

الجمهورية الجزائرية الديمقراطية الشعبية
وزارة التعليم العالي و البحث العلمي

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE SCIENTIFIQUE

جامعة سعد دحلب البليدة
Université SAAD DAHLEB de BLIDA



Faculté des sciences de l'ingénieur
Département d'électronique

كلية الهندسة
دائرة الإلكترونيك

Mémoire de projet de fin d'études

Pour l'obtention du diplôme de MASTER en électronique
option automatique

Implémentation d'un contrôleur PI pour la commande du déplacement d'un robot mobile

Présenté par :

TSALAMLAL Mohamed Yacine

BOULENOUAR Bilel

Proposé par :

Mr. B.KAZED

Session : Septembre 2011

REMERCIEMENTS

*Nous tenons, avant tout, à remercier **ALLAH**, tout clément, tout puissant, de nous avoir donné la force de réaliser ce travail.*

*Nous remercions notre promoteur, le **D^r B. Kazed** pour nous avoir permis ainsi qu'à de nombreux étudiants de prendre part à un concours de robotique aussi motivant que passionnant et pour nous avoir proposé de réaliser notre travail dans ce cadre attrayant. Nous le remercions, également, pour ses nombreux conseils et son aide avisée.*

*Nous exprimons notre profonde reconnaissance aux membres du **jury** pour avoir accepté de juger notre travail.*

Que tous nos professeurs qui ont contribué à notre formation trouvent ici notre plus profonde gratitude.

*Nous adressons nos plus sincères remerciements à tous ceux qui ont contribué, de près ou de loin, à l'aboutissement de ce travail notamment notre ami **A. Longo** pour nous avoir aidé à réaliser la partie mécanique à notre chère ami **M.Derbal** notre ami **M.Achour** de l'équipe de robotique **Area16** pour nous avoir prêté du matériel sans oublier **M^rA.Saidani** du comité d'organisation Eurobot Algérie.*

Enfin, nous souhaitons dédier ce mémoire à nos collègues, amis et familles. Mais avant tout à nos parents, rien n'aurait été possible sans leur soutien, confiance et générosité.

REMERCIEMENTS

Résumés

Liste des figures

Liste des tableaux

Sommaire

Introduction générale..... 1

Chapitre I Généralités sur les robots

I.1.Introduction	2
I.2.Principe de fonctionnement d'un robot.....	3
I.3.La robotique mobile	4
I.4.Problèmes en robotique mobile	5
I.5.Le concours Eurobot	6
I.6.Objectifs du projet	7
I.7.Choix de l'architecture globale du robot.....	8
I.7.1.Classes de robots à roues	8
I.7.1.1.Robot unicycle	8
I.7.1.2.Robot tricycle	9
I.7.1.3.Robot voiture	10
I.7.1.4.Robot omnidirectionnel	10
I.7.2 Comparaison entre les différents types de robots	11
I.7.3 Modélisation d'un robots unicycle	12
I.7.3.1.Commande	12
I.7.3.2.Localisation	12
I.8.Choix de la méthode de contrôle	13
1.8.1 Définition du problème	13
I.8.1.1.Problème principal	13
I.8.2.Choix de la solution	14
I.8.2.1.Suite de rotations et lignes droites	14
I.8.2.2.Asservissement sur la position finale	14
I.8.2.3.Suivi de chemin	15
1.8.2.4 Stabilisation sur une trajectoire avec profil de vitesse :	15
I.8.2.5.Approche retenue	16
I.9.Résumé du travail	16
I.10.Conclusion	17

Chapitre II

application de la théorie du PID sur le modèle du moteur à courant continu

II.1 Introduction	18
II.2 Généralités sur les systèmes asservis	18
II.2.1 Principe général de la régulation	18
II.2.2 Système en boucle ouverte	19
II.2.3 Système en boucle fermée	19
II.2.4 Notion de performance	20
II.2.4.1 Précision	20
II.2.4.2 La rapidité	22
II.2.4.3 Stabilité	22
II.2.5 Influence des perturbations	23
II.2.6 Compensation des systèmes asservis	24
II.3 Modélisation du moteur à courant	25
II.3.1 Description d'un moteur à courant continu	25
II.3.2 Model physique du moteur a courant continu	26
II.3.3 Fonction de transfert du moteur	27
II.4 L'application de la commande PID sur le model du moteur à courant continu	31
II.4.1 Asservissements et influence des coefficients	32
II.4.1.1 Asservissement P	32
II.4.1.2 Asservissement PI	33
II.4.1.3 Asservissement PID	34
II.5 Réglage des paramètres	36
II.5.1 Méthode de Ziegler & Nichols	36
II.5.1.1 Méthode de la courbe de réaction (Première méthode)	36
II.5.1.2 Méthode d'oscillation (Seconde méthode)	38
II.5.2 Application de la méthode de Ziegler-Nichols pour un asservissement en vitesse du moteur Maxon(110160)	39
II.5.2.1 Premier méthode	39
II.5.2.2 Deuxième méthode	41
II.5.3 Méthode avancée	42
II.5.4 Limitations et approximations	43

II.5.4.1 Limitation de la tension de commande	43
II.6 Conclusion	44

Chapitre III

Mise en œuvre de la commande PID d'un moteur a courant continu

III.1 Introduction	45
III.2 Système à commander	45
III.3 Calcul de la vitesse et de la position	46
III.3.1 Lecture de l'encodeur	46
III.3.2 mesure de vitesse et calcul de position	47
III.3.3 Mesure de la position et calcule de vitesse	49
III.4 la carte de Puissance	51
III.4.1 Le pont H	51
III.4.2 Principe de fonctionnement du pont H	52
III.4.3 Les deux principaux modes de commande du pont H	53
III.4.3.1 Le mode Sign – Magnitude	53
III.4.3.2 Le mode Locked Anti-phase	54
III.4.3.3 Comparaison entre les deux modes	55
III.4.4 Pilotage d'un moteur par PWM	55
III.4.4.1 Principe de la PWM	55
III.4.4.2 Modes de la PWM	57
III.4.4.3 Choix de la fréquence de la PWM	58
III.4.5 Génération de la PWM	59
III.5 Communication	60
III.6 Implémentation de l'asservissement	60
III.6.1 Choix de la fréquence d'échantillonnage	61
III.6.2 Calculs et approximations	62
III.6.2.1 Calcul approximé de la dérivée	62
III.6.2.2 Calcul approximé de l'intégrale	62
III.6.2.3 Calcul du PID	63
III.7 Mise au point de l'asservissement – Résultats	63
III.8 Conclusion	67

Chapitre IV

Mise en œuvre du système de locomotion

IV.1 Introduction	68
IV.2 Conception de la base Robotique	69
IV.2.1 Objectifs	69
IV.2.2 Partie mécanique	70
IV.2.3 Partie électronique	72
IV.2.3.1 Puissance	72
IV.2.3.2 Logique	74
IV.2.3.3 Le microcontrôleur	74
IV.2.3.4 Les capteurs	75
IV.2.3.5 Les encodeurs en quadrature	76
IV.2.3.6 Courant consommé	76
IV.2.3.7 Mesure de proximité et de distance	76
IV.2.3.8 Isolation galvanique	78
IV.2.3.9 Afficheur LCD	79
IV.2.3.10 L'alimentation	79
IV.2.3.11 Résultat	79
IV.3 Mise en œuvre du control	79
IV.3.1 Démarche	79
IV.3.2 Système à commander	80
IV.3.2 .1 Approche multivariable	80
IV.3.2 .2 Modèle physique	80
IV.3.3 Principe de la commande	84
IV.3.3.1 Synthèse des régulateurs	87
IV.3.4 Odométrie	89
IV.3.5 Implémentation sur le microcontrôleur	92
IV.3.5.1 Contraintes et solution	93
IV.4 Résultats et évaluation des performances	94
IV.4.1 Résultats	94

IV.4.1.1 Manipulation rotation symétrique	94
IV.4.1.2 Manipulation ligne droite	98
IV.4.2 Évaluation des performances	101
IV.4.2.1 La Partie mécanique	101
IV.4.2.2 La partie électronique	102
IV.4.2.3 Partie contrôle	102
IV.4.2.4 Localisation par odométrie	103
IV.5 Conclusion	103
Conclusion générale	105
Bibliographie	106
ANNEXES	

Résumé :

Dans le domaine de la robotique mobile, l'étude de la locomotion possède une place prépondérante. De nombreuses approches et solutions peuvent être envisagées et il convient d'apporter un soin particulier quant à leur sélection afin de garantir les performances du système final. Ce travail de fin d'études a pour objectif la conception complète d'un système de locomotion pour robot mobile. La réalisation de la partie mécanique et électronique, ainsi que la synthèse des lois de commande sont étudiées en détails. Ces lois seront principalement divisées en deux phases : d'une part, un contrôle sur l'orientation et la distance du robot avec la génération d'un profil de vitesse, et d'autre part, un contrôle local de la vitesse des roues motrices. La commande est basée sur un correcteur PID classique. L'Odométrie est quand elle est utilisée pour l'estimation de la position du robot mobile.

Ce travail se termine, enfin, par une implémentation pratique du système étudié et par une évaluation de ses performances.

Mots clés : Robot mobile, Déplacement, Localisation, Asservissement, Régulateur PID, Odométrie.

Abstract

In the field of mobile robotics, the study of locomotion has a prominent place. Many approaches and solutions are possible and we should provide a special care for their selection to ensure the performance of the final system. Our final year project aims at designing a complete locomotion system for a mobile robot. The designs of the mechanical and electronic parts, as well as the synthesis of control laws are studied in detail. These laws are mainly divided into two phases: first, control over the direction and distance of the robot with the generating of a velocity profile and on the other hand, local control of the speed of the drive wheels. This control laws are based on a classical PID algorithm. Odometry is used to estimate the position of the mobile robot.

Finally, the work ends with a practical implementation of the studied system and the evaluation of its performances.

Key words: Mobile robot, Position estimation, Motion control, PID controller, Odometry.

INTRODUCTION GENERALE

La technologie moderne a permis le développement des sciences tout en imposant l'exploration de domaines théoriques de plus en plus complexes. Parmi ces sciences en pleine expansion et intégrant rapidement l'apport des technologies modernes, on compte l'automatique. Le substantif « automatique » a été utilisé pour la première fois en 1914 dans un article « Essai sur l'Automatique » publié dans une revue scientifique.

De nos jours, l'automatique fait partie des sciences de l'ingénieur. Cette discipline traite de la modélisation, de l'analyse, de la commande et de la régulation des systèmes dynamiques. Elle a pour fondements théoriques les mathématiques, la théorie du signal et l'informatique.

Les robots mobiles à roues se trouvent parmi les applications qui suscitent beaucoup d'engouement chez les automaticiens. Malgré leur simplicité apparente (mécanismes plans, à actionneurs linéaires) ces systèmes ont soulevé un grand nombre de problèmes difficiles. Nombre de ceux-ci ne sont d'ailleurs toujours pas résolus. Ainsi, alors que les robots manipulateurs se sont aujourd'hui généralisés dans l'industrie, rares sont les applications industrielles qui utilisent des robots mobiles. Si l'on a vu depuis peu apparaître quelques produits manufacturiers (chariots guidés) ou grand public (aspirateur).

La gamme des problèmes potentiellement soulevés par le plus simple des robots mobiles à roues en fait un sujet d'étude à part entière et forme une excellente base pour l'étude de systèmes mobiles plus complexes.

Ce travail a pour objectif la conception complète du mécanisme de locomotion d'un robot participant à un concours de robotique mobile. La réalisation est la plus générale possible afin de pouvoir être appliquée à d'autres robots mobiles se déplaçant sur une surface plane. En effet le concours n'étant l'objectif finale en soit mais plutôt une possibilité d'appliquer et de tester notre travail en situation réelle d'autant plus que la compétition s'avère une source énorme de motivation.

Ce projet présente deux aspects distincts. Premièrement, il s'agit de développer une base robotique destinée à tester des algorithmes de commande. Il était donc principalement question d'électronique et de mécanique. Le second aspect du projet est plus tourné vers la théorie, il s'agit de faire la synthèse des lois de commandes basées sur le régulateur PID pour le contrôle du déplacement robot, et de l'utilisation de l'odométrie pour sa localisation. Finalement l'implémentation pratique permet d'évaluer les performances du système.

Chapitre I

Généralités sur les robots

I.1.Introduction

L'origine du mot robot est issu du tchèque "robota" qui signifie travail forcé. Le terme de robotique est apparu en 1942 dans le cycle universellement connu rédigé par Isaac Asimov et intitulé "Les robots". Un robot est une machine programmable qui imite des actions d'une créature intelligente. Isaac Asimov a écrit les 3 lois fondamentales de la robotique. Celles-ci permettent d'éviter que cette science amène l'apocalypse avec elle :

- **Première loi:** Un robot ne peut porter atteinte à un être humain, ni en restant passif, laisser cet être humain exposé au danger.
- **Deuxième loi:** Un robot doit obéir aux ordres donnés par les êtres humains, sauf si de tels ordres sont en contradiction avec la première loi.
- **Troisième loi:** Un robot doit protéger son existence dans la mesure où cette protection n'est pas en contradiction avec la première ou la deuxième loi.

Les robots prennent actuellement une place importante dans notre vie. On les trouve dans toutes les entreprises pour accélérer la production ou pour agir là où l'homme ne peut travailler à cause du danger, à la maison pour aider à la cuisine et aux tâches ménagères, à la guerre, dans les services publics: hôpitaux, casernes de pompiers, la police... etc.

La robotique est devenue, en quelques années, une science importante qui ne cesse d'évoluer. Les chercheurs parviennent petit à petit à donner à des machines une intelligence artificielle. Les robots envahissent littéralement notre vie, on les trouve dans des concours, dans les films de science-fiction et dans plein d'autres domaines.

Au cours de l'histoire, 3 types différents de robots sont apparus, attestant chacun de nouvelles évolutions:

Le premier type de machine que l'on peut appeler robot correspond aux "Automates". Ceux-ci sont généralement programmés à l'avance et permettent d'effectuer des actions répétitives.

Le second type de robot correspond à ceux qui sont équipés de capteurs (les sens du robot). On trouve tout type de capteurs (température, photo électronique, à ultrasons) pour par exemple éviter les obstacles et/ou suivre une trajectoire. Ces capteurs vont permettre au robot une relative adaptation à son environnement afin de prendre en compte des paramètres

aléatoires qui n'auraient pu être envisagés lors de leur programmation initiale. Ces robots sont donc bien plus autonomes que les automates.

Enfin, le dernier type de robot existant correspond à ceux disposant d'une intelligence dite "artificielle" et reposant sur des modèles mathématiques complexes tels que les réseaux de neurones. En plus de capteurs physiques comme leurs prédécesseurs, ces robots peuvent prendre des décisions beaucoup plus complexes et s'appuient également sur un apprentissage de leurs erreurs comme peut le faire l'être humain. Bien sûr il faudra attendre encore longtemps avant que le plus "intelligent" des robots ne soit égal, tant par sa faculté d'adaptation que par sa prise de décisions, à l'Homme.

I.2.Principe de fonctionnement d'un robot

Un robot, est une machine qui va évoluer en fonction de son environnement, soit de manière répétitive en effectuant une tâche programmée sans se tromper, soit de manière pseudo intelligente en évoluant dans un environnement aléatoire. Cela signifie qu'il va réagir en fonction de diverses contraintes qui lui seront soumises comme des obstacles, un objet, un trou, un changement de luminosité...etc. Face à ces contraintes, le robot va devoir réagir, mais avant, il faut aussi qu'il reconnaisse la contrainte. Pour faire une approche plus précise, regardons généralement la façon de fonctionner d'un être humain :

L'être humain vit, se déplace et évolue dans un environnement. Il est capable de reconnaître, de se déplacer sans tomber où se cogner, de communiquer...etc. Si l'homme est capable de répondre aux exigences que l'environnement lui soumet, c'est parce qu'il est capable de les sentir, de les voir. Exemple : l'être humain est, par exemple, capable de repérer un mur par la vision, de sentir une odeur de gaz par le nez ou repérer un sol instable par le toucher. Toutes les informations que l'homme va découvrir sur son environnement, vont permettre de le faire évoluer. Il saura qu'en voyant un mur il ne pourra pas passer. S'il est capable de réagir, c'est parce que les informations captées de l'environnement ont été analysées par le cerveau et celui-ci en a ressorti une solution. Cette dernière a été mise en application à travers le corps: un déplacement avec les jambes, un son avec la bouche, un geste de la main...etc.

Globalement, l'homme reçoit des informations par ses différents sens, ces informations sont analysées par le cerveau qui envoie la solution à appliquer aux muscles.

Pour un robot, c'est la même chose. La perception des sens est résolue par des capteurs. Ces capteurs peuvent être de tout type, du simple interrupteur à la caméra. Toutes les informations en provenance de ces capteurs sont envoyées à un « cerveau » qui peut-être constitué d'un montage électronique ou d'un microcontrôleur programmable (voire un pc). Ce cerveau va analyser les informations reçues et va envoyer la réaction à prendre à un module permettant de contrôler les différentes actions qui agissent sur l'environnement, comme des moteurs, des bruits, des lumières...etc. Le robot est dès lors capable de réagir, selon sa programmation et ses capacités de mouvement, à un environnement aléatoire ou d'exercer une tâche répétitive [1].

I.3.La robotique mobile

De manière générale, on regroupe sous l'appellation robots mobiles l'ensemble des robots à base mobile, par opposition notamment aux robots manipulateurs. L'usage veut néanmoins que l'on désigne le plus souvent par ce terme les robots mobiles à roues.

