

Rapport de Programmation Synchrone

Féaux de Lacroix Martin
22015860

Bin LIU
22116863

December 2021

1 PID contrôleur - Méthode heuristique :

Nous avons utilisé un PID pour suivre la route. Le réglage du PID consiste à trouver une combinaison de trois paramètres qui donne une réponse appropriée à une perturbation. Et les Kp, Ki et Kd respectent :

$$\begin{cases} u(t) = Kp * \left(e(t) + \frac{1}{Ti} * \int_0^t e(r)dr + Td * \frac{de(t)}{dt} \right) \\ Ki = Kp/Ti & Ti : \text{Temps d'anticipation pour estimer le comportement futur} \\ Kd = Td * Kp & Td : \text{Temps de montée pour éliminer l'erreur du système stable} \end{cases}$$

Afin de trouver les bons gains qui conviennent sur toutes les cartes, nous proposons donc une méthode heuristique selon ces nombreuses situations.

Les étapes de réglage sont principalement :

1. Réglez tous les gains (Kp, Ti et Td) à zéro.
2. Augmentez le Kp jusqu'à ce que la réponse à une perturbation soit une oscillation stable.
3. Augmenter le Td jusqu'à ce que les oscillations s'arrêtent (c'est-à-dire qu'il soit fortement amorti).
4. Répétez les étapes 2 et 3 jusqu'à ce que l'augmentation du Td n'arrête pas les oscillations.
5. Réglez Kp et Td sur les dernières valeurs stables.
6. Augmenter le gain Ti jusqu'à ce que la convergence vers le point de consigne se produise avec ou sans dépassement à un taux acceptable.

Pour l'étape 1, nous allons chercher la limite de stabilité du système (figure 1). Ce qui nous donne $Kp = 7.8$

Pour l'étape 3, nous cherchons la valeur de Td pour lequel les oscillations s'arrêtent. Ce qui nous donnent $Td = 0.44$

Après répétitions des étapes 2 et 3 sur toutes les maps, nous trouvons les valeurs suivantes $Kp = 8.8$, $Ti = 6.3$ et $Td = 0.44$. Finalement, nous utilisons la formule au _dessus et déterminons les valeurs de Kp, Ki et Kd (respectivement 8.8, 1.39 et 3.87).

2 Etapes :

Algorithme du contrôleur :

- Si l'action à terminée à l'instant précédent passer à l'étape suivante sinon récupérer l'étape courante (Init : 0)
- Sélectionne l'action courante (tourner/avancer)

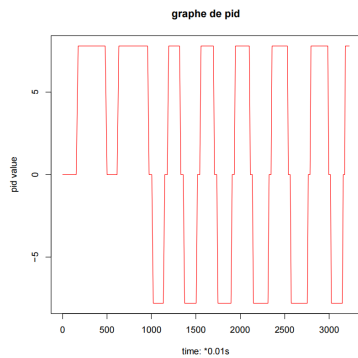


FIGURE 1 – Oscillation stable avec $K_p=7.8$, $T_i=0$, $T_d=0$

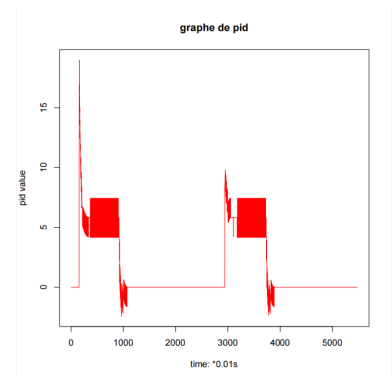


FIGURE 2 – Sans oscillation selon la map 02 avec $K_p=7.8$, $T_i=0$, $T_d=0.44$

2.1 Tourner :

Cette action se termine si la différence avec l'angle souhaité est $< 1^\circ$

2.2 Avancer :

Au cours de son trajet, la voiture peut rencontrer différentes situations qui impactent sa décision de finir son action ou non :

1. Cas où il n'existe pas de vert \rightarrow l'action ne se termine pas, nous n'avons pas croisé d'étape
2. Cas où il existe une bande verte \rightarrow l'action se termine lorsqu'on la quitte
3. Cas où il existe deux bandes vertes collées \rightarrow l'action doit toujours se terminer lorsqu'on quitte la première \rightarrow au bout de n instants (trouver en expérimentant) la voiture à dépasser la première bande, fin de l'action
4. Cas où la voiture passe par une bande verte non voulu \rightarrow particulièrement dans les intersections où l'étape vient de changer, la voiture peut dévier un peu et scanner ces bandes vertes indésirables \rightarrow ajout d'un lock qui ignore le scan de la voiture durant n instants juste après une étape et ajout d'un minimum de m instants verts pour finir l'action.

3 Feu rouge :

La détection de la couleur du feu n'est possible que sur une bande rouge. Si lorsque la voiture est sur une bande rouge et le feu n'est pas vert, la voiture s'arrête. Note : dans certaines maps, la voiture va trop vite et ne peut s'arrêter à temps.

4 Obstacle :

Un obstacle est détecté s'il est à moins de 100 de distance. Dans ce cas, la nouvelle vitesse maximale est : $vitesse_maximale \times \frac{distance_obstacle - 10}{100}$, permettant à la voiture de s'approcher de l'obstacle en ralentissant et en s'arrêtant à une distance de 10.