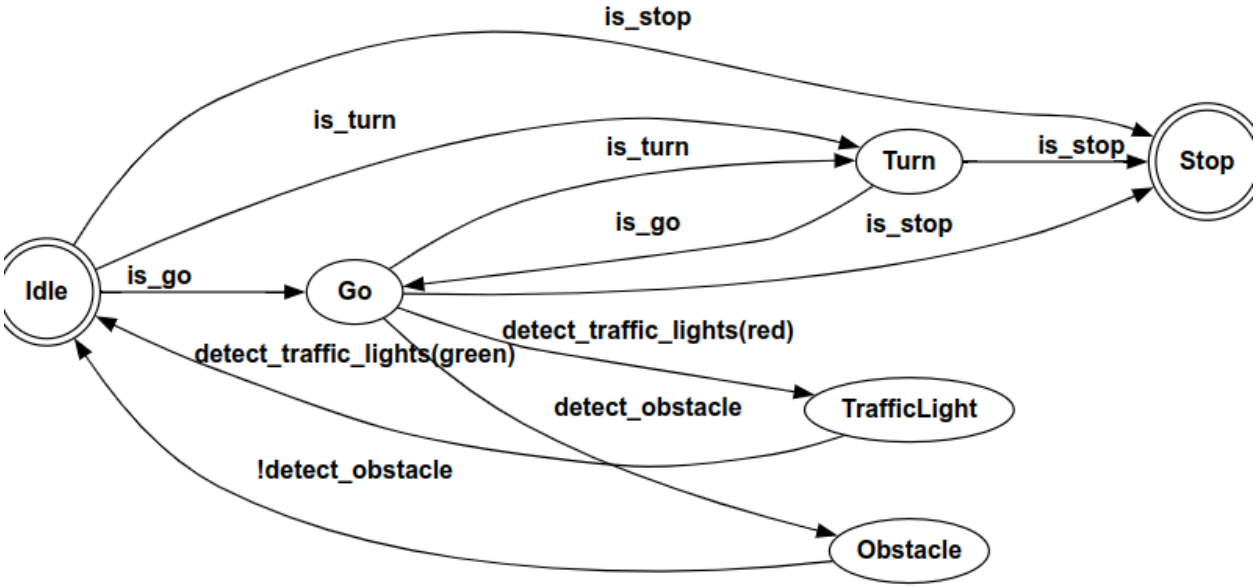
Rapport de projet de Programmation Synchrone

Introduction

Le but de ce projet est de développer un contrôleur pour un véhicule autonome évoluant dans une ville virtuelle, tout en respectant les conditions telles que les limitations de vitesse, les feux rouges et des obstacles.

Fonctionnement de notre véhicule

Notre véhicule autonome possède un automate qui lui permet de faire des choix par rapport à son environnement.



L'automate peut se trouver dans 6 états:

1. L'état **<Idle>** : Le véhicule se trouve par défaut dans cet état. Il y retourne lorsqu'un obstacle ou un feu a été détecté.
2. L'état **<Go>** : On se trouve dans cette état lorsque l'action de l'étape courant est *go*. Elle ajuste sa vitesse de manière autonome en régulant la rotation de ses roues. Un contrôleur PID calcule d'abord la fonction de commande en se basant sur l'écart entre la position courante du véhicule et la position où il devrait être (le guide bleu). Cette commande permet d'ajuster la vitesse de rotation des roues. Le contrôleur détermine ensuite la vitesse de rotation maximale des roues en tenant compte de la limitation de vitesse. Enfin, en fonction

de la direction à suivre (déterminée grâce à la position du véhicule sur la route), cette vitesse maximale est ajustée pour guider le véhicule.

3. La fonctionnalité **<Turn>** : On se trouve dans cette état lorsque l'action est *turn.* Pour faire pivoter le véhicule, on calcule le temps nécessaire pour le faire par rapport à la vitesse de rotation des roues puis on effectue ce pivotement de l'angle donné pendant le temps calculé.
4. L' état **<Obstacle>** : Lorsqu'un obstacle est détecté à une certaine distance grâce au sonar situé à l'avant du véhicule. Le véhicule est arrêté tant que l'obstacle est présent.
5. L'état **<TrafficLight>** : Lorsqu'on détecte un feu rouge, la vitesse du véhicule est mise à zéro et le véhicule attend que son capteur frontal détecte un feu vert avant de redémarrer.
6. L'état **<Stop>** : Il s'agit de l'état de fin d'itinéraire. Lorsqu'on atteint le point d'arrivée.

Remarques sur le code

- Initialement, un contrôleur Bang-Bang a été utilisé, mais il causait des oscillations croissantes. Nous avons donc décidé d'implémenter un contrôleur PID avec les valeurs kP, kI et kD qui ont été déterminées après des tests répétés sur les différentes cartes fournies.
- Nous avons identifié certains cas spécifiques qui nécessitent l'ajout de valeurs « magiques » pour ajuster le comportement du véhicule.
    - *guide_close* : on obtient grâce à la fonction compare_colors une corrélation des couleurs entre 0.70 et 1 pour les couleurs de la route. Avec notre PID, lorsque le véhicule est proche du guide, l'ajustement calculé est trop faible mais le véhicule "tourne" toujours. Cette variable égale à 0.85 de dire a notre véhicule d'aller tout droit si on dépasse cette valeur.
    - *no_trigger_speed_radar* : une valeur de sécurité pour éviter de dépasser les radars.
- Problème sur les maps 6 et 7: Lorsqu'on atteint l'étape 4 de l'itinéraire (*go 10*), le véhicule devrait continuer le tour du "rond point". Mais le véhicule croise la route perpendiculaire et roule sur la partie cyan de cette route qui fait tourner le véhicule à droite ce qui provoque une sortie de route.
- Problème sur la map 9: Sur l'étape 12 (*turn 100*) le véhicule ne détecte pas la marque verte sur la route. Après des tests, le véhicule est pourtant au centre de la route et devrait donc détecter cette marque.