Les autres robots mobiles sont, en effet, le plus souvent désignés par leur type de locomotion, qu'ils soient marcheurs, sous-marins ou aériens. On peut estimer que les robots mobiles à roues constituent le gros des robots mobiles. Historiquement, leur étude est venue assez tôt, suivant celle des robots manipulateurs, au milieu des années 70. Leur faible complexité en a fait de bons premiers sujets d'étude pour les roboticiens intéressés par les systèmes autonomes. Cependant, malgré leur simplicité apparente (mécanismes plans, à actionneurs linéaires), ces systèmes ont soulevé un grand nombre de problèmes difficiles. Nombre de ceux-ci ne sont d'ailleurs toujours pas résolus. Ainsi, alors que les robots manipulateurs se sont aujourd'hui généralisés dans l'industrie, rares sont les applications industrielles qui utilisent des robots mobiles. Si l'on a vu depuis peu apparaître quelques produits manufacturiers (chariots guidés) ou grand public (aspirateur),



Figure I.1 convoyeur automatisés AGV

L'industrialisation de ces systèmes bute sur divers problèmes délicats. Ceux-ci viennent essentiellement du fait que, contrairement aux robots manipulateurs prévus pour travailler exclusivement dans des espaces connus et de manière répétitive, les robots mobiles sont destinés à évoluer de manière autonome dans des environnements peu ou pas structurés. Néanmoins, l'intérêt indéniable de la robotique mobile est d'avoir permis d'augmenter considérablement nos connaissances sur la localisation et la navigation de systèmes autonomes.

La gamme des problèmes potentiellement soulevé par le plus simple des robots mobiles à roues en fait un sujet d'étude à part entière et forme une excellente base pour l'étude de systèmes mobiles plus complexes.

I.4.Problèmes en robotique mobile

On distingue sans trop d'ambiguïté un certain nombre de problèmes en robotique mobile. Bien évidemment, l'aspect matériel, qui consiste à choisir et dimensionner aussi bien la structure mécanique du système que sa motorisation, son alimentation et l'architecture informatique de son système de contrôle-commande apparait comme le premier point à traiter.

Le choix de la structure est souvent effectué parmi un panel de solutions connues et pour lesquelles on a déjà résolu les problèmes de modélisation, planification et commande.

Le choix des actionneurs et de leur alimentation est généralement assez traditionnel. La plupart des robots mobiles sont ainsi actionnés par des moteurs électriques à courant continu alimentés par des convertisseurs de puissance fonctionnant sur batterie.

De la même façon, les architectures de contrôle-commande des robots mobiles ne sont pas différentes de celles des systèmes automatiques ou robotiques plus classiques. On y distingue cependant, dans le cas général, deux niveaux de spécialisation, propres aux systèmes autonomes: une couche décisionnelle, qui a en charge la planification et la gestion (séquentielle, temporelle) des événements et une couche fonctionnelle, chargée de la génération en temps réel des commandes des actionneurs. Bien évidemment, l'architecture du robot dépend fortement de l'offre et des choix technologiques du moment. Même si le novice en robotique croit parfois que tout l'Art de la discipline consiste à fabriquer le robot le plus beau et le plus rapide, est forcé est de constater que cette étape, certes nécessaire, n'est pas au centre des préoccupations de la robotique mobile. Les problèmes spécifiques à la robotique mobile n'apparaissent finalement que lorsque l'on dispose d'une structure mobile dont on sait actionner les roues. Tous les efforts du roboticien vont alors consister à mettre en place les outils permettant de faire évoluer le robot dans son environnement de manière satisfaisante, qu'il s'agisse de suivre un chemin connu ou au contraire d'aller d'un point à un autre en réagissant à une modification de l'environnement ou à la présence d'un obstacle [2].

1.5.Le concours Eurobot

Eurobot est un concours de robotique amateur européen rassemblant principalement des équipes d'étudiants et des clubs indépendants. Le concours est constitué de qualifications nationales puis d'une finale internationale accueillant plus de 27 pays.

Chaque match oppose deux robots autonomes se déplaçant sur une surface de trois mètres sur deux pendant une durée de quatre-vingt-dix secondes. Le Vainqueur étant le robot qui marque le plus de points. Le thème définissant le comptage des points change chaque année [3].

La participation à ce concours est un projet pluridisciplinaire de grande envergure. Il donne la possibilité d'appliquer et de tester notre travail en situation réelle et a donc été une énorme source de motivation.

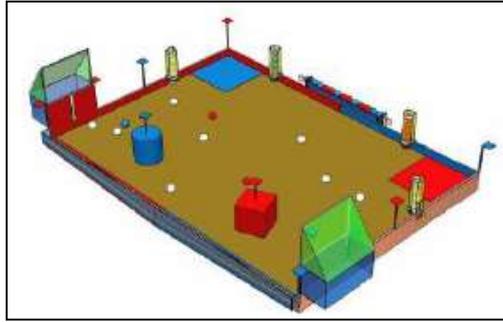


Figure I.2 Table Eurobot

I.6.Objectifs du projet

Dans le domaine de la robotique mobile, l'étude de la locomotion a, bien entendu, une place prépondérante. De nombreuses approches et solutions peuvent être envisagées et il convient d'apporter un soin particulier quant à leur sélection afin de garantir les performances du système final.

Ce travail s'inscrit dans le cadre d'une participation à un concours de robotique mobile et a pour objectif la conception complète du mécanisme de locomotion du robot participant. La réalisation se devra d'être la plus générale possible afin de pouvoir être appliquée à d'autres robots mobiles se déplaçant sur une surface plane.

L'objectif final sera la possibilité de déplacer le robot d'une position A à une position B tout en restant dans un périmètre fixe, et ceci, de la manière la plus rapide et la plus précise possible.

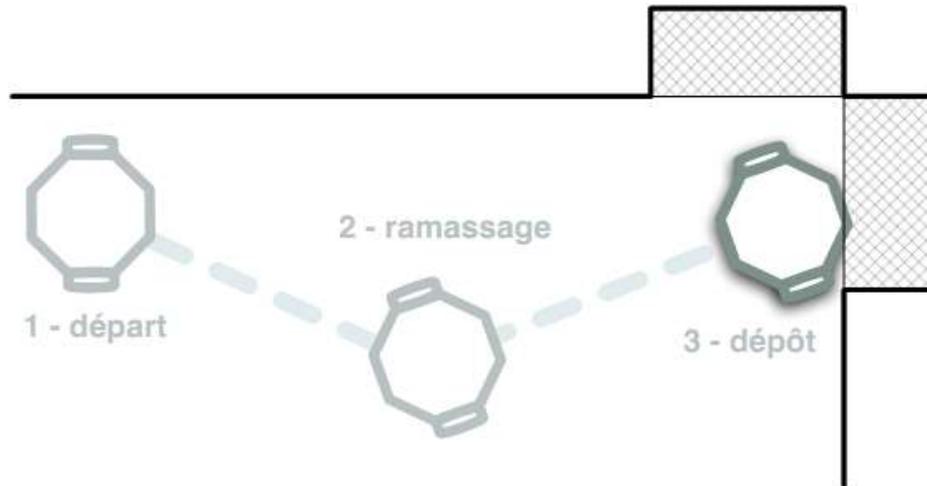


Figure I.3 déplacement en ligne droite entre 3 positions

On entend par conception complète de la locomotion, la réalisation de la partie mécanique et de la partie électronique et de la partie contrôle de la base robotique.

I.7.Choix de l'architecture globale du robot

I.7.1.Classes de robots à roues

Il existe plusieurs classes de robots à roues déterminées, principalement, par la position et le nombre de roues utilisées. Nous citerons ici les quatre classes principales de robots à roues [2] :

I.7.1.1.Robot unicycle

Un robot de type unicycle est actionné par deux roues indépendantes, il possède éventuellement des roues folles pour assurer sa stabilité. Son centre de rotation est situé sur l'axe reliant les deux roues motrices. C'est un robot non-holonyme¹, en effet il est impossible de le déplacer dans une direction perpendiculaire aux roues de locomotion. Sa commande

¹ Non holonome signifie que le système comporte une contrainte non intégrable (i.e. le système ne peut pas effectuer certains mouvements).

peut être très simple, il est, effectivement, assez facile de le déplacer d'un point à un autre par une suite de rotations simples et de lignes droites.

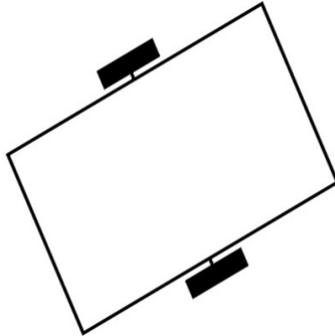


Figure I.4 Robot de type unicycle

I.7.1.2. Robot tricycle

Un robot de type tricycle est constitué de deux roues fixes placées sur un même axe et d'une roue centrée orientable placée sur l'axe longitudinal.

Le mouvement du robot est donné par la vitesse des deux roues fixes et par l'orientation de la roue orientable. Son centre de rotation est situé à l'intersection de l'axe contenant les roues fixes et de l'axe de la roue orientable.

C'est aussi un robot non-holonome. Sa commande est plus compliquée. Il est en général impossible d'effectuer des rotations simples à cause d'un rayon de braquage limité de la roue orientable.

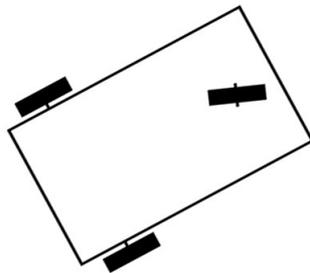


Figure I.5 Robot de type tricycle

I.7.1.3. Robot voiture

Un robot de type voiture est semblable au tricycle, il est constitué de deux roues fixes placées sur un même axe et de deux roues centrées orientables placées elles aussi sur un même axe.

Le robot de type voiture est cependant plus stable puisqu'il possède un point d'appui supplémentaire.

Toutes les autres propriétés du robot voiture sont identiques au robot tricycle, le deuxième pouvant être ramené au premier en remplaçant les deux roues avant par une seule placée au centre de l'axe, et ceci de manière à laisser le centre de rotation inchangé.

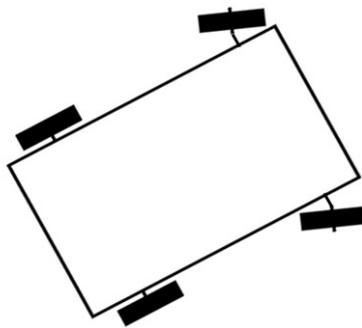


Figure I.6 Robot de type voiture

I.7.1.4. Robot omnidirectionnel

Un robot omnidirectionnel est un robot qui peut se déplacer librement dans toutes les directions. Il est en général constitué de trois roues décentrées orientables placées en triangle équilatéral.

L'énorme avantage du robot omnidirectionnel est qu'il est holonome puisqu'il peut se déplacer dans toutes les directions. Mais ceci se fait au dépend d'une complexité mécanique bien plus grande.

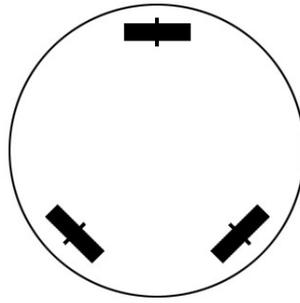


Figure I.7 Robot de type omnidirectionnel

I.7.2 Comparaison entre les différents types de robots

Nous pouvons observer dans le tableau ci-dessous un récapitulatif des avantages et des inconvénients des différents types de robots à roues.

Tableau I.1 comparaison des différents types de robot a roues

	Avantages	inconvénients
Robot unicycle	<ul style="list-style-type: none"> - stable - rotation sur soi-même - complexité mécanique faible 	<ul style="list-style-type: none"> - non-holonyme
Robot tricycle	<ul style="list-style-type: none"> - Complexité mécanique modérée 	<ul style="list-style-type: none"> - non-holonyme - peu stable - pas de rotation sur soi-même
Robot voiture	<ul style="list-style-type: none"> - Stable - complexité mécanique modérée 	<ul style="list-style-type: none"> - non-holonyme - pas de rotation sur soi-même
Robot omnidirectionnel	<ul style="list-style-type: none"> - holonyme - stable - rotation sur soi-même 	<ul style="list-style-type: none"> - complexité mécanique importante

Notre choix: Pour la résolution de notre problème, nous allons nous concentrer sur l'étude des robots à roues unicycle. Ceux-ci sont, en effet, particulièrement adaptés à un environnement plan tel que celui rencontré dans le concours qui nous intéresse. Ils ont une construction mécanique relativement simple et jouissent d'une certaine contrôlabilité.

I.7.3 Modélisation d'un robots unicycle

Comme nous l'avons décrit précédemment, les robots unicycles sont une classe de robots à roues. Ils représentent un modèle de robot mobile relativement simple et qui est, de ce fait, largement utilisé en pratique. Nous allons décrire ici plus en détail les propriétés de commande et de localisation particulières du robot unicycle.

I.7.3.1. Commande

Dans des conditions de roulement sans glissement², les vitesses d'avance et de rotation du robot sont liées aux vitesses de rotation angulaire des deux roues motrices (Ω_{droite} et Ω_{gauche}) par les relations suivantes :

Pour la vitesse longitudinale v du robot , on a :

$$v = \frac{v_{droite} + v_{gauche}}{2} = \frac{r(\Omega_{gauche} - \Omega_{droite})}{2} \quad (\text{I. 1})$$

Pour la vitesse de rotation du robot ω on a :

$$\omega = \dot{\theta} = \frac{v_{droite} - v_{gauche}}{L} = -\frac{r(\Omega_{gauche} + \Omega_{droite})}{L} \quad (\text{I. 2})$$

Avec r le rayon d'une roue et L la distance entre les deux roues motrices.

I.7.3.2. Localisation

Dans des conditions de roulement sans glissement, il est possible de déterminer la position du robot à l'instant t connaissant sa position à l'instant t_0 .

En effet, on peut écrire :

$$\omega = \dot{\theta} \quad (\text{I.3})$$

$$\dot{x} = v \cos \theta \quad (\text{I.4})$$

$$\dot{y} = v \sin \theta \quad (\text{I.5})$$

² On dit qu'un corps est en roulement sans glissement (r.s.g.) sur une surface lorsque la vitesse relative du corps par rapport à cette surface au point de contact est nulle. (i.e. le corps reste en contact avec la surface sans glisser)

On pourra donc déterminer la position du robot à l'instant t par intégration :

$$\theta(t) = \int_0^t \dot{\theta}(\tau) d\tau + \theta_0 = \int_0^t \omega(\tau) d\tau + \theta_0 \quad (\text{I.7})$$

$$x(t) = \int_0^t \dot{x}(\tau) d\tau + x_0 = \int_0^t v(\tau) \cos \theta(\tau) d\tau + x_0 \quad (\text{I.8})$$

$$y(t) = \int_0^t \dot{y}(\tau) d\tau + y_0 = \int_0^t v(\tau) \sin \theta(\tau) d\tau + y_0 \quad (\text{I.9})$$

L'utilisation de capteurs (encodeurs) sur les moteurs ou sur les roues, permet donc de déterminer la position d'un robot unicycle à tout instant, à la condition de connaître sa position initiale. Cette méthode est connue sous l'appellation d'Odométrie.

I.8.Choix de la méthode de contrôle

1.8.1 Définition du problème

Avant d'établir les lois de contrôle qui vont être utilisées, il convient de définir de manière détaillée les fonctionnalités attendues.

Remarque: Dans toute l'étude qui va suivre, nous considérerons seulement la classe de robots à roues de type unicycle, celle-ci étant choisie comme architecture globale de notre robot.

I.8.1.1.Problème principal

Soit le robot dans une certaine configuration³ donnée A ($v_A, \omega_A=0, \theta_A, x_A, y_A$), et soit une configuration désirée B ($v \leq v_{BMAX}, \omega_B=0, \theta_B, x_B, y_B$). Trouver une loi de contrôle permettant de passer de la configuration A à la configuration B.

Contraintes:

- Le robot doit rester dans un certain périmètre fixe ;
- Le dérapage des roues doit être évité (pour rester en roulement sans glissement) ;
- Le temps d'exécution doit être minimisé ;
- Les déplacements doivent pouvoir être enchainés.

³ Par configuration, on entend la valeur de l'ensemble des paramètres du robot : sa vitesse d'avance v , sa vitesse de rotation ω , son orientation, sa position en x et y .

Remarquons que la vitesse de rotation aux points A et B est imposée nulle, cette règle est peu contraignante et a été fixée pour faciliter la possibilité d'enchaînement des déplacements, le point B pouvant ainsi être facilement réutilisé comme nouveau point A'.

Notons aussi que la vitesse d'avance du robot au point B n'est pas imposée, seule une limite maximale est fixée. En effet, vu le périmètre limité pour effectuer le déplacement, il n'existera pas toujours de solution nous permettant d'atteindre une vitesse imposée au point B (l'accélération du robot n'étant pas infinie).

Ces choix nous permettent de limiter les contraintes sur les trajectoires suivies et nous assurent de toujours trouver une solution au problème posé si, et seulement si, il existe à partir du point A une trajectoire de freinage permettant au robot de s'immobiliser dans la limite du périmètre imposé.

I.8.2.Choix de la solution

Maintenant que les bases du problème ont été posées, il reste à déterminer la méthode que nous allons utiliser pour sa résolution.

Quatre approches ont été envisagées:

I.8.2.1.Suite de rotations et lignes droites

Comme nous l'avons vu à la section I.7.2.1, un robot unicycle peut avoir une commande très simple constituée d'une suite de rotations simples et de lignes droites. Cette solution est souvent utilisée en pratique mais possède un défaut : à chaque transition rotation \leftrightarrow ligne droite, nous sommes obligés de passer par une position de repos ($v = 0, \omega = 0$). Nous serons donc souvent un peu loin de l'optimum quant à la minimisation du temps d'exécution.

I.8.2.2.Asservissement sur la position finale

Une autre solution souvent utilisée consiste à asservir la vitesse d'avance et la vitesse de rotation du robot sur la position en x et y de l'objectif. La vitesse d'avance du robot étant régulée par rapport à la distance entre la position courante du robot et la position finale, la vitesse de rotation étant, quant à elle, régulée par rapport à l'angle entre la direction courante du robot et la direction relative de l'objectif. Lorsque la position finale en x et y est atteinte, on effectue éventuellement une rotation simple afin d'atteindre la position angulaire désirée. Cette solution nous permet d'atteindre des vitesses plus grandes, la trajectoire suivie étant une courbe, et donc un temps d'exécution en général bien meilleur que pour la solution précédente.

