

# Projet de Systèmes Bouclés

## (Partie 3 : réglage du PID)

### ROBOT : le contrôleur de moteur DC

**Présentation** Au terme des parties précédentes, nous disposons d'un code phone.ino qui gère le robot via une interface web sur votre téléphone mobile. Nous sommes alors prêts à réaliser la présente partie 3 du projet qui consiste à implémenter un PID afin de réguler par une fonction d'asservissement la vitesse de rotation de chaque moteur, gauche et droit. Nous allons donc coder un correcteur de type P, puis PI et enfin PID (but final) et les régler afin obtenir la meilleure réponse possible de chaque moteur à une consigne en vitesse de rotation. L'interface web sera alors complétée avec  $2 \times 3 = 6$  sliders qui régleront les coefficients  $k_p$ ,  $k_i$  et  $k_d$  de chacun des deux PID (un pour chaque moteur). Nous obtiendrons ainsi le code final qui contient donc aussi la présente partie asservissement en vitesse du robot : asservitesse.ino.

#### Travail demandé : Partie 3

Dans un même fichier asservitesse.ino (IDE Arduino) :

1. Coder une fonction d'asservissement de type PID pour chaque moteur.
2. Interfacer ces PID avec la page web de commande en ajoutant des sliders qui permettent d'ajuster les valeurs des coefficients  $k_p$ ,  $k_i$  et  $k_d$  pour chaque PID.
3. Tester alors le code pour déterminer d'abord  $k_p$ , puis  $k_i$  et enfin  $k_d$  pour chaque moteur. Détailler les phénomènes observés et justifier en termes de performances l'ordre de grandeur choisi pour chaque coefficient, à chaque étape. Fournir obligatoirement des courbes (screenshots de l'interface web avec les courbes) explicatives avec leurs analyses.
4. Créer alors un slider final de consigne unique de vitesse du robot qui commande maintenant les deux moteurs asservis d'un seul coup.
5. Tester le robot et observer son comportement (trajectoire en ligne droite ? propriétés telles que le temps de réponse ...)

#### Conseils :

- La fonction d'asservissement introduira forcément le calcul de l'erreur, différence entre la consigne et la mesure. Ainsi que de la nouvelle commande =  $k_p \cdot \text{erreur}$  + autres termes (I et D) ... Je vous conseille éventuellement de réviser le TP5 (Asservissement de l'éclairage d'une LED) sur le PID avec la diode et le capteur photodiode, il comporte un exemple de fonction d'asservissement qui peut vous inspirer pour ce projet.
- On peut, pour le réglage du PID, éventuellement s'inspirer de la vidéo : <https://www.youtube.com/watch?v=uXnDwojRb1g>
- On peut ajouter aussi un bouton ON/OFF qui active ou pas la correction PID.
- Le fichier asservitesse.ino, but final de l'asservissement en vitesse, contient à minima forcément tout le code précédent de phone.ino !