
Comment utiliser le code ?

Généralités

MainLabyrinthe permet au robot de se déplacer dans le labyrinthe tout en se recalant sur les murs et en se remettant droit en s'aidant des lignes blanches. Il utilise la classe ChangeSquare qui rassemble les différentes capacités du robot.

MainSuiveur permet à un robot à 2 capteurs lumineux de se déplacer en suivant une ligne et de prendre la route désignée à une intersection. Il utilise la classe PID qui permet le calcul de PID.

Réglages

Pour les déplacements dans le labyrinthe, il suffit, dans MainLabyrinthe, de régler motorG, motorD, wheelDiameter, et trackWidth. En fonction de la constitution mécanique du robot et des branchements, on modifiera la section correspondante dans ChangeSquare. Après avoir défini dans le String[] parcours le chemin que doit suivre le robot, on upload le code sur la brique NXT et le tour est joué !

Si les moteurs de propulsion du robot sont montés « à l'envers » (càd que pilote.forward() ; fait reculer le robot), il faut mettre un signe « - » devant tous les angles et les distances, et remplacer forward() par backward() et inversement.

Pour les déplacements en suivant une ligne, il faut régler les sections « A changer en fonction de la mécanique du robot » et « Valeurs à régler si on le souhaite » dans MainSuiveur. Les valeurs de kp, ki et kd dans la classe ToolOne sont aussi à définir avec soin. Valeurs conseillées : ki = 0, régler le kp comme on le souhaite en fonction des réactions du robot et régler le kd environ 10 fois plus grand la plupart du temps.

Conclusions

Vidéos illustrant le déplacement d'un robot dans un labyrinthe :

- Sans recalages : https://youtu.be/tlb713Xm_5o
- Avec recalage sur les lignes blanches pour se remettre droit, mais sans recalage grâce aux ultrasons : <https://youtu.be/hVklAHzKBA>
- Avec recalage sur les lignes blanches et recalage ultrason en se repérant par rapport aux murs : <https://youtu.be/UEQHLLIFqv4>

Les différents recalages semblent utiles lorsque le robot réalise un parcours plutôt long dans le labyrinthe, puisque cela lui permet d'arriver à destination sans rentrer dans les murs, même s'il est un peu plus lent.

Robot suivant une ligne et prenant la route de gauche à une intersection :
<https://youtu.be/GLzIlB1S6ao>

Le mode de déplacement en suivant les lignes semble aussi fiable et plus rapide que le déplacement case par case dans un labyrinthe. Il est ludique, assez rapide et pas trop compliqué à coder, ce qui le rend adapté pour une compétition de robotique accessible à davantage de monde.