L'énorme défaut de cette approche est que nous n'avons aucune connaissance à priori sur la trajectoire qui sera empruntée par le robot, nous n'avons donc, par exemple, aucune garantie que celui-ci reste dans son périmètre imposé.

I.8.2.3.Suivi de chemin

Afin de garantir le chemin emprunté par le robot, la solution est bien évidemment d'effectuer l'asservissement sur ce même chemin. C'est un problème non trivial pour lequel plusieurs approches sont proposées.

Un des défauts de cette approche est qu'elle ne fait aucune hypothèse sur la trajectoire à suivre. Or dans le problème qui nous préoccupe, si nous voulons rester en roulement sans glissement, la trajectoire doit respecter certaines propriétés de courbures et les accélérations des roues doivent être maîtrisées. Cette approche ne permet pas de tenir compte de ces particularités [2].

I.8.2.4 Stabilisation sur une trajectoire avec profil de vitesse :

Afin de rester en roulement sans glissement, nous fixons des contraintes d'accélération et de vitesse sur chacune des roues du robot. Afin de respecter ces contraintes, un profil de vitesse devra être calculé en même temps que la trajectoire.

Ce profil, consistant à donner à tout endroit de la trajectoire la vitesse de chacune des roues du robot, pourra être utilisé lors du contrôle. En effet, il peut être vu comme la définition d'un robot virtuel de référence. Le problème de contrôle pourra, dès lors, être ramené au problème de stabilisation du robot mobile par rapport à ce robot de référence.

En théorie cette approche permet de résoudre notre problème en respectant toutes les contraintes et ceci avec un temps de parcours proche de l'optimum. Mais elle reste de loin la plus difficile à mettre en œuvre. En effet, cette méthode repose sur des algorithmes complexes, son implémentation est gourmande en ressources et nécessite dans le cas du concours la disposition d'un ordinateur embarqué. Les calculs étant trop lourds pour tourner sur un microcontrôleur [2].

I.8.2.5.Approche retenue

Pour la suite de ce travail nous avons adopté l'approche mettant en œuvre une suite de rotations et lignes droites avec profil de vitesse. En effet, c'est celle qui va permettre de résoudre notre problème en respectant toutes les contraintes, mis à part le temps de parcours qui n'est pas optimum, mais ce point peut être amélioré en mettant le robot dans ses limites d'accélération acceptables.

Le control consiste globalement en un asservissement à deux niveaux :

- un asservissement local PID en vitesse de chaque moteur ;
- un asservissement PID sur l'orientation et l'avance du robot.

I.9.Résumé du travail

Pour mettre en œuvre la méthode de contrôle choisie dans la section précédente, plusieurs étapes sont à suivre :

La première est de faire la synthèse des régulateurs, il est nécessaire de passer par l'étude théorique de la commande PID appliquée au modèle physique du moteur à courant continu avec des simulations sous le logiciel Matlab.

L'étape suivante consiste en la réalisation d'une carte électronique de contrôle pour un moteur à courant continu. Celle-ci passe par la réalisation d'un étage de puissance constitué principalement d'un pont H et par la sélection de ses composants. Le choix du mode et de la fréquence de pilotage du pont H doivent être discuté.

Nous entamons, ensuite, la conception de la partie logique avec le choix d'un microcontrôleur et des différents capteurs nécessaires au fonctionnement du robot. Elle doit, par exemple être capable d'effectuer des mesures de position et de vitesse du robot ou encore de détecter une situation dangereuse pour le moteur par une mesure du courant consommé.

Une fois la carte électronique réalisée et testée, nous en venons à l'étape d'implémentation du contrôle. Tout d'abord, le contrôle de bas niveau PID est codé dans le microcontrôleur et permet de contrôler localement en vitesse un moteur. Ensuite, un programme de contrôle PID de l'avance et de l'orientation du robot est implémenté sur le même microcontrôleur. S'ajoute à cela le calcul odométrique qui permet d'estimer les coordonnées du robot.

Enfin pour pouvoir régler l'asservissement et évaluer les performances du système, il a été nécessaire d'établir une communication entre le PC et le microcontrôleur pour récupérer l'historique d'évolution de diverses variables et d'en tracer la représentation graphique après adaptation des données brutes en un format plus exploitable.

I.10.Conclusion

L'objectif à atteindre étant maintenant défini et le plan de travail tracé, les prochains chapitres seront consacrés non seulement à l'étude détaillée et la mise en œuvre des solutions choisies, mais aussi à faire l'évaluation des performances du système ainsi que la proposition d'une liste pour des améliorations futures.

Chapitre II

Application de la théorie du PID sur le modèle du moteur à courant continu

II.1 Introduction

Dans ce chapitre, nous allons introduire les principes de base sur les systèmes asservis ainsi que la régulation en boucle ouverte et en boucle fermée. On va essayer ensuite de comprendre les principes de la commande PID et de l'appliquer sur le modèle physique du moteur à courant continu.

II.2 Généralités sur les systèmes asservis

II.2.1 Principe général de la régulation

Dans la plupart des appareils dans les installations industrielles et domestiques, il est nécessaire de maintenir des grandeurs physiques à des valeurs déterminées, en dépit des variations externes ou internes influant sur ces grandeurs. Par exemple, le niveau d'eau dans un réservoir, la température d'une étuve, la vitesse et la position des moteurs, étant par nature variables, doivent donc être réglés par des actions convenables sur le processus considéré. Si les perturbations influant sur la grandeur à contrôler sont lentes ou négligeables, un simple réglage dit en boucle ouverte, permet d'obtenir et de maintenir la valeur demandée (par exemple : action sur un robinet d'eau). Dans la majorité des cas, cependant, ce type de réglage n'est pas suffisant, parce que trop grossier ou instable. Il faut alors comparer, en permanence, la valeur mesurée de la grandeur réglée à celle que l'on souhaite obtenir et agir en conséquence sur la grandeur d'action, dite grandeur régulée. On a, dans ce cas, constitué une boucle de régulation et plus généralement une boucle d'asservissement. Cette boucle nécessite la mise en œuvre d'un ensemble de moyens de mesure, de traitement de signal ou de calcul, d'amplification et de commande d'actionneur, constituant une chaîne de régulation ou d'asservissement. La consigne est maintenue constante et il se produit sur le procédé une modification d'une des entrées perturbatrices. L'aspect régulation est considéré comme le plus important dans le milieu industriel, car les valeurs des consignes sont souvent fixes. Néanmoins, pour tester les performances et la qualité d'une boucle de régulation, on s'intéresse à l'aspect asservissement qui consiste à suivre rapidement les variations de la consigne.

II.2.2 Système en boucle ouverte

Un système est dit en boucle ouverte lorsque la commande est élaborée sans l'aide de la connaissance des grandeurs de sortie. Au niveau des inconvénients, il n'y a aucun moyen de contrôler, à plus forte raison de compenser les erreurs, les dérives, les accidents qui peuvent intervenir à l'intérieur de la boucle, autrement dit, il n'y a pas de précision ni surtout de fidélité qui dépendent de la qualité intrinsèque des composants. Enfin, le système en boucle ouverte ne compense pas les signaux de perturbation, le schéma bloc de système en boucle ouverte est donné par la figure ci-dessous [4].

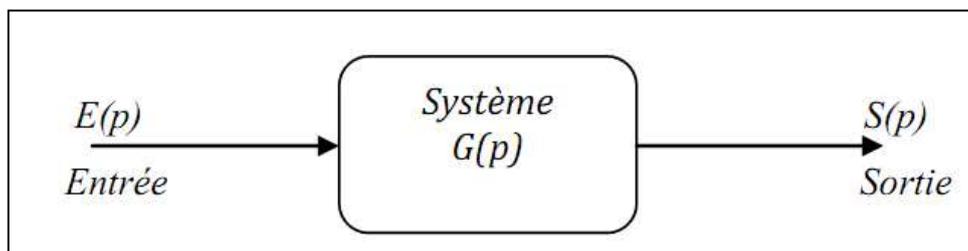


Figure II.1 Schéma bloc d'un système en boucle ouverte

Dans ce dernier cas, le comportement du processus est décrit par la relation :

$$S(p) = G(p) * E(p) \quad (\text{II.1})$$

S : grandeur réglée (sortie)

E : grandeur réglante (entrée)

$G(p)$: fonction de transfert

II.2.3 Système en boucle fermée

La boucle fermée (contre réaction) est capable de stabiliser un système instable en boucle ouverte. Dans une régulation en boucle fermée, une bonne partie des facteurs perturbateurs externes sont automatiquement compensés par la contre-réaction à travers le procédé. L'utilisation du retour d'information est le principe fondamental en automatique. La commande appliquée au système est élaborée en fonction de la consigne et de la sortie. La (Figure II.2) représente le principe de retour unitaire [4].

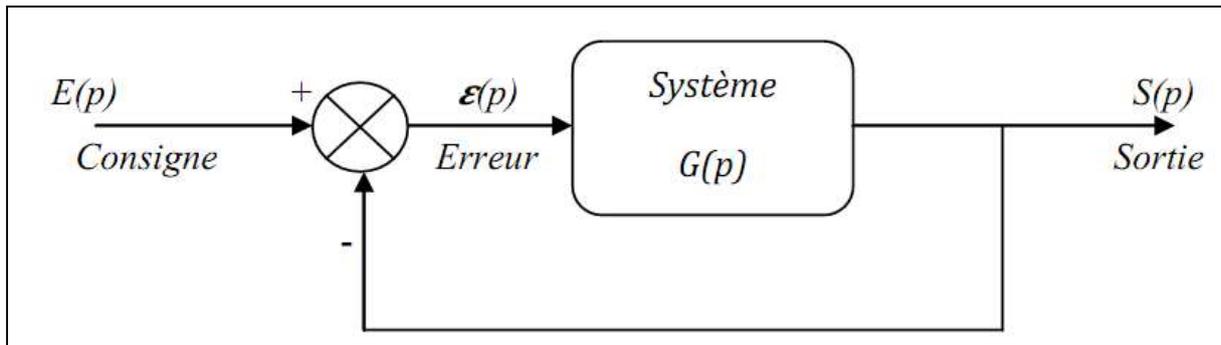


Figure II.2 Schéma bloc d'un système en boucle fermée avec un retour unitaire

$$H_{BF}(P) = \frac{G(P)}{1+G(p)} \quad (\text{II.2})$$

H : fonction de transfert en boucle fermée

E : grandeur réglante (consigne)

S : grandeur réglée e : erreur = $E(p) - S(p)$

II.2.4 Notion de performance

Un système asservi performant doit satisfaire à trois critères [5] :

- Précision ;
- Rapidité ;
- Stabilité.

II.2.4.1 Précision

Un système asservi en boucle fermée est dit précis, si sa sortie $s(t)$ est proche de la consigne (valeur désirée) $e(t)$.

On peut représenter l'erreur entre la consigne et la sortie. Dans ce cas, on s'intéresse à :

$$\lim_{t \rightarrow \infty} \varepsilon(t) = \lim_{p \rightarrow 0} p (E(p) - S(p)) \quad (\text{II.3})$$

Cette erreur dépend de l'entrée et du gain de la fonction de transfert, car l'augmentation du gain permet d'avoir une meilleure précision.

La précision est définie principalement par deux grandeurs :

➤ **Erreur statique**

Le système est soumis à une entrée d'amplitude constante $e(t)=E_0$: c'est la consigne en échelon. La réponse du système atteint une valeur stable au bout d'un certain temps : c'est le régime permanent. On mesure alors l'écart entre la consigne (valeur souhaitée) et la valeur atteinte en régime permanent.

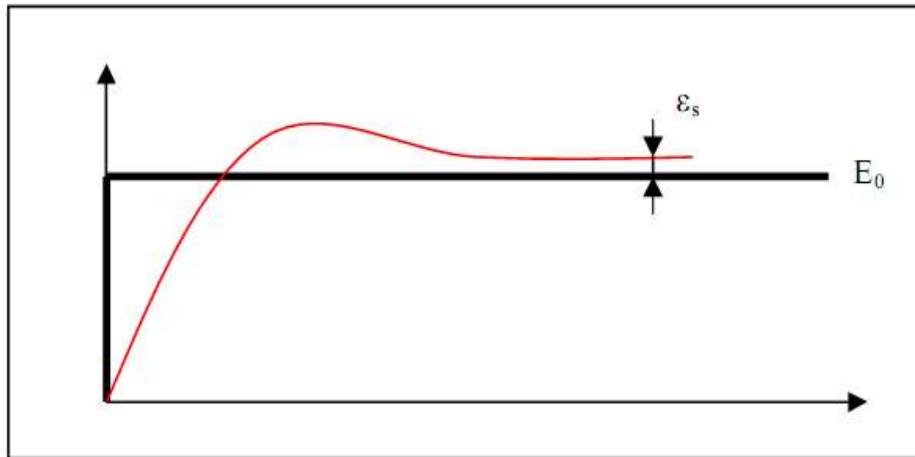


Figure II.3 Système possédant une erreur statique.

➤ **Erreur dynamique**

Le système est, cette fois, soumis à une entrée d'amplitude variable sous forme d'une droite de pente a : $e(t) = a.t$. C'est une consigne rampe de même que dans le cas d'entrée en échelon, on mesure l'écart en régime permanent qui est alors ϵ_v .

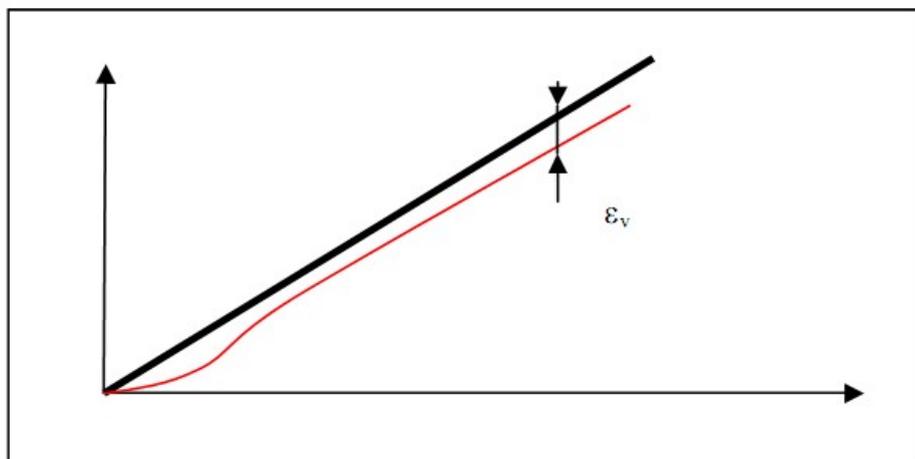


Figure II.4 Système à erreur de trainage.

II.2.4.2 La rapidité

Elle est caractérisée par le temps de réponse (en générale 5%). On mesure ou on calcule le temps nécessaire pour que la réponse du système ne s'écarte plus de plus de 5% de la valeur finale.

Le système est quelque peu précis. La valeur finale est assez proche de la valeur de consigne, et on peut alors considérer que le temps de réponse à 5% est le temps au bout duquel la réponse est confinée en une bande de largeur 5% autour de la consigne. Cette approximation est un peu rapide mais souvent acceptable.

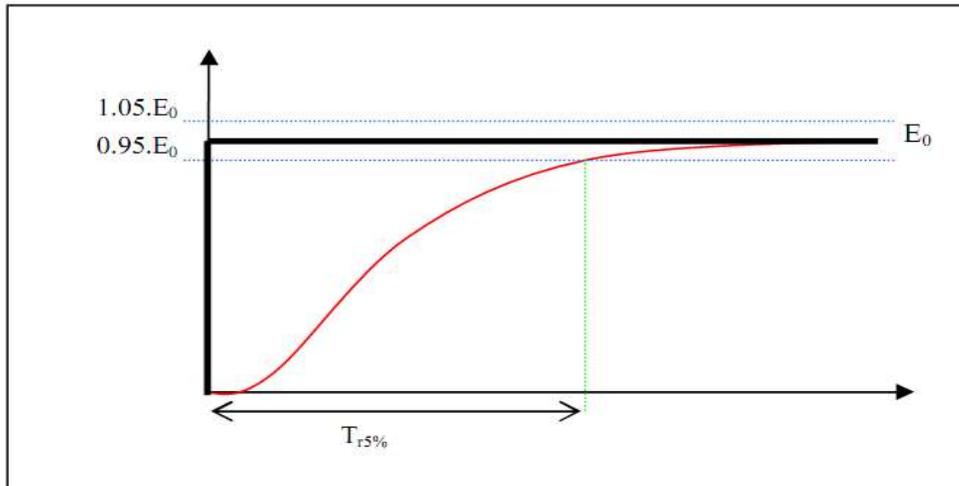


Figure II.5 Temps de réponse a 5%.

II.2.4.3 Stabilité

Dans le cas des systèmes linéaires représentés par une fonction de transfert, l'analyse des pôles permet de conclure sur la stabilité du système. On rappelle que, si les pôles de la fonction de transfert $G(p)$ sont P_0, P_1, \dots, P_n . Dans le cas d'une fonction de transfert continue utilisant la transformée de Laplace, tous les pôles doivent être à partie réelle strictement négative pour que le système soit stable .

Pour : $i = 0 : n$

$$P_i = Re_i + j.Im_i \quad (II.4)$$

P_i : Pôles de la fonction de transfert

Re_i : Partie réelle de P_i

Im_i : Partie imaginaire de P_i

On définira la stabilité par une des propositions suivantes : Un système linéaire est stable :

- Lorsque sa réponse à un échelon prend une valeur finie en régime Permanent.
- Lorsque sa réponse à une impulsion tend vers 0.
- Lorsque sa réponse à une sinusoïde est une sinusoïde d'amplitude finie.

