TP Correction d'un système en boucle fermée :

Asservissement de la position d'un mobile sur une barre

Pour ce TP vous aurez besoin de :

- Installer l'IDE Arduino sur votre ordinateur : https://www.arduino.cc/en/software
- Télécharger les fichiers des programmes sur Chamilo : **TP_Etalonnage.ino et TP_Programme.ino**

1 - Objectif du TP

Etudier la réalisation et le comportement d'un système réel asservi en BF (maquette) et le corriger à l'aide d'un correcteur PID. Il s'agit donc de manipuler et identifier les valeurs des coefficients P, I et D pour obtenir un système réactif, précis et fiable.

Un compte rendu par groupe est à rendre à la fin de la séance de TP.

2 - Description de la maquette et préparation du matériel

Dans le cadre de ce TP, vous étudiez le système réel suivant :

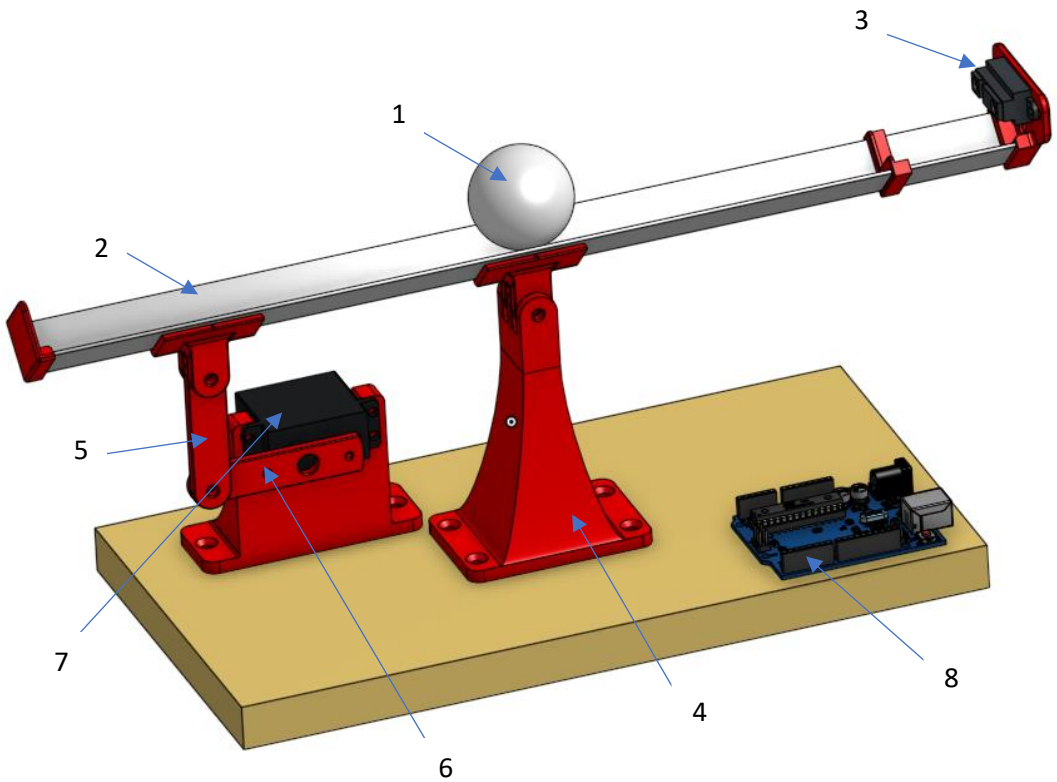

Fig. 1 : Maquette utilisée pour le TP

La figure 1 décrit la maquette utilisée pour ce TP. Soit une balle (1) de masse *m* roulant sur une cornière (2). Un capteur de distance infrarouge SHARP 2Y0A21 (3) mesure la distance entre le centre* de la balle et la face du capteur. La cornière est liée au support central (4) par une liaison pivot. L'inclinaison de la cornière est opérée par un système de bielle (5) et de manivelle (6) actionnés par un servomoteur FS5106B (7). L'ensemble du système est commandé par un microcontrôleur ATMega328 intégré sur une carte de développement de type Arduino Uno R3 (8).

