

A. Présentation du projet

Membres : Idjellidaine Joris, Petic Lucian

Le projet consiste à implémenter un algorithme efficace pour qu'un robot ev3dev puisse suivre une ligne tracée au sol. Ce type de robots sont notamment utilisés dans la logistique industrielle pour assurer l'acheminement des colis d'un point A à un point B.

Lors de son développement, le projet a été séparé en plusieurs parties avec notamment la partie apprentissage des couleurs et la partie programmation des algorithmes.

En dehors de l'installation du robot, une des premières difficultés du projet a été l'apprentissage des couleurs. Étant donné que de nombreux facteurs entrent en jeu, tels que l'éclairage de la salle ou bien les couleurs elles même, il nous aura fallu trouver le bon réglage de hauteur afin de maximiser l'espace vu par le robot sans pour autant perdre en précision sur les couleurs.

Après cette étape, il nous a fallu commencer à faire fonctionner le robot lui-même. Notre implémentation étant faite en Python, cette étape a été relativement rapide grâce au fait que Python est un langage de haut niveau et aux libraires déjà présentes.

L'implémentation de l'algorithme simple initial aura pris un peu de temps, le temps de découvrir quelle méthode utiliser pour suivre la ligne. Ensuite, implémenter et optimiser cette méthode nous aura pris quelques semaines.

Afin de permettre le travail de groupe, notre méthode de travail aura été de se donner rendez-vous sur Discord et de montrer l'avancement actuel du robot tout en donnant des idées supplémentaires et en travaillant sur le code.

Après la première soutenance des robots, l'implémentation de l'algorithme PID a été faite. Le plus dur aura été de trouver ce qui constitue une implémentation correcte de cet algorithme dans le contexte de la vue limitée du robot, puis de paramétrer cet algorithme afin d'obtenir un résultat satisfaisant.

B. Documentation

INSTALLATION DE L'ENVIRONNEMENT

Premièrement, il faut préparer l'environnement en installant le système d'exploitation sur la carte SD. Il faut rendre la carte SD bootable.

- Télécharger l'OS depuis le site [ev3dev](http://ev3dev.org)
- Installer l'OS sur la carte SD avec balenaEtcher
- (regarder le fichier **docs/history.md** pour plus de précisions)

INSTALLATION DU PROJET

Il y a deux manières différentes de copier le code Python sur le robot :

VIA BLUETOOTH

Il faut se connecter par Bluetooth au robot et effectuer les commandes suivantes:

```
make id # send ssh key
make send # send the script
```

Pour le développement, d'autres commandes sont disponibles dans le Makefile.

VIA CABLE

L'envoi des fichiers se passe comme pour un périphérique externe classique.

LANCEMENT DU PROJET

VIA BLUETOOTH

```
make con # connected with ssh
cd ./res # get to the directory
./Basic.py # launch the script
./PID.py # launch the script
```

VIA ROBOT

Il faut chercher l'exécutable Basic.py ou PID.py et le lancer depuis le robot.

Dans son état actuel, le robot implémente l'exécution de l'algorithme simple ou de l'algorithme PID, ainsi que la reconnaissance des couleurs afin de tester le robot et la gestion de la vitesse et de l'angle maximal avant lancement.

La reconnaissance des couleurs se contente d'afficher constamment la couleur perçue par le capteur. Cela permet de facilement lire et déboguer les valeurs obtenues par celui-ci.

Aux lancements, les deux algorithmes se comportent de la même façon : Il vont tout d'abord commencer par émettre un bip sonore afin de transmettre à l'utilisateur que le robot est prêt à capter les couleurs. Pour cette étape, l'ordre de scan est le suivant :

1er scan: le sol

2ème scan: la ligne

3ème scan: la ligne de couleur annexe constituant le croisement

4ème scan: la ligne de couleur correspondant a la ligne de départ

Un bip sonore est audible au début et à la fin de chaque scan.

Après ces 4 scans, il sera possible d'augmenter ou de diminuer la vitesse maximale avec les boutons haut et bas respectivement, ou bien d'augmenter ou diminuer l'angle maximal avec les boutons droite et gauche respectivement.

Appuyer sur le bouton central durant cette phase lance le robot et démarre l'exécution de l'algorithme choisi.

C. Détails d'implémentation

1. Algorithme basique

L'algorithme basique suit un principe simple, quand le robot rencontre le sol, il tourne vers la droite, quand il rencontre la ligne, il tourne vers la gauche. Initialement, la vitesse de rotation est faible puis augmente progressivement avec chaque scan successif du sol ou de la ligne.

Cet algorithme est seulement capable de reconnaître soit une ligne, soit le sol et ne considère pas les valeurs entre deux. De ce fait, le robot aura tendance à manquer de souplesse, notamment dans les virages

2. Algorithme PID

L'algorithme PID utilise le principe PID pour déterminer l'angle et la vitesse à utiliser. Cet algorithme utilise l'erreur obtenue par rapport à la valeur attendue pour la bordure de la ligne pour tourner correctement. En plus de cette valeur, l'algorithme possède une mémoire des erreurs précédentes et utilise chaque erreur successive afin d'augmenter l'angle de rotation.

Contrairement à l'algorithme simple, celui-ci détecte les nuances entre la ligne et le sol et est capable de corriger sa route en conséquence, il est donc beaucoup plus progressif dans sa direction et ne changera pas de direction soudainement, cependant, il demande aussi bien plus de temps à configurer.

3. Reconnaissance des couleurs

La reconnaissance des couleurs utilise la valeur RVB actuellement renvoyée par le capteur et la valeur en mémoire lors de la phase de scan pour déterminer la couleur perçue. Pour ce faire, l'algorithme utilisé calcule une marge acceptable dans laquelle la couleur perçue est considérée comme une des couleurs scannées précédemment.

La marge est calculée en fonction de la plus petite différence entre toutes les couleurs scannées et si les trois valeurs RVB de la couleur perçue sont dans la marge RVB d'une des couleurs scannée, alors la couleur perçue est considérée comme étant la même que cette couleur en mémoire.