II.2.5 Influence des perturbations

Une perturbation est une entrée supplémentaire au système qu'on ne peut contrôler ou maîtriser. Ces perturbations ont une influence sur l'asservissement des systèmes, voir la (figure II.6).

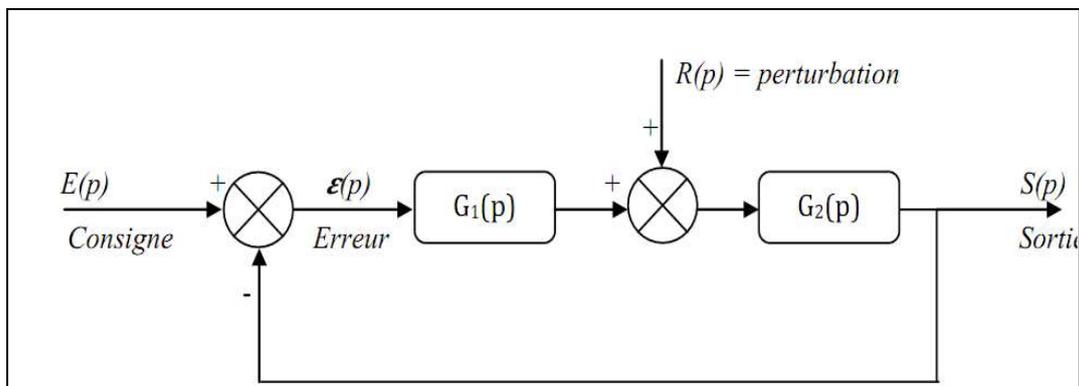


Figure II.6: Schéma bloc d'un asservissement avec une perturbation (Retour unitaire)

La fonction de transfert du bloc d'asservissement est représentée comme suit : Calculons $S(p)$ en fonction de $R(p)$, (En posant $E(p) = 0$) :

$$S_R(p) = \frac{G_2(p)}{1+G_1(p)G_2(p)} R(p) \quad (\text{II.5})$$

Calculons $S(p)$ en fonction de $E(p)$, (En posant $R(p) = 0$) :

$$S_E(p) = \frac{G_1(p)G_2(p)}{1+G_1(p)G_2(p)} E(p) \quad (\text{II.6})$$

La sortie totale est :

$$S(p) = \frac{G_2(p)}{1+G_1(p)G_2(p)}R(p) + \frac{G_1(p)G_2(p)}{1+G_1(p)G_2(p)}E(p) \quad (\text{II.7})$$

II.2.6 Compensation des systèmes asservis

La plupart des processus ont besoin de correcteurs pour compenser et améliorer la précision et la stabilité. Un correcteur est un système qui va élaborer la commande d'un système en fonction de l'erreur mesurée entre la sortie et la consigne. Si on prend le cas d'un correcteur proportionnel qui est un système qui donne une commande proportionnelle à l'erreur mesurée. Beaucoup de systèmes peuvent être commandés par ce type de correcteur, qui est simple à mettre en œuvre. Le principe est l'ajustement du gain qui va consister à obtenir un bon compromis entre la stabilité et la précision [6].

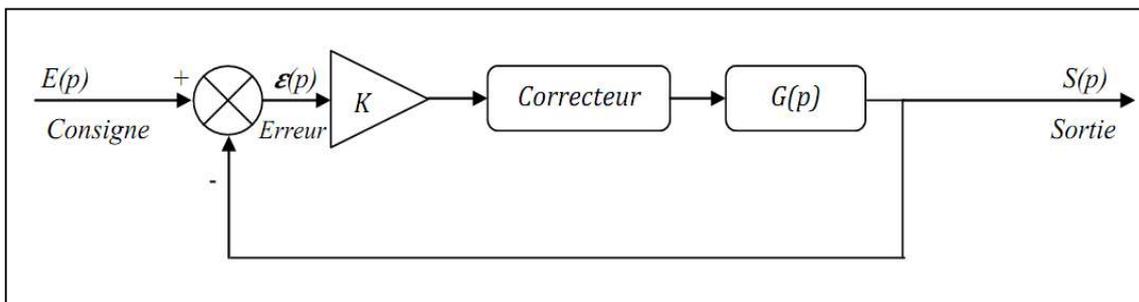


Figure II.7 Schéma bloc d'un système avec correcteur (Retour unitaire)

Ce genre de correcteur n'est pas toujours possible ou suffisant. Des correcteurs plus sophistiqués peuvent permettre de :

- Stabiliser un système instable ;
- Augmenter le degré de la stabilité sans réduire le gain K ;
- Réduire ou annuler les erreurs statiques sans toucher à la stabilité.

II.3 Modélisation du moteur à courant

II.3.1 Description d'un moteur à courant continu

Un moteur à courant continu est une machine électrique. Il s'agit d'un convertisseur électromécanique permettant la conversion bidirectionnelle d'énergie entre une installation électrique parcourue par un courant continu et un dispositif mécanique, d'où l'énergie électrique est transformée en énergie mécanique. Un moteur électrique à courant continu est constitué :

- D'un stator qui est à l'origine de la circulation d'un flux magnétique radial fixe créé soit par des enroulements statoriques (bobinage), soit par des aimants permanents. A l'arrière du stator, se trouve la partie porte balais et les balais assurant les contacts électriques avec le rotor. Il est aussi appelé inducteur.
- D'un rotor bobiné relié à un collecteur rotatif inversant la polarité dans chaque enroulement rotorique au moins une fois par tour de façon à faire circuler un flux magnétique transversal en quadrature avec le flux statorique. Les enroulements rotoriques sont aussi appelés enroulements d'induits, ou communément induit [7].

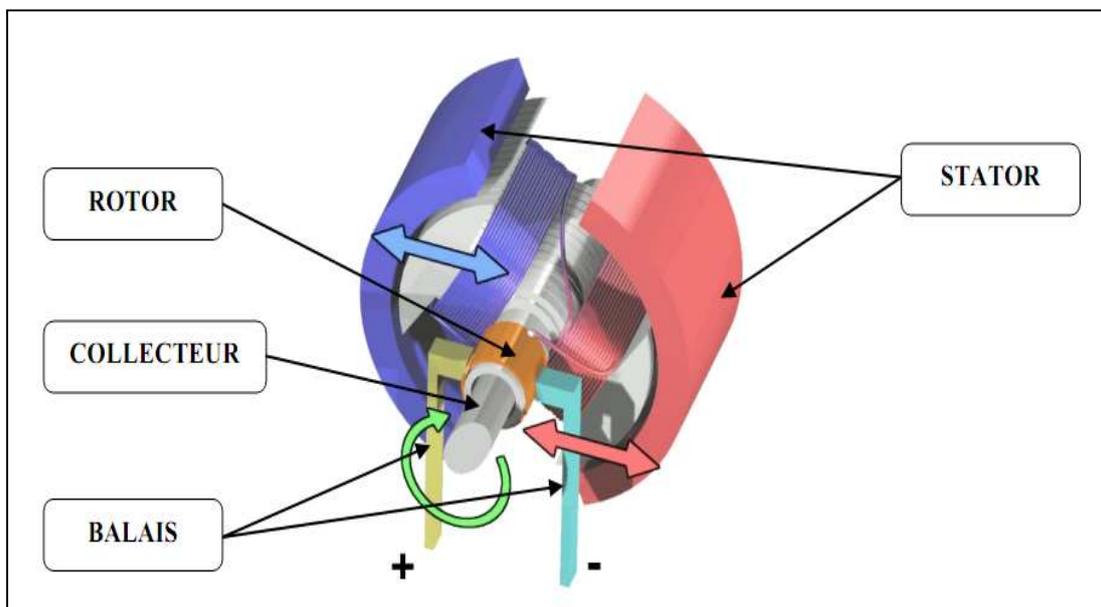


Figure II.8 Schéma d'un moteur à courant continu

II.3.2 Model physique du moteur a courant continu

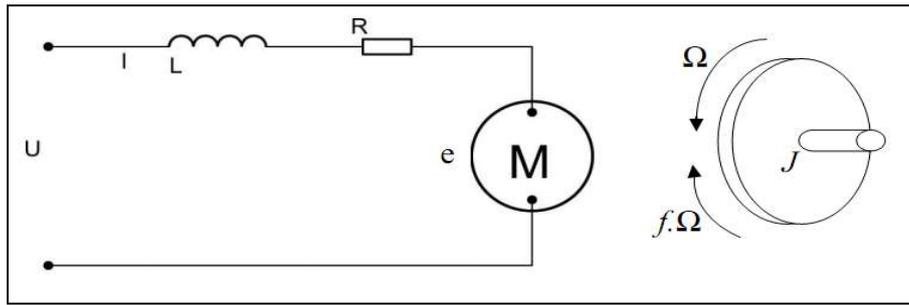


Figure II.9 Schéma équivalent d'un moteur à courant continu

Selon le schéma de la (figure II.9), un moteur électrique à courant continu est régi par les équations physiques découlant de ses caractéristiques électriques, mécaniques et magnétiques. D'après la loi de Newton, combiné à des lois de Kirchhoff, On peut écrire les équations différentielles de premiers ordres suivantes :

$$u(t) = Ri(t) + L \frac{di(t)}{dt} + e(t) \tag{II.8}$$

$$e(t) = K_e \Omega(t) \tag{II.9}$$

D'après le principe fondamental de la dynamique on a :

$$J \frac{d\Omega}{dt} = C_u - C_r \tag{II.10}$$

$$C_u = K_c i(t) - C_p \tag{II.11}$$

$$C_r = f \Omega(t) \tag{II.12}$$

$u(t)$: Tension appliquée au moteur

$e(t)$: Force contre électromotrice

$i(t)$: Intensité traversant le moteur

$\Omega(t)$: Vitesse de rotation du rotor

f : Coefficient de frottement visqueux

K_c : Constante de couple

C_p : Couple de pertes

C_u : Couple moteur généré

C_r : Couple résistant

K_e : Constante de vitesse

J : Moment d'inertie de l'axe du rotor

II.3.3 Fonction de transfert du moteur

On passe dans le domaine de Laplace

$$Ri(p) + Li(p) + E = U(p) \quad (\text{II.13})$$

$$E = K_e \Omega(p) \quad (\text{II.14})$$

$$Jp\Omega(p) = C_u - C_r \quad (\text{II.15})$$

En combinant (II.13) et (II.14) on obtient :

$$U(p) = Ri(p) + Lpi(p) + K_e \Omega(p) \quad (\text{II.16})$$

En modifiant (II.15) on a :

$$Jp\Omega(p) = K_c i(p) - C_p - f \Omega(p) \quad (\text{II.17})$$

On en déduit l'expression de $\Omega(p)$

$$\Omega(p) = \frac{K_c i(p) - C_p}{f + Jp} \quad (\text{II.18})$$

On peut en sortir l'expression de $i(p)$:

$$I(p) = \frac{f + Jp}{K_c} \left(\Omega(p) + \frac{C_p}{f + Jp} \right) \quad (\text{II.19})$$

On l'injecte à présent dans (II.13)

$$U(p) = \Omega(p) \left(\frac{(R + Lp)(f + Jp)}{K_c} + K_e \right) + \frac{R + Lp}{f + Jp} C_p \quad (\text{II.20})$$

On suppose que le moment du couple de pertes (qui est vu comme une perturbation) est négligeable devant le moment du couple électromagnétique ($K_c i(t)$) on peut alors prendre C_p nul pour simplifier le système.

On a donc :

$$U(p) = \Omega(p) \left(\frac{(R+Lp)(f+Jp)}{K_c} + K_e \right) \quad (\text{II.21})$$

La fonction de transfert recherchée $H(p)$ est entre la tension entrant dans le moteur $U(p)$ et la vitesse de sortie $\Omega(p)$ [8],[9].

$$H(p) = \frac{\Omega(p)}{U(p)} = \frac{K_c}{(R+Lp)(f+Jp)+K_e K_c} \quad (\text{II.22})$$

Il faut noter que si le moteur est équipé de réducteur mécanique on doit ajouter le moment d'inertie de ce dernier pour pouvoir modéliser tout le système tel que :

$$J = J_m + J_r \quad (\text{II.23})$$

J_m : Moment d'inertie rapporté au rotor du moteur

J_r : Moment d'inertie rapporté à l'axe du réducteur

Aussi la vitesse rapportée sur l'axe du réducteur est donnée par :

$$\Omega_r(p) = \Omega(p) \cdot \mathit{ratio} \quad (\text{II.24})$$

$\Omega_r(p)$: vitesse rapportée sur l'axe du réducteur

ratio : Rapport de réduction

Les valeurs numériques du MCC choisi (*Maxon: Référence 110160*) données par le Constructeur (Tableau II.1).

Tableau II.1 Valeurs numériques d'un moteur à courant continu

Tension nominale	$U_{nom} = 12 \text{ V}$
Courant permanent maximal	$I_{max} = 2.02 \text{ A}$
Résistance aux bornes	$R = 5.74\Omega$
Constante de couple	$K_c = 10.9 \text{ mNm.A}^{-1}$
Constance de vitesse	$K_e = 10.9 \text{ mNm.A}^{-1}$
Inductance	$L = 363 \text{ mH}$
Moment d'inertie rapporté au rotor	$J=4.26 \text{ gm.cm}^2$
Coefficient de frottement visqueux	$f = 0$

On peut établir le modèle mathématique de la réponse en vitesse du moteur électrique qui est donné par la figure suivante (figure II.10):

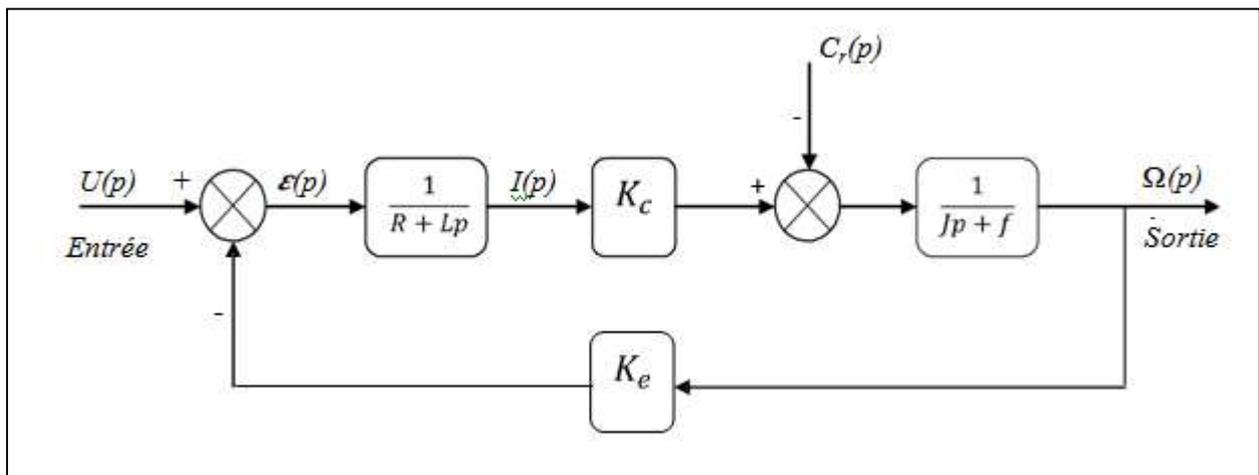


Figure II.10 Schéma bloc du Modèle de moteur électrique en vitesse

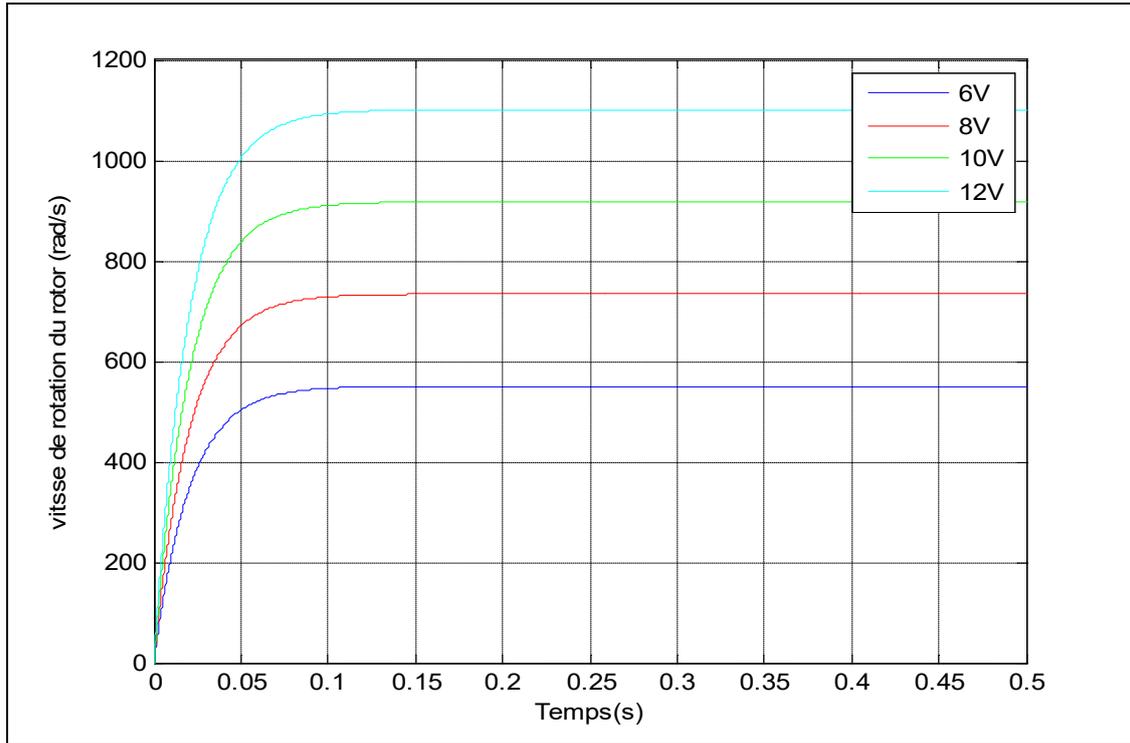


Figure II.11 Modélisation sous Matlab de la réponse de vitesse

D’après l’équation (II.22), le système modélisé est du second ordre, lorsque l’inductance interne est négligeable devant la résistance interne (ce qui est généralement le cas) il s’apparente à un système du premier ordre. On observe bien sur la (figure. II.11) ci-dessus que le moteur change de vitesse de rotation pour chaque valeur de la tension d’entrée, d’où la vitesse de rotation est proportionnelle à la tension d’entrée.

