

Description du code bas niveau du robot principal de la Coupe de Robotique 2018 (aussi appelé *Kévin*)

Thomas Jeantet (Apache)

Résumé

Ce document présente 3 codes différents qui ont été développés pour le robot principal du club Robotik de l'UTT pour la coupe de France de Robotique 2018. Nous décrirons le code d'asservissement, développé pour Nucleo F303K8, le code de la carte actionneurs, développé pour Arduino Mega, et enfin les codes des émetteurs et récepteurs Zigbee, pour Arduino Nano.

Table des matières

I	Asservissement	3
1	Fonctionnalités	3
1.1	Positionnement	3
1.1.1	Variables de positionnement	3
1.1.2	Roues Codeuses	3
1.2	Déplacement	4
1.3	Protocole de communication	4
2	Architecture du code	4
3	Description des algorithmes	4
3.1	Fonctionnement par goals	4
3.2	Aller d'un point à un autre	4
4	Réglages	4
4.1	Décodeurs à Quadrature	4
4.2	PID(s)	4
4.3	Comportements	4
4.3.1	Fréquence d'asservissement	4
4.3.2	Vitesse max et accélération max	4
4.3.3	entraxe	4
4.3.4	Rayon des roues encodeuses	4
4.3.5	Erreurs d'angle et de positionnement	4
4.3.6	Garder le dernier goal	4
4.4	Asservissement en position ou en vitesse	4
5	Problèmes connus	4
5.1	Démarrage à chaud (hotstart)	4
5.2	Recommandations en cas de problème random, soudain et chelou	4
5.3	Compilation sous ubuntu 18.04	4

II	Carte Actionneurs	5
1	Fonctionnalités	5
1.1	Moteurs CC	5
1.2	AX12	5
1.3	Sonars	5
1.4	Moteurs Pas à Pas	5
2		5
III	Communication Zigbee	6
1	Emetteur	6
1.1	6
2	Récepteur	6
IV	Protocole de Communication	6

Remerciements

Je tiens tout d'abord à remercier l'équipe de l'UTC de 2018 pour nous avoir autorisé à utiliser leur code d'asservissement comme base. Nous le détaillerons plus loin, le code qui nous a servi de base a été téléchargé depuis le github d'UTCoupe en février 2018. Je remercie également l'équipe qui a rédigé ce code, celle d'UTCoupe 2013.

Je voudrais remercier Willy Maine, enseignant à l'UTT, qui a su guider la conception mécanique en un très court délai. Je remercie les courageux qui ont passé 10 semaines à bosser sur le code du robot tous les jours quasiment sans interruption, à savoir Alexis et Quentin. Sans vous, faire marcher ce code aurait été beaucoup plus lent et difficile pour le moral.

Enfin, je tiens à remercier toute l'équipe de Robotik de cette année pour le travail accompli, à savoir Baptiste, Samuel, Nina, Clara, Clement-Joseph et Romain. Grâce à vous notre club a réalisé un score qu'il n'avait pas atteint depuis de nombreuses années, espérons qu'il en continuera ainsi.

Première partie

Asservissement

Le programme d'asservissement permet d'avoir une interface simple du point de vue de l'intelligence artificielle pour contrôler le déplacement du robot. Il mémorise et met à jour la position du robot, et commande les moteurs en fonction des ordres reçus via le protocole de communication. Ce code s'exécute sur la carte d'asservissement, la STM32. Dans cette partie, nous aborderons de manière exhaustive les fonctionnalités du programme, son architecture, les algorithmes de positionnement et de déplacement, la méthode pour régler l'asservissement et enfin les problèmes que nous avons pu découvrir mais qui n'ont pas été corrigés.

1 Fonctionnalités

1.1 Positionnement

Le positionnement est crucial pour le déplacement. En effet, si le robot ne connaît pas sa position, il est incapable de calculer la trajectoire à prendre pour atteindre un point.

1.1.1 Variables de positionnement

Pour caractériser la position de notre robot sur la table, on utilise 3 variables : Les positions cartésiennes du robot, x et y , mais aussi son orientation, θ , l'angle entre une direction de référence et la direction du robot. Le choix de la direction de cette référence peut être arbitraire, et ne dépend pas du code de l'asservissement. C'est dans la logique de l'intelligence artificielle que cette référence est décidée. Nous verrons plus tard comment l'intelligence artificielle communique avec l'asservissement. Dorénavant, quand nous parlerons de la position du robot, cela désignera les 3 variables (x, y, θ) .

1.1.2 Roues Codeuses

Notre asservissement utilise un positionnement pas roues codeuses. Ce sont les fines roues aux extrémités gauche et droite du robot. Le capteur qui y est fixé envoie une information à la carte d'asservissement à chaque rotation significative de la roue. Cette information permet de savoir de quelle distance chaque roue avance. Ainsi, en connaissant la position cartésienne du robot et son orientation, on peut utiliser l'information des roues codeuses pour mettre à jour ces 3 paramètres de positionnement.

1.2 Déplacement

1.3 Protocole de communication

2 Architecture du code

3 Description des algorithmes

3.1 Fonctionnement par goals

3.2 Aller d'un point à un autre

4 Réglages

4.1 Décodeurs à Quadrature

4.2 PID(s)

4.3 Comportements

4.3.1 Fréquence d'asservissement

4.3.2 Vitesse max et accélération max

4.3.3 entraxe

4.3.4 Rayon des roues encodeuses

4.3.5 Erreurs d'angle et de positionnement

4.3.6 Garder le dernier goal

4.4 Asservissement en position ou en vitesse

5 Problèmes connus

5.1 Démarage à chaud (hotstart)

5.2 Recommandations en cas de problème random, soudain et chelou

Généralement ces problèmes ne sont pas dûs au code, mais à une modification extérieure. Dans ce cas il y a une check-list :

- Reset la STM
- nettoyer les roues motrices et codeuses
- vérifier les branchements des codeuses
- vérifier la fixation des codeuses (les vis doivent être serrées mais pas trop pour que les codeuses suspendues puissent monter et descendre)

5.3 Compilation sous ubuntu 18.04

Deuxième partie

Carte Actionneurs

1 Fonctionnalités

1.1 Moteurs CC

1.2 AX12

1.3 Sonars

1.4 Moteurs Pas à Pas

2

Troisième partie

Communication Zigbee

1 Emetteur

1.1

2 Récepteur

Quatrième partie

Protocole de Communication