



Université de Paris

Projet Long

Féaux de Lacroix Martin
22015860

Fildago Alex
71600135

Année 2020-2021

1 Présentation du projet :

Notre but était de construire un robot et implémenter un algorithme capable de suivre une ligne. Le robot possède une caméra retournant la couleur vue et deux roues motrices (et une roue directrice à l'arrière). Le système d'exploitation que nous utilisons est Ev3dev avec une interface pour python. Notre algorithme est capable de scanner puis reconnaître plusieurs couleurs différentes puis de suivre une ligne de cette couleur sur un fond prédéfini.

2 Documentation :

Installation :

1. Installation d'Ev3dev
2. Upload les fichiers à la brick via ssh.

Utilisation du programme :

1. Lancer le fichier LineFollower.py
2. Calibre le capteur sur du blanc
3. Scanner la ou les différentes couleurs de la ligne (à l'aide des boutons).
4. Scanner le background si il est non blanc.
5. Lancer le suivit de ligne (penser à mettre le robot à gauche de la ligne).

3 Implémentation :

Le capteur :

Notre capteur retourne la moyenne des couleurs présentes dans la zone "observée". C'est-à-dire que si la zone observé est composé de 50% de blanc et 50% de noir, le capteur renverra un triplé approchant (127, 127, 127).

PID :

L'algorithme gère la réponse des moteurs en fonction de l'erreur.

Proportionnelle :

La composante la plus importante qui renvoie une valeur proportionnelle à l'erreur courante.

Intégral :

L'intégral correspond à la valeur sous la courbe d'erreur. Ici, on donne $integral = \sum_{n=0}^{erreur=0} erreur$. L'augmentation de l'intégral correspond à une perte de la ligne.

Dérivé :

La dérivé mesure l'ampleur du changement de la courbe d'erreur. Ici, on donne $driv = erreur - prcdenteerreur$. On utilise la dérivé pour ajuster en cas de changement rapide de l'erreur. (exemple : le robot n'étais plus sur la ligne, il l'as retrouve et pour éviter qui dépasse la ligne, on ajuste la réponse).

Reconnaissance des couleurs :

Méthode 1 :

Uniquement utile si la couleur possède d'important variation de son teint (partie à l'ombre, l'autre à la lumière, ect..).

Prend pleins de points en se baladant entre les zones, pour chaque composant calculé la moyenne et l'écart type. Tout autre point appartient à la couleur si pour chacun de ses composant du triplé RGB, $moyenne - \sigma \times 3 < valeur < moyenne + \sigma \times 3$

Méthode 2 :

Mesuré la distance entre la couleur courante et une couleur que l'on connaît (formule : $\sqrt{(R - R')^2 + (G - G')^2 + (B - B')^2}$)

Le tout :

Une seule couleur :

On utilise le PID pour gère la puissance des moteurs, avec $pid = PID(erreur)$ et $moteur_gauche = pid - offset$ et $moteur_droite = -pid + offset$. L'off-

set permet d'aller à gauche de base et l'erreur est calculé par $erreur = \frac{distance(couleur_actuel, couleur_ligne)}{distance(couleur_ligne, couleur_fond)} \times 100$

Plusieurs couleurs :

En cas de plusieurs couleurs, il faut d'abord estimer quel est la couleur courante, pour cela on regarde quelle couleur à la plus petite distance avec la couleur visionnée. (note : plus la caméra est en dehors de la ligne, plus il y a une chance qu'on ne matche pas la bonne couleur, cela dit la différence entre deux erreurs élever ne change pas grand-chose). Puis une fois la couleur identifiée on fait la même chose que pour une seule couleur.