Fiche de TP : Contrôle automatique d'un KIT ROBOT à l'aide d'arduino

Objectifs du TP : L'objectif de ce TP est de comprendre et de mettre en pratique les principes de contrôle en boucle ouverte et fermée sur un kit robot. A l'aide de la simulation sur Simulink et de l'implémentation sur une carte arduino, les étudiants mettront en œuvre un système de commande automatique-numérique.

Outils Nécessaires :

- Ordinateur avec IDEs Arduino, MATLAB et Simulink installés
- kit robot avec moteurs à courant-continu (alimentation 3 à 6 V)
- Capteur Ultrason
- Servo moteur
- Interface de commande (Arduino)
- Motor Shield compatible avec l'interface de commande (Arduino Motor Shield)
- Encodeur rotatif de vitesses angulaire

Partie 1 : Montage et prise en main des outils

1.1. Se familiariser avec la datasheet d'un shield arduino pour comprendre comment contrôler correctement les moteurs (Schéma de commande, alimentation …).

1.2. Apprendre à commander une carte arduino à partir d'un arduino motor shield (Astuce: rechercher et lire la documentation sur les motor shield)

1.3. Réaliser le montage du dispositif (kit robot + arduino + shield)

1.4. Écrire un programme **test.ino** pour faire des premiers essais sur le dispositif (faire avancer, reculer, … le robot mobile)

Partie 2 : Contrôle en Boucle Ouverte

2.1 Choix d'une consigne

2.1.1 Vitesse angulaire du mobile: ici on choisit de faire varier les vitesses angulaires du kit robot afin d'obtenir des valeurs de PWM correspondantes. Le signal PWM obtenu représente la consigne.

2.1.2 Angle de rotation du mobile: ici on choisit de faire varier les angles de rotation du kit robot afin d'obtenir des valeurs de PWM correspondantes. Le signal PWM obtenu représente la consigne.

Objectif : Étudier comment différentes consignes affectent la performance du système. Pour ce qui suit, nous avons choisi de procéder avec une consigne obtenue à partir de ***2.1.1.***

2.2 Détermination de la vitesse angulaire en fonction du PWM : modèle en boucle ouverte.

Objectif : Établir une relation entre la largeur d'impulsion modulée (PWM) et la vitesse angulaire du moteur.

Procédure :

- Écrire un programme **PWM.ino** qui permet d'envoyer des signaux PWM au moteur à partir de la carte arduino.
- Mesurer la vitesse angulaire du moteur pour différentes valeurs de PWM. (Astuce, faire une mesure de distance pour une durée déterminée et calculer la vitesse angulaire en partant de la relation qui la lie à la vitesse linéaire)
- Tracer la courbe de la vitesse angulaire en fonction du PWM.
- Analyser la linéarité de la relation observée et déterminer le modèle mathématique.

2.3 Détermination de la distance minimale de détection pour le capteur Ultrason

Objectif : Trouver la distance minimale à laquelle le capteur peut détecter un objet.

Procédure :

- Écrire un programme **détection.ino** permettant au capteur de détecter un objet à une distance déterminée. Ajouter une fonction pour arrêter les moteurs une fois que l'obstacle est détecté
- Faire varier la durée de détection (durée émission-réception) du capteur et enregistrer la distance minimale à laquelle chaque objet est détecté.
- Évaluer la précision.

Partie 3 : Contrôle en Boucle Fermée (Rétroaction)

3.1 Réalisation d'un modèle théorique des moteurs

Objectif : Choisir les paramètres optimaux pour caractériser les moteurs.

Procédure :

- Écrire un programme **moteur.m** pour tester différentes configurations de paramètres (puissances, couples…) du moteur et visualiser la courbe de PWM en fonction de la vitesse angulaire sur Simulink. (Astuce, s'inspirer d'un modèle de moteur DC-DC connu).
- Analyser la réponse du moteur à chaque configuration.
- Sélectionner la configuration qui se rapproche le mieux de la courbe obtenue de façon pratique à l'étape ***2.2***.
- La configuration obtenue à l'étape précédente représente le modèle théorique d'un des moteurs DC-DC utilisé dans le cadre dans ce TP.

3.2 Boucle de régulation : rétroaction pour ajuster le modèle en boucle ouverte à la consigne (Vitesse angulaire/PWM) qui dépend du modèle théorique.

Objectif : Optimiser le modèle de contrôle en ajustant les paramètres basés sur la rétroaction.

Procédure :

- Écrire un programme **vitesse_real.ino** qui calcul les vitesses de rotation des moteurs à l'aide des encodeurs rotatifs et affiche le résultat à l'écran .
- Écrire un programme **correcteur.ino** (Correcteur de vitesse angulaire) qui compare la consigne à la sortie obtenue à l'aide des encodeurs et ajuste la sortie pour la rapprocher à la consigne. Afficher également le résultat (sortie du système corrigé) à l'écran. (Indication, choisir une condition d'arrêt pour la boucle de correction)
- Collecter les données affichées sur le terminal dans les deux cas et tracer une courbe sur MATLAB.dans chaque cas.
- Comparer la courbe du modèle théorique des moteurs et celle obtenue à partir du correcteur.
- Ajuster les paramètres du correcteur pour obtenir de meilleurs résultats.

Partie 4 : Contrôle en Boucle Fermée (Évitement d'Obstacles en Boucle Fermée)

4.1 Modélisation du virage face à un obstacle

Objectif : Intégrer la détection d'obstacles dans le modèle de contrôle en BF.

Procédure :

- Choisir une trajectoire initiale (rectiligne).
- Écrire un programme **detect_180.ino** qui permet au capteur de détecter des obstacles dans une plage allant de 0 à 180 degré. (Astuce: se servir du servo moteur pour le balayage sur 180 degré)
- Écrire un programme **virage.ino** qui permet au robot d'éviter un obstacle en suivant un virage précis.
- Analyser différents cas possibles (obstacles à gauche et à droite par exemple).

4.2 Retour à cette trajectoire

Objectif : Permettre au robot de modifier sa trajectoire en présence d'obstacles et de revenir à la trajectoire initiale une fois l'obstacle évité. (Reprendre le code précédent et l'adapter pour remettre le robot sur sa trajectoire initiale)

Conclusion

Ce TP vise à fournir une expérience pratique dans la modélisation, le contrôle, et la simulation de systèmes robotisés, en se concentrant sur le contrôle des moteurs et l'évitement d'obstacles. Les étudiants développeront des compétences en programmation Simulink et en analyse de système de contrôle.

ANNEXE

Montage du véhicule mobile (https://www.gotronic.fr/pj2-35326-robot03-montage-1601.pdf).