On peut établir le modèle mathématique de la réponse en position du moteur électrique qui est donné par la figure suivante (figure II.12):

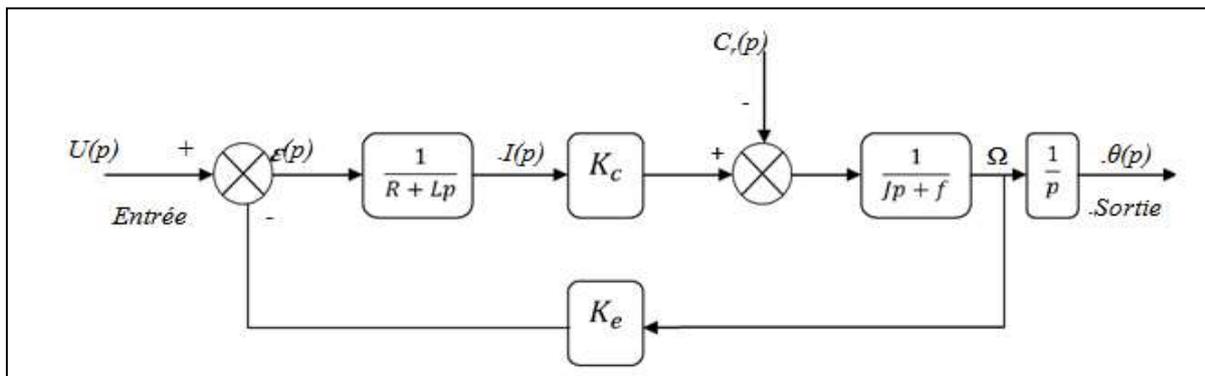


Figure II.12 Schéma bloc du Modèle de moteur électrique en position

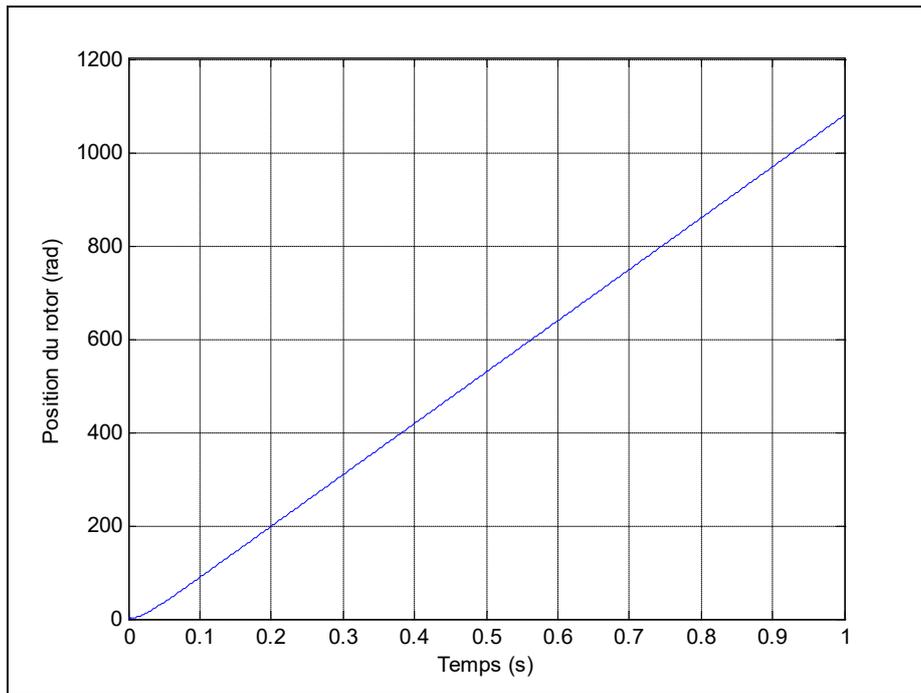


Figure II.13 Modélisation sous Matlab de la réponse de position

D'après la (figure II.13), on voit bien que la réponse en position est une rampe, d'où il nécessite une régulation pour stabiliser la réponse en position et d'améliorer les performances en vitesse.

Remarque: Pour un moteur électrique à courant continu, on a constaté que la réponse en position est une rampe, alors que la réponse en vitesse converge vers la stabilité, d'où, on voit bien la nécessité d'une commande pour stabiliser un système, et améliorer les performances.

II.4 L'application de la commande PID sur le model du moteur à courant continu

La commande PID est dite aussi (correcteur, régulateur, contrôleur), se compose de trois termes P, I et D, d'où le 'P' correspond au terme proportionnel, 'I' pour terme intégral et 'D' pour le terme dérivé de la commande. Les régulateurs PID sont probablement les plus largement utilisés dans le contrôle industriel. Même les plus complexes systèmes de contrôle industriel peuvent comporter un réseau de contrôle dont le principal élément est un module de contrôle PID.

Le régulateur PID est une simple implémentation de retour d'information (Feedback). Il a la capacité d'éliminer la compensation de l'état d'équilibre grâce à l'action intégrale, et il peut anticiper le futur grâce à une action dérivée, il existe plusieurs types de montage pour ce régulateur (série, parallèle, mixte...) [10].

Cette partie a pour but, d'implémenter la commande PID classique (parallèle) pour un moteur à courant continu, avec pour objectif d'annuler l'erreur statique, diminuer le dépassement, diminuer le temps de réponse et le temps de montée afin d'obtenir une réponse adéquate du procédé, de la régulation et d'avoir un système précis, rapide, stable et robuste. Le réglage des coefficients (paramètres) de la commande PID est basé sur la méthode de « Ziegler & Nichols ».

II.4.1 Asservissements et influence des coefficients

II.4.1.1 Asservissement P

L'asservissement de type P est le plus simple qui soit. Il s'agit d'appliquer une correction proportionnelle à l'erreur corrigeant de manière instantanée tout écart de la grandeur à régler :

$$U(t) = K_p \cdot \varepsilon(t) \quad \xrightarrow{t} \quad U(p) = K_p \cdot \varepsilon(p) \quad (\text{II.25})$$

Son rôle est d'amplifier virtuellement l'erreur pour que le système réagisse plus vivement, comme si l'erreur était plus grande qu'elle ne l'est en réalité.

Il permet de vaincre les grandes inerties du système et diminue le temps de montée (plus l'erreur est grande, plus on donne de puissance au moteur). Lorsque l'on augmente K_p , le système réagit plus vite et l'erreur statique s'en trouve améliorée, mais en contrepartie le système perd en stabilité. Le dépassement se fait de plus en plus grand, et le système peut même diverger dans le cas d'un K_p démesuré.[11]

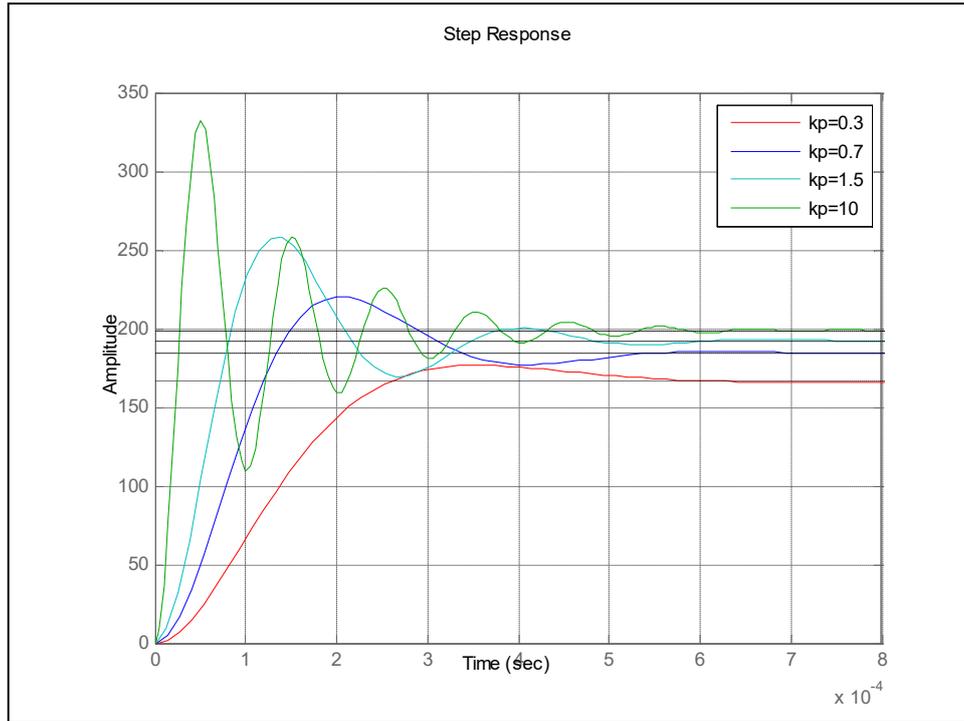


Figure II.14 correcteur proportionnel dans un asservissement en vitesse

Néanmoins, il faut toujours une certaine tension aux bornes du moteur pour que celui-ci puisse tourner. C’est pourquoi, lorsque l’on s’approche de la valeur demandée, l’erreur n’est plus assez grande pour faire avancer le moteur, ce qui fait que l’on n’atteint jamais vraiment la valeur demandée. Il subsiste alors une erreur statique, il est d’autant plus faible que K_p est grand.

II.4.1.2 Asservissement PI

L’asservissement de type PI est un asservissement de type P auquel on a ajouté un terme intégral :

$$U(t) = K_p \cdot \varepsilon(t) + K_i \cdot \int_0^t \varepsilon(\tau) d\tau \rightarrow U(p) = K_p \cdot \varepsilon(p) + K_i \cdot \frac{\varepsilon(p)}{p} = \varepsilon(p) \cdot \left[K_p + K_i \cdot \frac{1}{p} \right]$$

(II.26)

Le terme intégral complète l’action proportionnelle puisqu’il permet de compenser l’erreur statique et d’augmenter la précision en régime permanent. L’idée est d’intégrer l’erreur depuis le début et d’ajouter cette erreur à la consigne : lorsque l’on se rapproche de la valeur demandée, l’erreur devient de plus en plus faible. Le terme proportionnel n’agit plus mais le terme intégral subsiste et reste stable, ce qui maintient le moteur à la valeur demandée.

L'intégrale agissant comme un filtre sur le signal intégré, elle permet de diminuer l'impact des perturbations (bruit, parasites), et il en résulte alors un système plus stable.

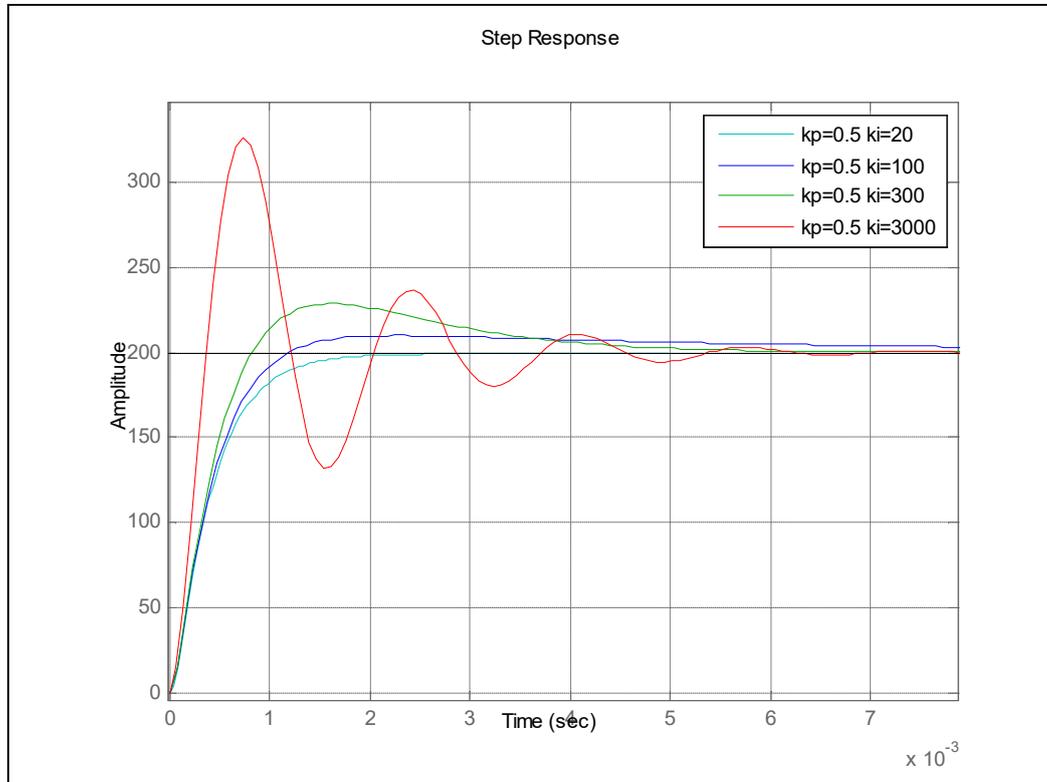


Figure II.15 correcteur PI dans asservissement en vitesse

Malheureusement, un terme intégral trop important peut lui aussi entraîner un dépassement de la consigne, une stabilisation plus lente, voire même des oscillations divergentes.[9],[10]

II.4.1.3 Asservissement PID

Les termes proportionnel et intégral peuvent amener un dépassement de la consigne et des oscillations dans des asservissements. Cela implique pour le moteur des inversions de polarité, ce qui est loin d'être idéal. Pour limiter ce phénomène indésirable, on introduit un troisième élément, le terme dérivé. Son action va dépendre du signe et de la vitesse de variation de l'erreur, et sera opposée à l'action proportionnelle. Elle devient prépondérante aux abords de la valeur demandée lorsque l'erreur devient faible, que l'action du terme proportionnel faiblit et que l'intégrale varie peu, elle freine alors le système, limitant le dépassement et diminuant le temps de stabilisation.[9],[10]

$$u(t) = K_p \cdot \varepsilon(t) + K_i \cdot \int_0^t \varepsilon(\tau) d\tau + K_d \cdot \frac{d}{dt} \varepsilon(t) \tag{II.27}$$

$$u(p) = K_p \cdot \varepsilon(p) + K_i \cdot \frac{\varepsilon(p)}{p} + K_d \cdot p \cdot \varepsilon(p) = \varepsilon(p) \cdot \left[K_p + K_i \cdot \frac{1}{p} + K_d p \right] \tag{II.28}$$

On obtient alors un asservissement composé d'un terme proportionnel, un terme intégral et un terme dérivé on parle d'asservissement PID.

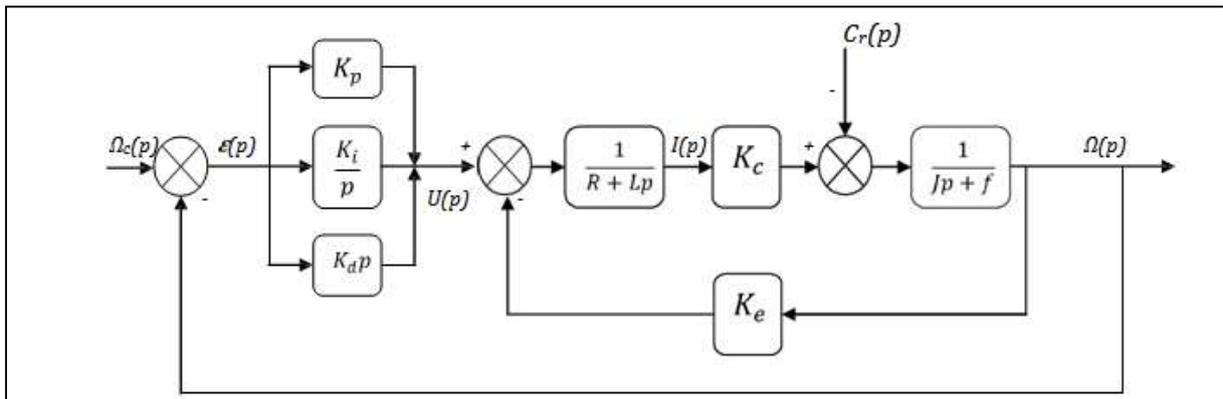


Figure II.16 Schéma bloc du correcteur PID en régulation de vitesse

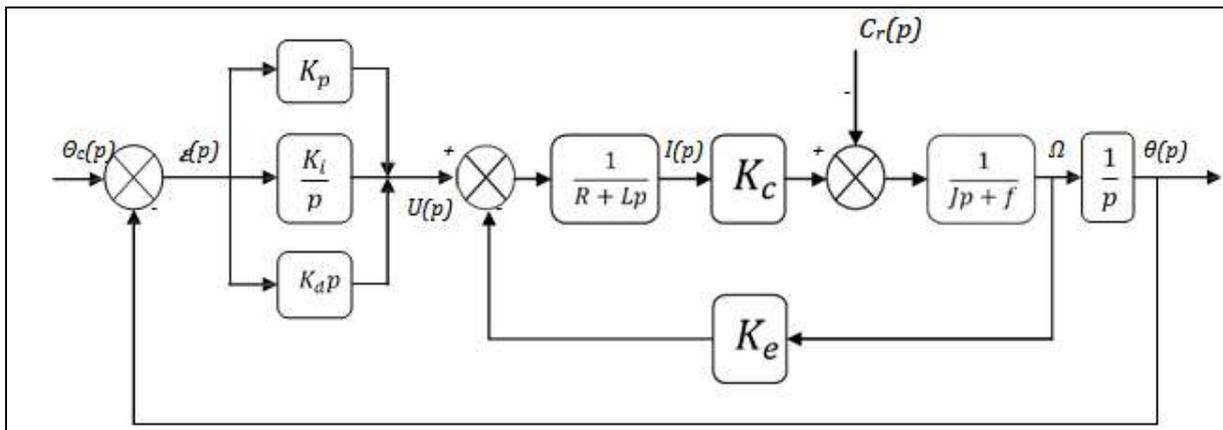


Figure II.17 Schéma bloc du correcteur PID en régulation de position

Remarque: L'action dérivée est surtout utilisée dans le cas de variables non bruitées, car la dérivation est très sensible au bruitage du signal : on diminuera donc son influence dans un

asservissement de vitesse, pour lequel la dérivée est l'accélération, variable soumise à de nombreuses perturbations.