Chaque maquette est équipée d'une alimentation 5V pour alimenter la carte Arduino et d'un cordon USB pour relier la carte Arduino et un ordinateur.

**La caractéristique du capteur a été déterminée pour cette maquette pour donner la position du centre de la balle et non de la surface, ainsi la distance de la balle au capteur est facilement mesurable au niveau du point de contact entre la balle et la cornière.*

Contrôle du câblage

La figure 2 schématise le câblage des composants de la maquette :

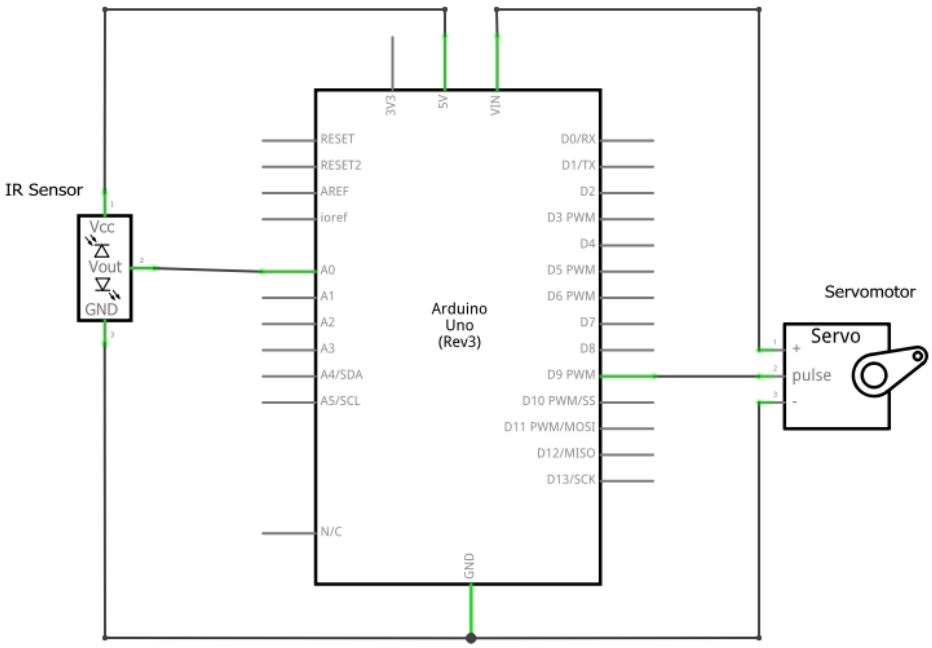

Fig. 2 : Schéma de câblage de la maquette

ATTENTION : Vérifier la conformité du câblage de votre maquette à l'aide du schéma électrique de la figure 2 AVANT TOUT BRANCHEMENT. En cas de doute appelez un enseignant.

Brancher l'alim PUIS l'usb sur l'ordinateur. Laisser brancher pendant tout le TP. Débrancher le pin 9 pour stopper la maquette.

Installation de l'IDE Arduino, des bibliothèques

Télécharger l'IDE Arduino et l'installer : https://www.arduino.cc/en/software

Installer la bibliothèque « Servo.h » :

- Démarrer l'IDE Arduino ;
- Aller dans le menu « Outils » puis « Gérer les bibliothèques » ;
- Taper « servo » dans le champs de recherche puis valider ;
- Installer la bibliothèque « Servo.h » voir figure 3 (cliquer sur le bouton installer).

