur retourne 0. Si la zone est complètement blanche, le capteur retourne 100.

Si un capteur retourne une valeur $c < 100$, on sait que le robot n'est plus aligné avec la ligne et il faut le faire tourner vers ce capteur d'un angle proportionnel à cette valeur c .

On peut utiliser les informations fournies par les capteurs comme suit (on suppose que le robot est initialement aligné sur la ligne) :

- Si les deux capteurs voient le blanc, le robot va tout droit (sauf dans des cas particuliers tels que dans une manœuvre de demi-tour).
- Si un seul capteur voit le blanc, le robot tourne vers le capteur qui voit le noir.
- Si les deux capteurs voient le noir, le robot s'arrête et attend l'intervention de l'utilisateur pour le remettre sur la ligne.

Pour réaliser cette stratégie, notons que l'orientation θ du robot est contrôlée par la différence entre les vitesses des deux roues v_G et v_D , ce qui est réalisé par le contrôleur d'angle. La vitesse du robot est contrôlée par la moyenne de v_G et v_D , ce qui est réalisé par le contrôleur de distance.

Dans le TD précédent, nous avons calculé deux contrôleurs qui réalisent les trajectoires du robot en fonction de la différence entre les coordonnées courantes du robot et celles du point cible (x_{final}, y_{final}) . Nous allons maintenant combiner ces contrôleurs avec un planificateur de trajectoire qui change le point cible « implicitement » en fonction des informations venant des capteurs.

Le problème est maintenant de suivre une ligne que le robot ne peut pas connaître. Il peut seulement, à partir des informations venant des capteurs, estimer sa déviation par rapport à la ligne.

Nous allons d'abord contrôler l'orientation du robot, c'est-à-dire déterminer la commande qui agit sur la différence $v_G - v_D$. Pour corriger l'angle du robot, on définit l'erreur d'angle (la différence entre l'angle courant et l'angle désiré) que l'on veut ramener à 0. Notons que dans le contexte du suiveur de ligne, on peut estimer cette différence sans connaître l'angle désiré.

Ensuite, cette erreur est donnée en entrée du PID du contrôleur d'angle.

On doit encore contrôler la vitesse du robot, car le contrôleur d'angle détermine seulement la différence entre v_G et v_D .

- Quand le robot ne doit pas tourner, on peut laisser la vitesse du robot constante, par exemple en créant à l'entrée du contrôleur de distance une « erreur » de distance constante. Dans ce cas, une solution plus simple est de désactiver la partie intégrale du PID et de garder l'entrée constante.
- Quand le robot doit tourner, si la vitesse courante du robot est trop grande et que la ligne est très courbée, le robot peut faire de grands dépassements. Donc, quand le robot doit tourner beaucoup, on devra réduire sa vitesse (par exemple, en réduisant l'erreur à l'entrée du contrôleur de distance).

Question 2 : Ecrire un modèle de commande qui réalise la stratégie ci-dessus.

A vous de le faire !!

3 Commande avec un seul capteur de lumière

Il est possible de réaliser un robot suiveur de ligne avec un seul capteur de lumière, en mettant le capteur au-dessus d'un bord de la ligne, par exemple celui de droite.

Question 3 : Proposer une stratégie de commande pour ce cas. Quelles sont les contraintes de cette méthode de commande (par rapport à celle qui utilise deux capteurs de lumière ?)

Réponse : *On peut positionner le capteur au-dessus d'un bord de la ligne, par exemple celui de droite. Si le capteur donne une valeur autour de 50, ceci indique que le robot est plus ou moins aligné avec la ligne. Une valeur plus petite que 50 indique que le robot est dévié vers la droite. Pourtant, avec un seul capteur il est difficile d'obtenir une bonne précision de l'estimation de l'erreur et le robot devra rester suffisamment lent.*

4 Émulation des capteurs

La figure 1 illustre les sorties des capteurs quand le robot est en train de suivre un segment de ligne dont un point limite est l'origine, et la pente de ce segment est θ_l . La position courante du robot est représentée par le point C avec les coordonnées (x, y) et son orientation est θ .

On peut voir sur cette figure que le robot est dévié vers la gauche et que la zone de vue du capteur G intersecte la ligne noire. La sortie de ce capteur dépend donc du ratio entre la surface de la zone sombre et la surface totale de la zone de vue.

Question 4 : Ecrire un programme Matlab qui simule les sorties des capteurs.

Un modèle Simulink pour l'émulation des sorties des capteurs est fourni (voir la page du cours). Des ajustements nécessaires peuvent être faits en fonction des tests sur le vrai robot.

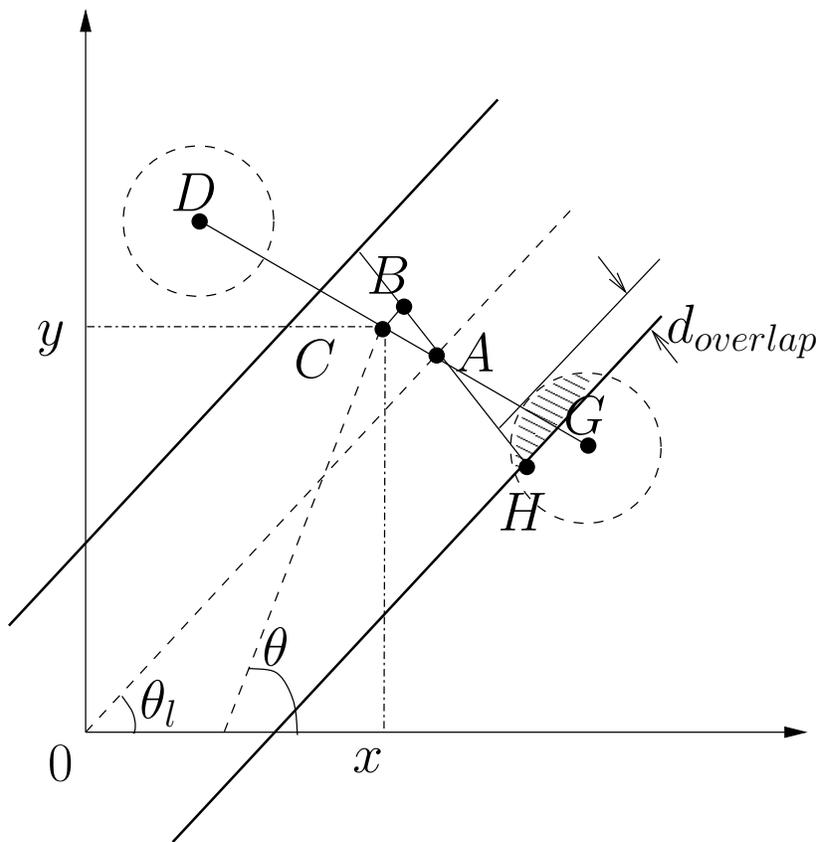


Fig. 1. Émulation des capteurs