Tableau II.2 tableau récapitulatif des paramètres PID

Coefficients	Temps de montée	Temps de stabilisation	Dépassement	Erreur Statique
K_p	Diminue	Augmente	Augmente	Diminue
K_i	Diminue	Augmente	Augmente	Elimine
K_d	Chang. faible	Diminue	Diminue	Chang. faible

II.5 Réglage des paramètres

Le réglage d'un PID consiste à trouver les meilleurs coefficients K_p , K_i et K_d dans le but d'obtenir une réponse adéquate du procédé et de la régulation. L'objectif est d'être, rapide et précis tout en limitant les dépassements [12].

II.5.1 Méthode de Ziegler & Nichols

En 1942, Ziegler et Nichols ont proposé deux approches heuristiques basées sur leur expérience et quelques simulations pour ajuster rapidement les paramètres des régulateurs P, PI et PID. La première méthode nécessite l'enregistrement de la réponse indicielle en boucle ouverte, alors que la deuxième demande d'amener le système bouclé à sa limite de stabilité.

II.5.1.1 Méthode de la courbe de réaction (Première méthode)

Pour obtenir les paramètres du régulateur PID, il suffit d'enregistrer la réponse indicielle du processus seul (c'est-à-dire sans le régulateur), puis de tracer la tangente au point d'inflexion de la courbe. On mesure ensuite sa pente p , le retard apparent L correspondant au point d'intersection de la tangente avec l'abscisse et le gain $K_0 = \frac{y(\infty)}{E}$. On peut alors calculer les coefficients du régulateur choisi à l'aide du tableau II.3.

Généralement, les gains K_p proposés par Ziegler-Nichols sont trop élevés et conduisent à un dépassement supérieur à 20%. Il s'agit ensuite de réduire K_p d'un facteur 2, par exemple pour obtenir une réponse satisfaisante.

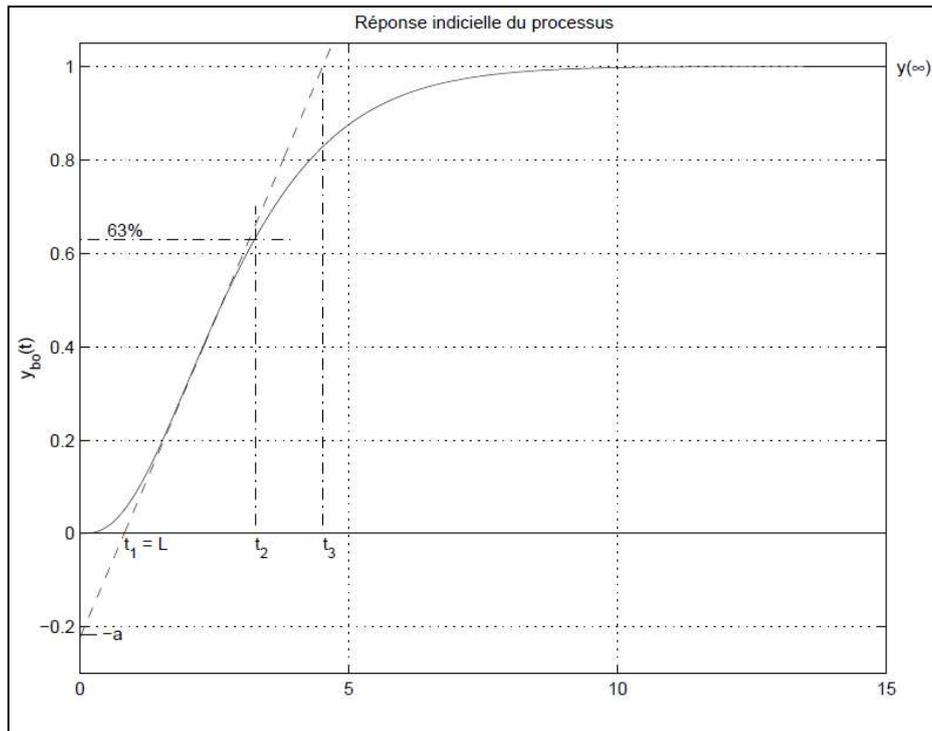


Figure II.18 réponse indicielle d'un processus du deuxième ordre

Tableau II.3 Paramètres PID obtenus à partir d'une réponse indicielle (ZN)

Type	K_p	$K_i=1/T_i$	$K_d=T_d$
P	$1/ (pLK0) = 1/ (aK0)$		
PI	$0.9/ (pLK0) = 0.9/ (aK0)$	3L	
PID	$1.2/ (pLK0) = 1.2/ (aK0)$	2L	0.5L

Avec :

Le retard apparent : $L = t_1$ (II.29)

La constante de temps apparente : $T = t_2 - t_1$ (II.30)

La pente de la tangente au point d'inflexion :

$$p = \frac{y(\infty)}{t_3 - t_1} = \frac{a}{L} \quad (II.31)$$

Le temps mort relatif :

$$\tau = \frac{L}{L+T} = \frac{t_1}{t_2} \quad (\text{II.32})$$

Le rapport entre la valeur asymptotique $y(\infty)$ et l'amplitude de l'échelon appliqué en entrée détermine le gain statique K_0 du processus :

$$K_0 = \frac{y(\infty)}{E} \quad (\text{II.33})$$

➤ Avantages

Cette méthode ne nécessite pas d'avoir un système déjà asservi, ni de deviner une valeur pour K_p .

➤ Inconvénients

- Elle nécessite plus de matériel : enregistrement d'une courbe de réponse. Elle oblige à mettre le système "offline" ;
- Si la réponse du système est trop différente de la réponse montrée ci-dessus, cette méthode donne des valeurs qui peuvent ne pas convenir du tout (certains systèmes par exemple ne sont pas du tout réglables par des régulateurs PID).[12]

II.5.1.2 Méthode d'oscillation (Seconde méthode)

Cette méthode est basée sur la connaissance du point critique du processus. Expérimentalement, on boucle le processus sur un simple régulateur proportionnel dont on augmente le gain jusqu'à amener le système à osciller de manière permanente ; on se trouve ainsi à la limite de stabilité. Ce point peut être caractérisé par deux paramètres de la fréquence ω_{180} et le gain à cette fréquence K_{180} . Pour des raisons historiques le point a été appelé le point ultime et caractérisé par les paramètres $K_u = 1/\omega_{180}$ que l'on appelle le gain ultime et $T_u = 2\pi/\omega_{180}$ la période ultime. On peut calculer les paramètres du régulateur choisi à l'aide du tableau II.4. Ici également, les valeurs proposées conduisent à un temps de montée relativement court malheureusement assorti d'un dépassement élevé. Cette situation n'étant pas toujours satisfaisante, on peut être amené à corriger les coefficients proposés et, en particulier, à diminuer le gain K_p .[12]

NB : avec la fonction de Matlab '**Margin**', on peut déterminer ω_{180} la fréquence du système.

Tableau II.4 Réglage du contrôleur par la méthode d'oscillation de Ziegler-Nichols

Régulateur	K_p	$K_i=1/T_i$	$K_d=T_d$
P	$K_u/2$	-	-
PI	$K_u/2.2$	$1.2/T_u$	-
PID	$K_u/1.7$	$2/T_u$	$T_u/8$

En pratique les paramètres peuvent être déterminés de la façon suivante : On connecte le régulateur au moteur électrique, fixe $K_i=0$ et $K_d=0$, on augmente doucement le gain du correcteur proportionnel pur K_p jusqu'à obtenir des oscillations persistantes, pour des moteurs, il s'agira plutôt d'une vibration autour de la position ou vitesse demandée. Le gain dans ce cas est K_u , et la période d'oscillations est T_u . [9].

➤ **Avantages**

La méthode est facile à mettre en œuvre physiquement et du point de vue calcul, elle peut être appliquée à un système déjà en production et permet une adaptation automatisée du régulateur pour s'adapter à l'évolution des paramètres intérieurs (usure) et extérieurs (environnement) au système.

➤ **Inconvénients**

Le système peut devenir instable ou passer dans des états dangereux (par exemple pour les systèmes chimiques). La méthode peut nécessiter beaucoup de temps si le système réagit très lentement (jours, semaine dans le cas de certaines réactions chimiques). Heureusement de nombreux systèmes ont des temps caractéristiques faibles (systèmes électroniques ou mécaniques). [12]

II.5.2 Application de la méthode de Ziegler-Nichols pour un asservissement en vitesse du moteur Maxon(110160)

II.5.2.1 Premier méthode

Pour plus de précision et pour éviter de faire des erreurs systématiques en utilisant la méthode graphique, nous avons préféré nous servir de l'outil *Control and Estimation Tool*

manager sur Simulink[14], on peut déterminer les coefficients des paramètres PID à l'aide de l'algorithme Ziegler-Nichols en boucle ouverte avec un dépassement de 20%.

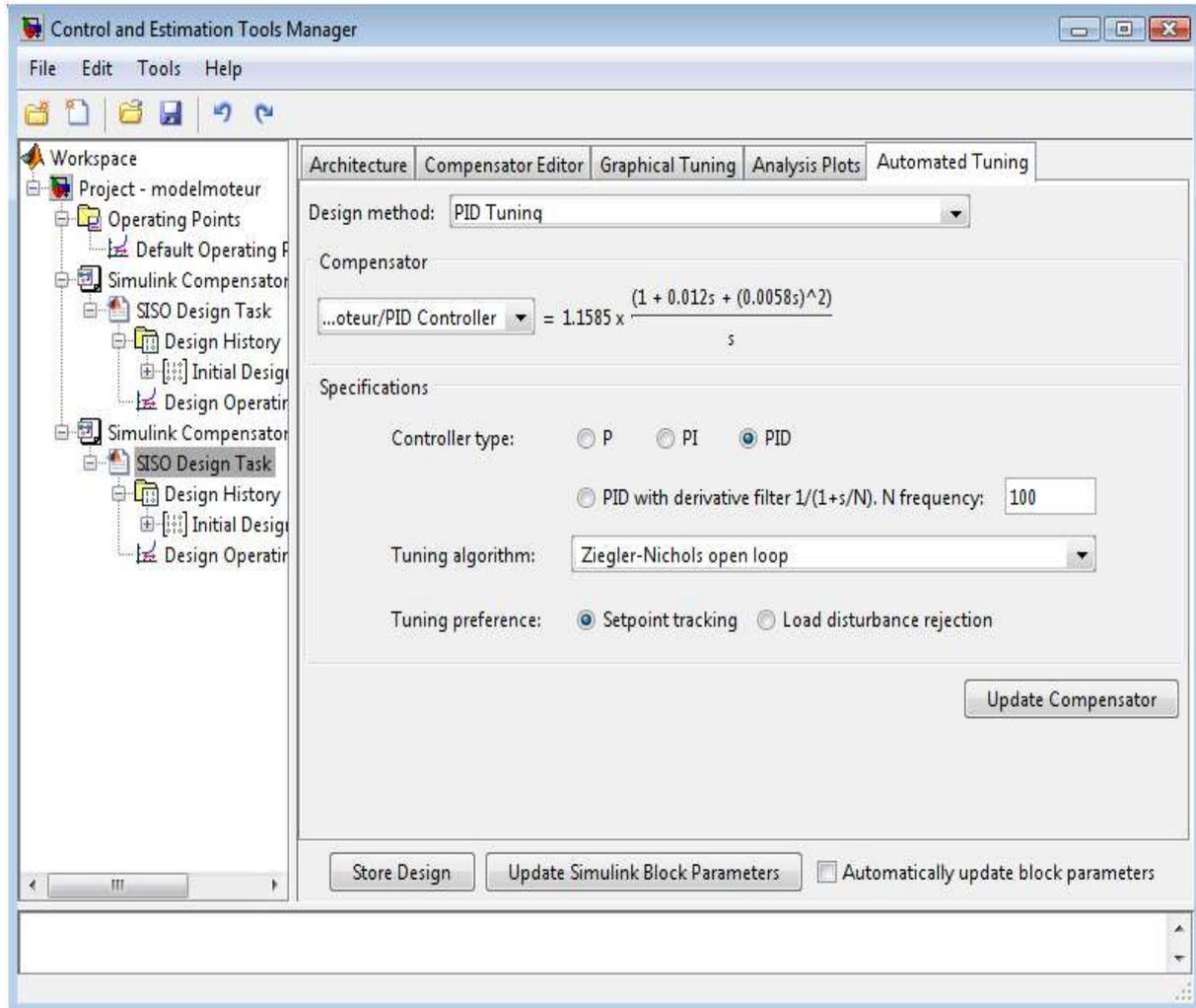


Figure II.19 Control and estimation tools manager

Le résultat donne les coefficients suivant :

Tableau II.5 les coefficients PID trouvés

Kp	0.13
Ki	0.9

Kd	0.00004
-----------	---------

En donnant une consigne de vitesse de 500 rad/s on a le résultat suivant :

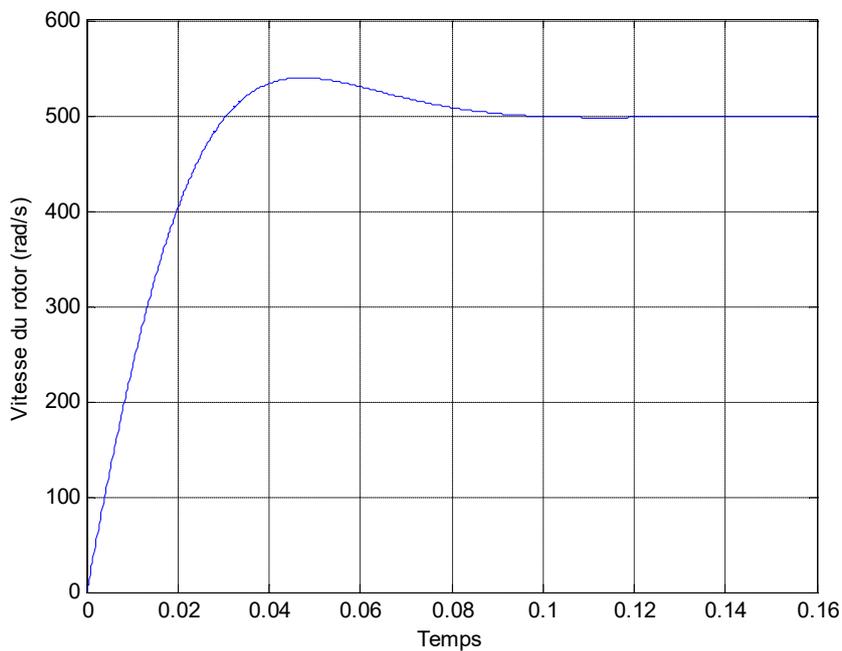


Figure II.20 Réponse du modèle moteur a une vitesse de 500 rad/s

II.5.2.2 Deuxième méthode

Nous avons instantanément remarqué que notre système est fortement stable. La méthode deuxième méthode de Ziegler-Nichols n'est alors pas nécessaire car le point ultime n'existe pas, en effet en calculant la marge de gain de notre système grâce à la fonction *Margin*, on obtient une marge de gain infinie.

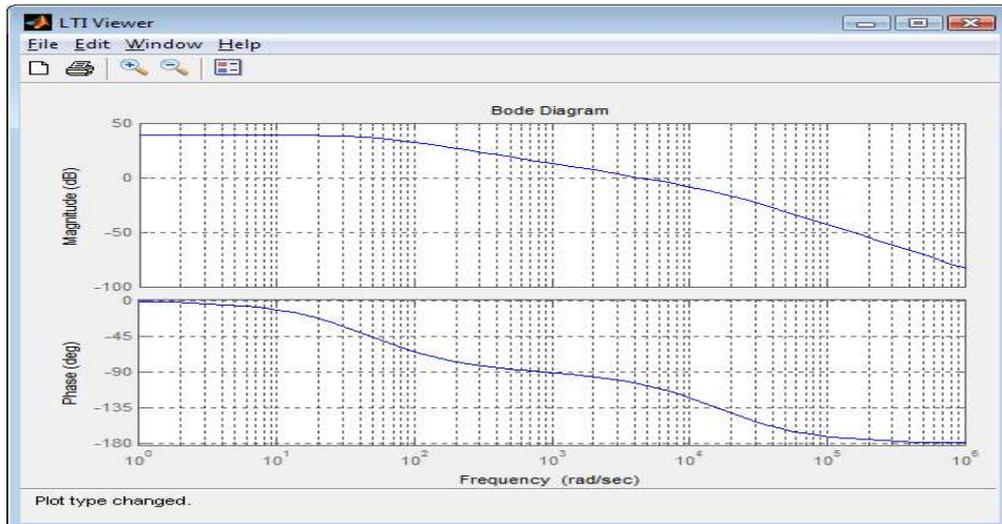


Figure II.21 Diagramme de Bode modèle du moteur

Puisque l'on peut se permettre d'augmenter K_p à volonté, on choisit une valeur relativement grande de K_p , puis on ajuste K_i et K_d pour éliminer l'erreur statique, limiter le dépassement et atteindre la consigne le plus vite possible.

Néanmoins, il faut aussi tenir compte du fait qu'un moteur n'accepte pas de trop fortes tensions : en pratique on ne peut augmenter les coefficients à volonté sans arriver à ce seuil de commande : le signal corrigé pourra éventuellement se retrouver tronqué, et le modèle considéré jusqu'ici ne sera alors plus valable.