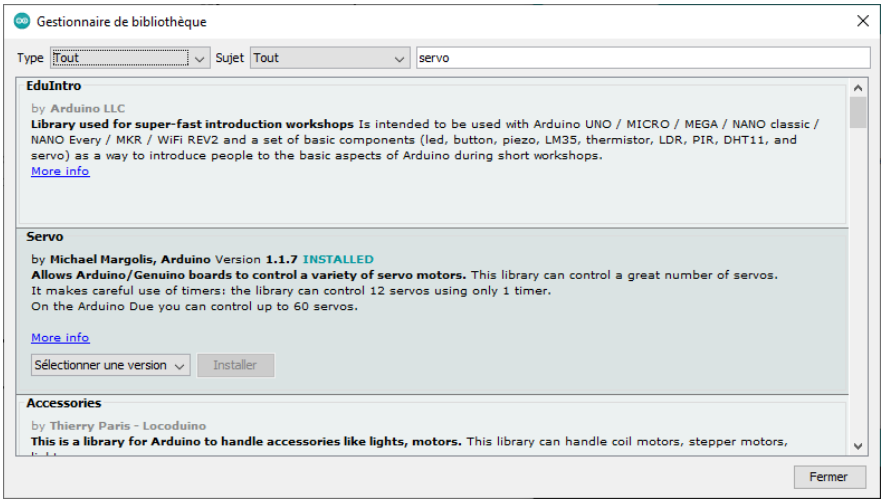

Fig. 3 : Installation de la bibliothèque « Servo.h »

3 – Modélisation du système

Q3.1 : Etablir le schéma bloc du système en y faisant apparaitre les éléments physiques du système et les grandeurs d'entrée et de sortie de chacun des blocs. Encadrer les éléments inclus dans le microcontrôleur.

On souhaite asservir la position *x* d'un mobile sphérique de masse *m* sur une barre. L'inclinaison de la barre est notée α. On suppose dans cette partie, l'actionneur (servomoteur) qui pilote l'inclinaison α infiniment rapide devant l'asservissement de la position du mobile. La figure 3 schématise le système.

On suppose α petit et les frottements négligeables. On néglige également le mouvement de rotation de la bille.

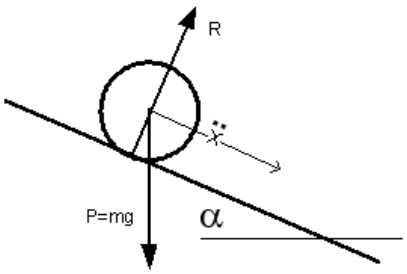

Fig. 4 : Schéma

Q3.2 : Ecrire l'équation différentielle régissant le déplacement x du mobile sur la barre en fonction de α (application du principe fondamental de la dynamique).

Q3.3 : En déduire la fonction de transfert continue F(p) entre α et x.

En supposant toujours α petit, on peut considérer que α et l'angle de sortie du servo moteur (de la manivelle) sont égaux. Nous supposerons également le capteur de distance parfait (retour unitaire).

Q3.4 : Construire le schéma bloc simplifié et calculer la fonction de transfert en boucle fermée.

Q3.5 : Que dire de la stabilité du système ? Justifier.

4 – Analyse du programme du microcontrôleur

Installer l'IDE Arduino et ouvrir le programme **TP_Programme.ino** (pour rappel, il est disponible sur Chamilo).

Etudier attentivement la structure et le contenu du programme. Un programme Arduino est écrit en langage C. Il est structuré en 3 parties :

Bloc 1 - entête du programme : Déclaration des variables globales du programme.

Bloc 2 – void setup() : Initialisation du programme, cette fonction est éxécutée une seule fois lors de la mise sous tension de la carte.

Bloc 3 – void loop() : Il s'agit de la boucle principale du programme qui s'exécute en boucle tant que la carte est sous tension

Afin de simplifier et alléger le programme, certaines fonctions peuvent être créées en dehors de la boucle void loop() et sont activées lorsqu'elles sont appelées dans void loop().

Q4.1 : Identifier la variable qui fixe la consigne.

Q4.2 : Décrire sous forme d'algorithme le fonctionnement de la boucle void loop() en omettant les fonctions d'affichage.

Q4.3 : Justifier le calcul de chacun des termes du PID dans le programme.

Q4.4 : Sur le schéma bloc de la question Q3.1, placer le correcteur PID.

5 – Etalonnage des capteurs et actionneurs et correction du système

Etalonnage

La caractéristique du capteur infrarouge de distance a été déterminée pour ce TP afin de mesurer la distance entre le centre de la balle et la face du capteur. La loi est la même pour tous les capteurs. Néanmoins, d'un capteur à l'autre il peut être nécessaire d'ajuster la précision avec un décalage (offset). Il en est de même pour les servomoteurs. Afin de déterminer les valeurs d'offset pour chacune des maquettes, suivre la procédure suivante :

- Télécharger le programme **TP_Etalonnage.ino** depuis Chamilo sur votre ordinateur.
- Ouvrir le programme dans l'IDE Arduino.
- Brancher l'alimentation PUIS la carte à l'ordinateur et au secteur à l'aide des cordons fournis.
- Télécharger le programme dans la carte Arduino.
- Observer la position de la manivelle (sortie du servomoteur), elle doit être parallèle à la base (horizontale). Si elle ne l'est pas tout à fait, modifier la valeur de la variable servoOffset de quelques degrés (positifs ou négatifs). Recharger le programme et réitérer jusqu'à trouver la bonne valeur.
- Une fois le servomoteur étalonné à l'horizontal, placer le centre de la balle à 15 cm de la face du capteur et afficher le moniteur série pour vérifier la valeur issue du capteur. Si la valeur n'est pas tout à fait exacte, modifier la valeur de la variable sensorOffset (valeur en cm). Recharger le programme et réitérer jusqu'à trouver la bonne valeur (correspondance entre valeur réelle et valeur lue sur le moniteur serie).
- Noter les valeurs trouvées pour servoOffset et sensorOffset. Elles seront à reporter dans le programme TP_Programme.ino.

Correction du système

Afin de rendre le système stable, rapide et précis, il est nécessaire de trouver les valeurs de réglage du PID, c'est-à-dire les valeurs des coefficients kp, kd et ki dans le programme.

Nous allons appliquer dans ce TP la méthode empirique pour déterminer ces coefficients.

- Ouvrir de nouveau le programme TP_Programme.ino dans l'IDE Arduino et reporter les valeurs déterminées précédemment dans les variables servoOffset et senorOffset.

Par défaut, le PID est réglé uniquement avec un gain proportionnel de 1 (kp = 1 et kd et ki sont à 0)

- Télécharger le programme dans la carte Arduino.

Q5.1 : Quel est le comportement du système ?

- Faire varier le coefficient kp de deux en deux jusqu'à 9 et observer le comportement du système.

Q5.2 : Quel est le comportement du système en fonction de la valeur de kp ?

- Trouver une valeur de Kp qui donne une oscillation entretenue rapide mais qui ne fait pas sortir la balle. Ouvrir le traceur série de l'IDE Arduino.

Q5.3 : Quelle valeur de kp trouvez-vous ? Réaliser une copie d'écran de la courbe visualisée sur le traçeur série et l'ajouter dans votre compte rendu.

- Tout en conservant la valeur de kp trouvée précédemment dans le programme, faire varier la valeur de kd de 500 en 500 jusqu'à 3000.

Q5.4 : Quelle valeur de kd donne un système stable ? Quelle est l'erreur statique ? Réaliser une copie d'écran de la courbe visualisée sur le traçeur série et l'ajouter dans votre compte rendu.

- Maintenant mettre kp = 0 et laisser la valeur trouvée précédemment pour kd. Bouger la balle entre 13 et 20 cm de la face du capteur.

Q5.5 : Expliquer le comportement du système.

L'ajout d'un offset pour le servomoteur nous a permis de corriger la géométrie du système. Nous allons maintenant rendre la géométrie du système imparfaite en mettant la valeur servoOffset = 0 (ou en introduisant volontairement quelques degrés d'erreur).

- Remettre les valeurs de kp et kd trouvées précédemment (Q5.3 et Q5.4).

Q5.6 : Quelle erreur statique mesurez-vous ? Réaliser une copie d'écran de la courbe visualisée sur le traçeur série et l'ajouter dans votre compte rendu.

- Faire varier la valeur de ki de deux en deux jusqu'à 10.

Q5.7 : Quelle valeur de ki donne un système stable et une erreur statique nulle ? Réaliser une copie d'écran de la courbe visualisée sur le traçeur série et l'ajouter dans votre compte rendu.