II.5.3 Méthode avancée

Une méthode plus avancée consiste à modéliser parfaitement le système asservi, mais nécessite une connaissance parfaite de l'ensemble des paramètres permettant d'obtenir une modélisation précise. On s'applique alors à chercher à optimiser le temps de réponse et la stabilité par une étude mathématique.

L'avantage est que cette méthode permet en théorie d'obtenir les réglages les plus fins possibles. Mais l'inconvénient est qu'il est parfois difficile de modéliser parfaitement un système lorsque de nombreux phénomènes parasites (jeu mécanique dans les réducteurs, frottements et pertes omniprésents, usure, ...) ne sont pas négligeables. La précision des paramètres déterminés est alors à l'image de celle de la modélisation.

Remarque: On peut aussi tester directement la réponse du système à des signaux particuliers et en déduire une forme approchée de la fonction de transfert, puis se baser sur ce modèle approché pour mener à bien les calculs mathématiques.[9]

II.5.4 Limitations et approximations

II.5.4.1 Limitation de la tension de commande

Lors du réglage des coefficients du PID, il est primordial de se rappeler qu'un moteur n'est utilisable que dans une certaine plage de tensions (12V dans notre cas). Hors de cette zone, non seulement le comportement du moteur ne suivra plus obligatoirement le modèle établi, mais le moteur pourra éventuellement se détériorer irrémédiablement. Nous ne sommes donc pas totalement libres de choisir les coefficients K_p , K_i et K_d .

D'autre part, les frottements secs n'ont pas pu être pris en compte lors de cette modélisation

Il est probable que leur modélisation aurait entraîné deux principales conséquences qui n'ont pas pu être décelées :

- Un moteur n'est pas sensé démarrer pour des tensions trop faibles en raison de ce frottement sec ;
- L'erreur statique aurait probablement été plus grande.

Approximations

Le choix du modèle de moteur peut induire certaines approximations :

- Le jeu mécanique des pièces (réducteurs s'il y en a) n'est pas pris en compte ;
- L'usure des pièces n'est pas prise en compte ;
- Le modèle n'est valable que pour une plage de tension et de courant, au-delà de laquelle le comportement et la durée de vie du moteur sont incertains.

Le choix de la méthode d'implantation du PID peut elle aussi induire certaines approximations :

- Approximations des calculs du nombre dérivé et de l'intégrale ;
- Approximations en considérant le temps comme discret et non continu ;
- Approximations au niveau des capteurs (nombre de pas des encodeurs, mauvais comptage de pas, conversion analogique-numérique des tachogénérateurs, ...).

Finalement, l'erreur est faible, mais il est nécessaire d'en tenir compte. Pour minimiser son effet, diverses solutions sont envisageables :

- Diminuer dt pour se rapprocher d'un modèle (idéal) où le temps est continu ;
- Améliorer la précision des capteurs ;

- Recaler régulièrement le système (signal d'index sur les encodeurs, réaffectation des coordonnées en un point de coordonnées connues).[9]

II.6 Conclusion

Les trois actions du régulateur PID permettent de commander le moteur électrique à courant continu, tout en garantissant une annulation de l'erreur permanente de la sortie régulée, vis-à-vis d'échelons en entrée de consigne. Cette propriété de précision est due à la présence d'une action intégrale. L'ajout d'une action dérivée permet d'augmenter la stabilité du moteur électrique, et donc de diminuer le dépassement de la réponse indicielle. L'ajout de cette dérivée n'est pas systématique et dépend des propriétés du moteur électrique, du cahier de charges imposé (en terme de dépassement), dans certains cas il suffit d'utiliser un régulateur PI uniquement, comme l'asservissement en vitesse du moteur électrique.

Un des intérêts du régulateur PID, qui explique sa popularité dans le milieu industriel, est sans conteste la possibilité de le régler sans connaissance approfondie du système. En effet, on dispose de méthodes empiriques, fondées uniquement sur la réponse temporelle du système, selon une procédure expérimentale, comme la méthode d'oscillation de Ziegler-Nichols, permettant dans la majorité des cas d'aboutir à des performances acceptables.

Chapitre III

Mise en œuvre de la
commande PID d'un
moteur à courant
continu

III.1 Introduction

Dans le chapitre précédent nous avons vu que l'application de la théorie de la commande PID sur le modèle d'un moteur à courant continu en simulation sous Matlab pouvait offrir de bonnes performances, que ce soit au niveau de la précision, du temps de réaction, du dépassement, ou de la stabilité. Nous avons abordé les différentes méthodes de réglages des coefficients PID et leur efficacité.

Il s'agit maintenant de mettre en œuvre la commande PID en vitesse sur un moteur à courant continu. Les différentes parties de la mise en œuvre de la régulation seront d'abord exposées (système à commander, mesures..), ensuite l'algorithme sera implémenté sur le microcontrôleur. Les différentes variables seront récupérées et traitées sur un PC grâce à une communication série, finalement, les résultats seront discutés et les performances évaluées.

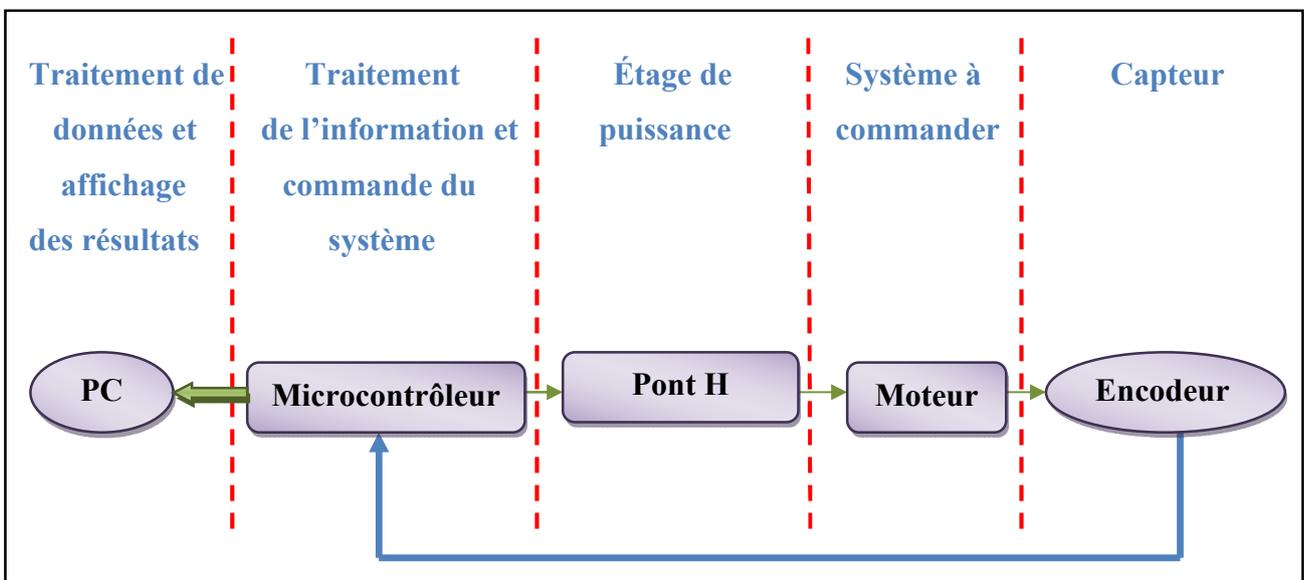


Figure III.1 Schéma de la mise en œuvre de l'asservissement

III.2 Système à commander

Pour notre application, nous disposons d'un moteur à courant avec balais. Le moteur est équipé d'un réducteur mécanique pour réduire la vitesse et augmenter le couple. Un encodeur optique incrémental à quadrature de phase est relié au rotor du moteur, ce qui nous permet de récupérer une information sur l'incrémentangulaire du rotor, à partir de laquelle on peut calculer la vitesse de rotation du rotor et sa position angulaire par rapport à sa position initiale.

Il s'agit d'un disque en rotation doté de deux rangées de fentes décalées. Un système optique détecte ces fentes et émet 512 impulsions par tour de rotor, sur deux signaux en quadrature de phase la quadrature servant ici à déterminer le sens de rotation du rotor les caractéristiques techniques sont donnée en l'annexe2.

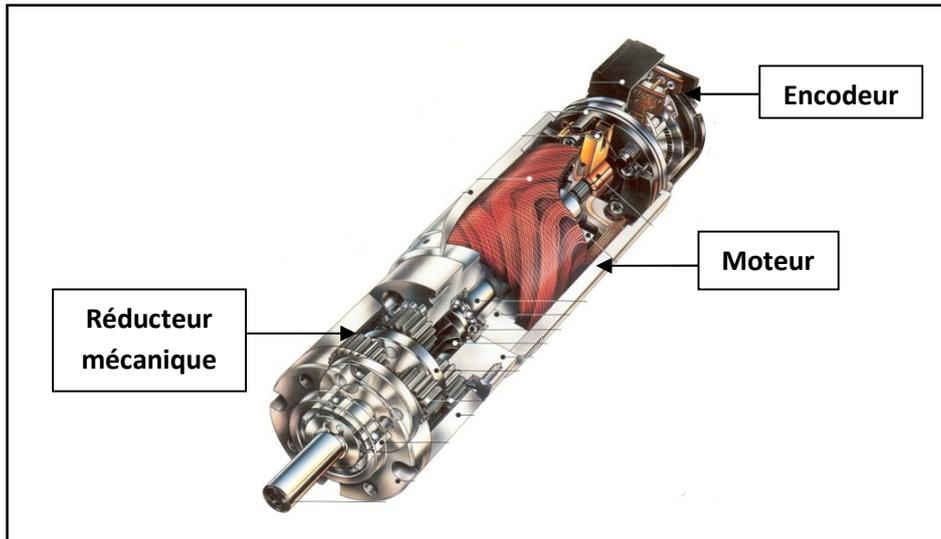


Figure III.2 Moteur à courant continu équipé d'un encodeur et d'un réducteur mécanique

III.3 Calcul de la vitesse et de la position

III.3.1 Lecture de l'encodeur

Dans notre application, la carte électronique s'occupe de l'acquisition et du traitement des signaux, elle s'articule autour d'un microcontrôleur (AtMega2560 à 16MHz). Ce microcontrôleur acquiert les signaux venant de l'encodeur optique par l'intermédiaire d'une bascule D.

Un encodeur incrémental possède généralement plusieurs voies:

- Voie Z donnant une impulsion par tour,
- Voie A donnant n impulsions par tour,
- Voie B identique à voie A, mais dont les signaux sont déphasés de + ou - 90°, suivant le sens de rotation.

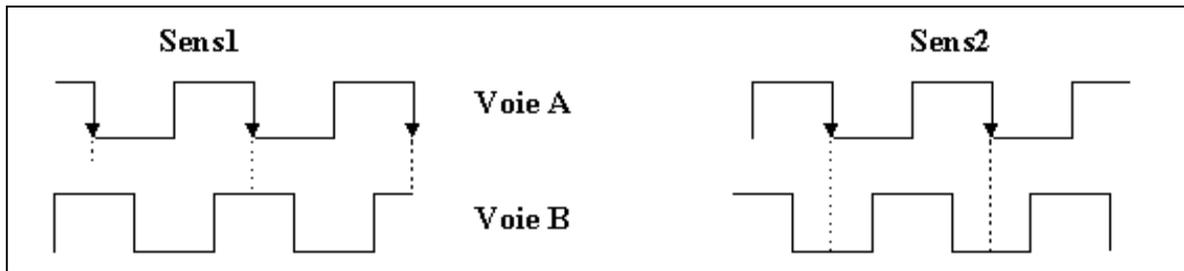


Figure III.3 Signaux en quadrature

La simple utilisation d'une bascule D permet d'obtenir une information logique concernant le sens de rotation:

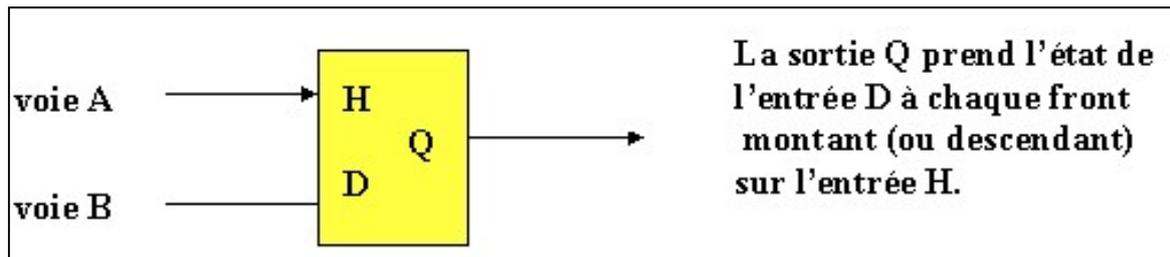


Figure III.4 Détection du sens de rotation à l'aide d'une bascule D

Les caractéristiques principales d'un encodeur incrémental sont :

- nombre de points par tour ou nombre d'impulsions par tour, dans notre cas 512points/tr ;
- nombre de voies ou nombre de pistes (exemple: 6 pistes A, B, Z, A*, B*, Z*) ;
- vitesse maximale de rotation.

Après avoir récupéré les signaux provenant des encodeurs, nous pouvons avoir deux approches pour calculer la vitesse et la position du rotor. La première consiste à prendre la mesure sur la vitesse et en déduire la position, et la deuxième à prendre la mesure sur la position pour en déduire la vitesse.

III.3.2 mesure de vitesse et calcul de position

Dans cette approche, nous exploitons l'unité de capture du microcontrôleur. C'est un périphérique qui permet aux événements d'être datés dès qu'ils surviennent. Ce calendrier peut souvent être précis à un certain nombre de nanosecondes. L'unité est une partie intégrante d'un ou plusieurs des timers du microcontrôleur.

Une des lignes d'E / S du processeur est l'élément déclencheur de l'appareil. Une fonction Register spéciale est utilisée pour configurer le comportement de l'entrée de capture. Souvent ce registre contient un bit ou deux qui décrivent les types d'événements déclenchant une capture d'entrée. Par exemple, la capture d'entrée pourrait se produire sur un front montant ou descendant.[15]

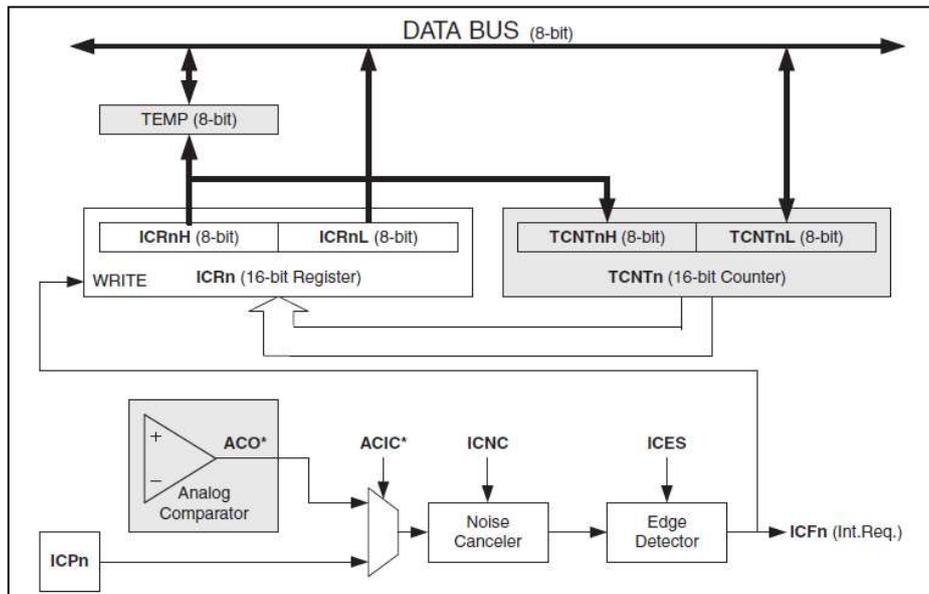


Figure III.5 Schéma de l'unité de capture de l'atmega2560

Le timer du microcontrôleur est basé sur un compteur à 16 bits qui peut être constamment incrémenté (ou décrémenté) automatiquement+. La vitesse à laquelle le timer est incrémenté est fixée par un ou plusieurs registres à fonction spéciale associé à cette temporisation. Quand un événement se produit à la capture d'entrée, la valeur du compteur du timer à ce moment est copiée sur l'unité de prise d'entrée.

Pour mesurer la vitesse de rotation du rotor, on configure le déclenchement d'une capture sur un front montant du signal provenant du canal A de l'encodeur. Dès qu'une capture se déclenche, cet instant est enregistré comme étant T_i . Au prochain front montant, cet instant est enregistré comme étant T_f . On aura alors la période T du signale qui est donnée par $T = T_f - T_i$.

Connaissant la période T du signal et la distance entre deux fronts montants qui est donnée par la résolution du codeur :

$$\mathit{résolution} = \frac{2\pi}{\mathit{nombre\ de\ point\ par\ tour}} \quad (\text{III.1})$$

Il est facile de calculer la vitesse ω en radian/s du rotor :

$$\omega = \mathit{résolution} / T \quad (\text{III.2})$$

A chaque intervalle de temps fixe, une nouvelle estimation de la position sera effectuée en comparant la valeur de la vitesse avec sa valeur au cycle précédent, la somme de ces valeurs multipliée par le temps de cycle donne la position sur le cycle précédent. Si le temps de cycle est suffisamment court, on peut considérer le résultat comme la position instantanée.

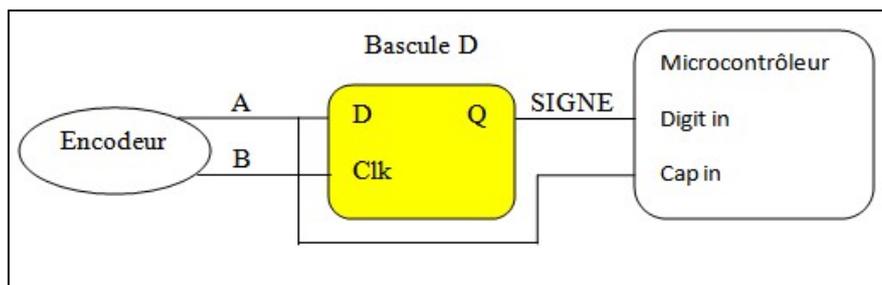


Figure III.6 Schéma de mise en œuvre de la mesure de vitesse

Cette approche nous donne une mesure de la vitesse plus précise que celle sur la position. C'est pour cette raison que nous l'utilisons pour l'asservissement en vitesse.

III.3.3 Mesure de la position et calcul de vitesse

Dans cette approche on utilise les interruptions. On connecte le canal A de l'encodeur sur une entrée interruption du microcontrôleur et le bit de signe sur une entrée numérique, une variable est mise à jour à chaque front montant du signal canal A et en comparant avec le bit de signe la variable est soit incrémentée soit décrétementée, une simple conversion nous permet d'obtenir l'angle β en radian effectué par le rotor du moteur

$$\beta = \mathit{pascodeur} * \mathit{resolution} \quad (\text{III.3})$$

On peut augmenter la résolution déclenchant une interruption sur front montant et descendant et sur chacun des deux signaux en quadrature. Ainsi on pourra avoir jusqu'à 4 fois la résolution initiale de l'encodeur ; malheureusement, cela peut provoquer un débordement plus rapide de la variable, une gestion des débordements devra être mise en place. À chaque cycle, une nouvelle estimation de la vitesse de chaque roue sera effectuée en comparant la valeur de la variable avec sa valeur au cycle précédent. La différence entre ces valeurs divisée par le temps de cycle donne la vitesse moyenne sur le cycle précédent. Si le temps de cycle est suffisamment court, on peut considérer le résultat comme la vitesse instantanée.

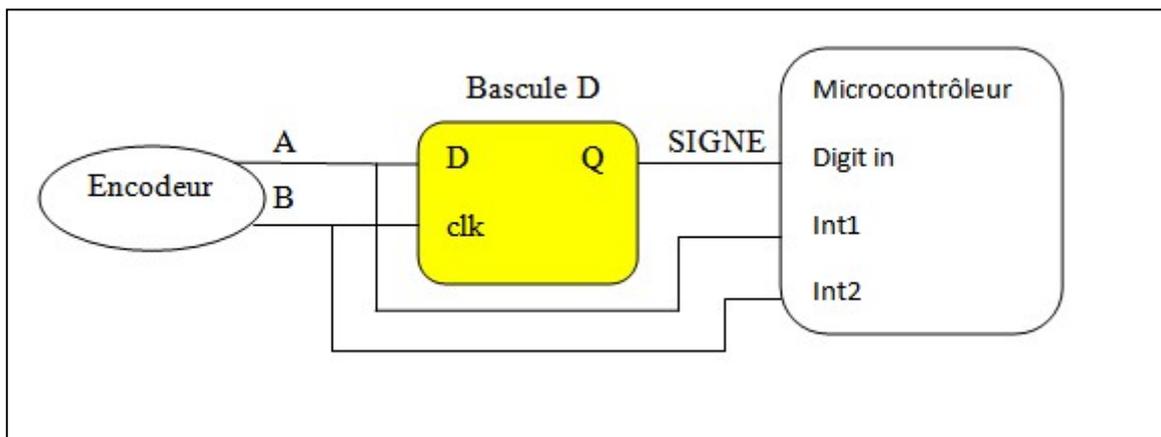


Figure III.7 Schéma de mise en œuvre de la mesure de position

Cette approche nous donne une mesure en position plus précise que celle sur la vitesse. C'est pour cette raison que nous l'utilisons pour l'asservissement en position.

Remarque: pour appliquer cette méthode, nous avons dû utiliser deux microcontrôleurs: le premier étant chargé de mesurer le nombre de pas effectués par l'encodeur, de l'envoyer via le bus I2c au deuxième qui va calculer la vitesse et réaliser l'asservissement. En effet, si cette approche utilise le principe de l'interruption qui va arrêter l'exécution du programme principal pour effectuer une routine (routine d'interruption), la boucle de régulation sera rompue. Cela étant plus contraignant si la vitesse du moteur est élevée, ce qui est le cas dans notre application, sachant que la fréquence du signal provenant d'un seul canal et déclenchant l'interruption sur un front montant peut atteindre jusqu'à 8kHz. Le microcontrôleur ne pourra exécuter aucune autre instruction que celle de la routine d'interruption. L'asservissement devient alors impossible.

On peut dire que l'utilisation d'un composant spécialisé ou d'un microcontrôleur disposant de périphérique spécialisé dans la gestion des signaux en quadrature serait plus appropriée pour ce type d'application.

III.4 la carte de Puissance

III.4.1 Le pont H

Les moteurs électriques ont besoin de courants importants par rapport à la sortie maximale d'un microcontrôleur. Un moteur ne marcherait pas si on le branchait directement à la sortie d'un microcontrôleur. Un des montages les plus utilisés pour la commande d'un moteur à courant continu est le variateur de courant continu bidirectionnel ou pont en H [16], [17]. Ce montage est représenté à la figure III.8.

La première fonction du pont en H est de fournir de forts courants aux moteurs en permettant leur contrôle par le microcontrôleur. Une autre fonction est de pouvoir inverser Le courant traversant le moteur pour inverser le sens de rotation ce qui est souvent utile en robotique. Inverser le sens d'un courant est une tâche simple dans les circuits électriques, il suffit de bien placer quelques interrupteurs. On utilise des transistors à la place d'interrupteurs parce qu'un transistor peut aussi amplifier un courant et est plus facile à contrôler par des signaux numériques ou analogiques.

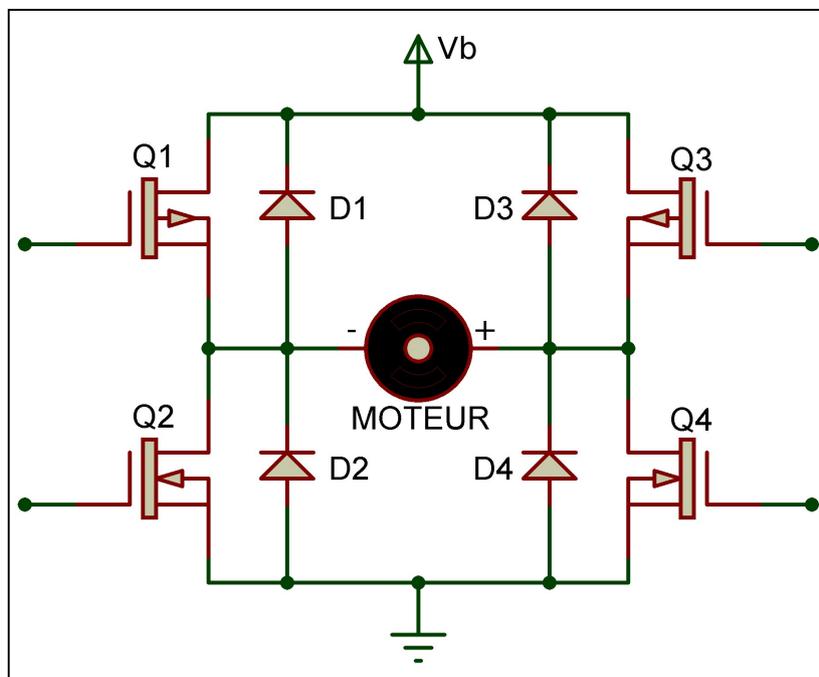


Figure III.8 Schéma du montage du pont H

III.4.2 Principe de fonctionnement du pont H

Le fonctionnement de base du pont en H est relativement simple, il permet d'alimenter le moteur dans un sens ou dans l'autre, rendant ainsi possible son changement de sens de rotation. En effet, si Q2 et Q3 conduisent, la borne moins du moteur sera connectée à la masse et la borne plus à la tension d'alimentation faisant ainsi tourner le moteur dans son sens de rotation normal. Si maintenant ce sont Q1 et Q4 qui conduisent, la borne moins du moteur sera connectée à la tension d'alimentation et la borne plus à la masse faisant alors tourner le moteur en sens inverse.

Pour faire varier la vitesse du moteur, il faut appliquer une tension variable aux bornes de celui-ci. Pour ce faire, la commutation des transistors est commandée par un signal PWM (Pulse Width Modulation) de fréquence relativement élevée. Le moteur passe ainsi de manière très rapide d'état alimenté à non alimenté et vice-versa. La tension moyenne vue par le moteur (charge inductive) sera alors déterminée par le rapport cyclique (le rapport entre le temps ON et le temps OFF) du signal PWM utilisé. Pour calculer la valeur de la tension aux bornes du moteur on se réfère à l'équation suivante :

$$U = E(2\alpha - 1) \quad \text{(III.4)}$$

Les diodes D1 à D4 sont appelées diodes de roue libre et sont nécessaires au bon fonctionnement du pont en H. Lorsque le moteur est alimenté (Q2 et Q3 conduisent ou Q1 et Q4 conduisent), ces diodes n'ont aucun effet. Elles servent uniquement lorsque le moteur passe d'un état alimenté à un état non alimenté. En effet, dans ce cas, les transistors vont arrêter brusquement de conduire le courant et celui-ci pourra alors passer par les diodes de roue libre. Cela permet de limiter les pics de tension inverse aux bornes du moteur et ainsi d'éviter la destruction des composants électroniques. Remarquons que les transistors Q1 et Q2 ou Q3 et Q4 ne doivent jamais conduire simultanément sous peine de court-circuit franc et de destruction des transistors.

Le pont en H peut être commandé de plusieurs façons différentes suivant la manière dont les commutations des transistors sont enchainées. Nous allons étudier ci-après les deux modes de commande les plus connus. Les informations concernant les modes de contrôle particuliers des ponts en H sont principalement tirées de notes d'applications de pont H intégrées, en particulier, nous citerons celle du L298 [18].

III.4.3 Les deux principaux modes de commande du pont H

III.4.3.1 Le mode Sign – Magnitude

Dans le mode *Sign - Magnitude*, pour une direction donnée, le signal PWM fait varier les transistors entre deux phases. Dans la première phase, un des transistors de la partie haute du pont et le transistor opposé de la partie basse conduisent et alimentent le moteur.

Dans la deuxième phase, le transistor de la partie basse est coupé mettant alors le moteur en roue libre. Lors de la transition de la première phase à la deuxième phase, le courant ne pouvant plus passer par le transistor bas, passe par la diode haute correspondante.

La (figure III.9) représente le diagramme temporel du mode *Sign - Magnitude*. Les signaux de commutation des transistors, les tensions aux bornes du moteur et le courant passant dans le moteur y sont représentées.

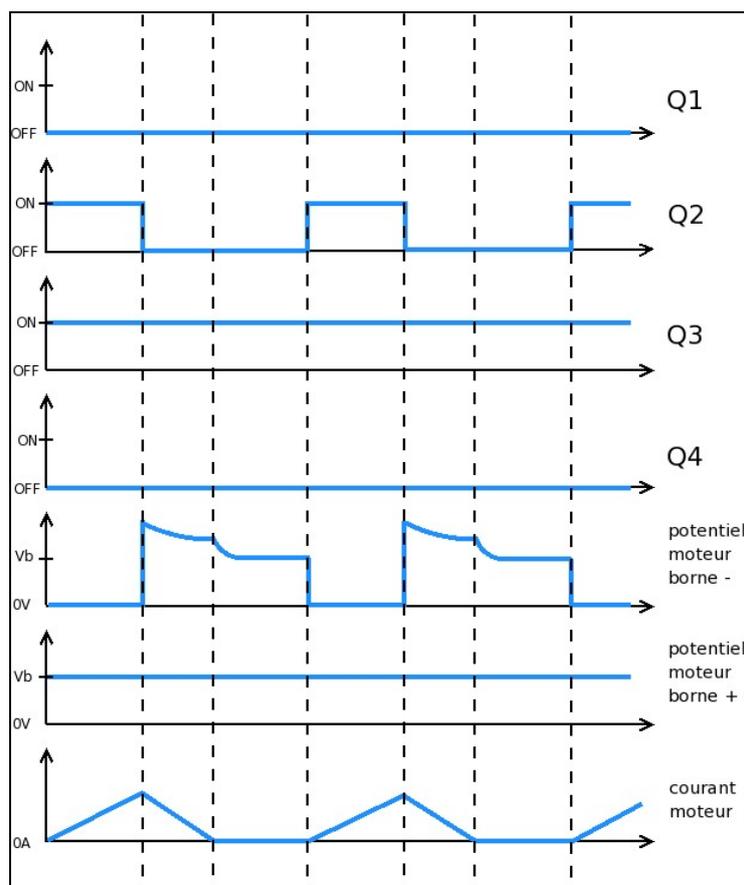


Figure III.9 Diagramme temporel Sign – Magnitude

III.4.3.2 Le mode Locked Anti-phase

Dans le mode *Locked Anti-phase*, aucune distinction n'est faite quant à la direction de rotation du moteur, les deux directions étant contrôlées directement par le signal PWM. Ce mode est aussi divisé en deux phases.

Dans la première phase, un des transistors de la partie haute du pont et le transistor opposé de la partie basse conduisent et alimentent le moteur dans un certain sens.

Dans la deuxième phase, ce sont les deux autres transistors qui conduisent et qui alimentent alors le moteur dans le sens inverse. Le moteur sera donc connecté en permanence à une source de faible impédance.

Il est à remarquer que pour mettre le moteur à l'arrêt, un rapport cyclique de 50% doit être utilisé.

La (figure III.10) représente le diagramme temporel du mode *Locked Anti-phase*. Les signaux de commutation des transistors, les tensions aux bornes du moteur et le courant passant dans le moteur y sont représentées.

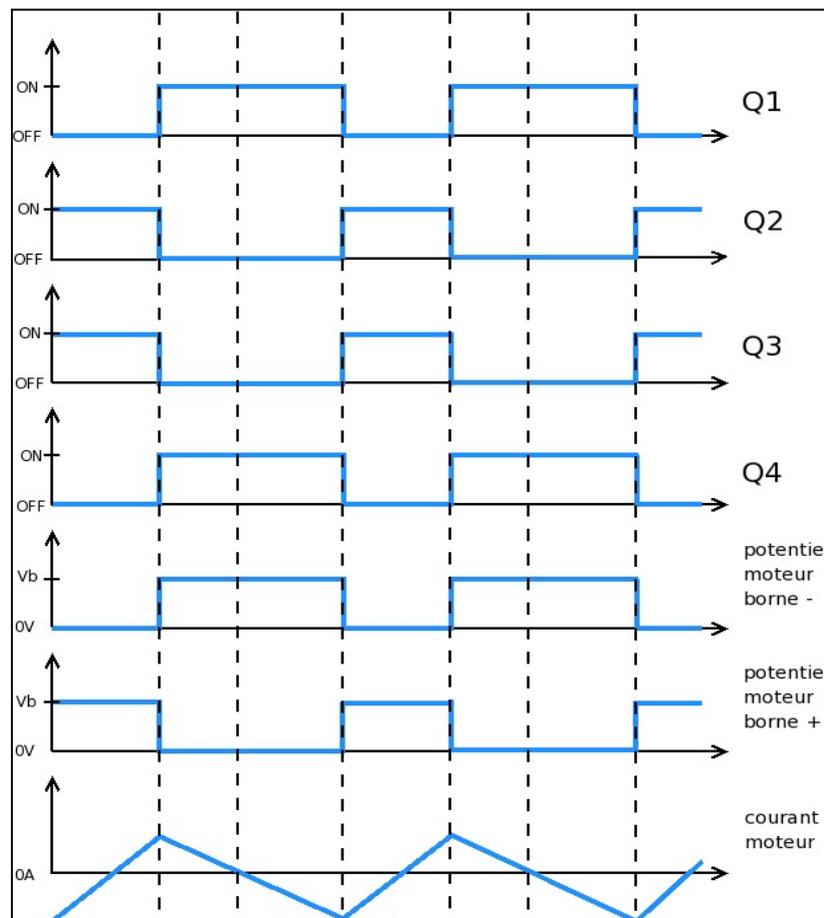


Figure III.10 Diagramme temporel Locked Anti-phase

III.4.3.3 Comparaison entre les deux modes

Bien que le mode *Sign - Magnitude* soit très souvent utilisé, c'est loin d'être la solution idéale pour notre utilisation. En effet, il est particulièrement mal adapté au contrôle en boucle fermée car il ne permet pas d'avoir un contrôle total du moteur. Le moteur étant en roue libre pendant une partie du temps, il sera en fait impossible de le freiner activement. La seule solution pour réaliser un freinage dans ce mode est de court-circuiter les bornes du moteur.

Le mode *Locked Anti-phase* permet, lui, de contrôler parfaitement l'accélération et la décélération du moteur. En effet, le moteur est toujours connecté à une source de faible impédance et donc la tension aux bornes du moteur est toujours imposée. La tension moyenne vue par le moteur sera, des lors, directement proportionnelle au rapport cyclique de la PWM. Ce qui est capital pour notre contrôle en boucle fermée.

Le principal problème du *Locked Anti-phase* est qu'il nécessite de travailler avec des fréquences de PWM bien plus élevée que pour le *Sign -Magnitude*, ceci afin de lisser, correctement le courant passant dans le moteur.

Les temps de commutation des transistors devront donc être très courts afin de limiter les pertes.

Le contrôle en boucle fermée étant un des points clés de notre application, c'est le mode *Locked Anti-phase* qui a été retenu. Un effort tout particulier sera donc fait dans la suite de notre conception afin de tenir compte des fréquences de commutation relativement élevées qui seront utilisées[19].

III.4.4 Pilotage d'un moteur par PWM

III.4.4.1 Principe de la PWM

La PWM (Pulse Width Modulation) se traduit par Modulation de largeurs d'imputions (MLI). elle consiste à alterner rapidement entre deux états distincts du système afin d'obtenir en moyenne un signal analogique PWM[20].

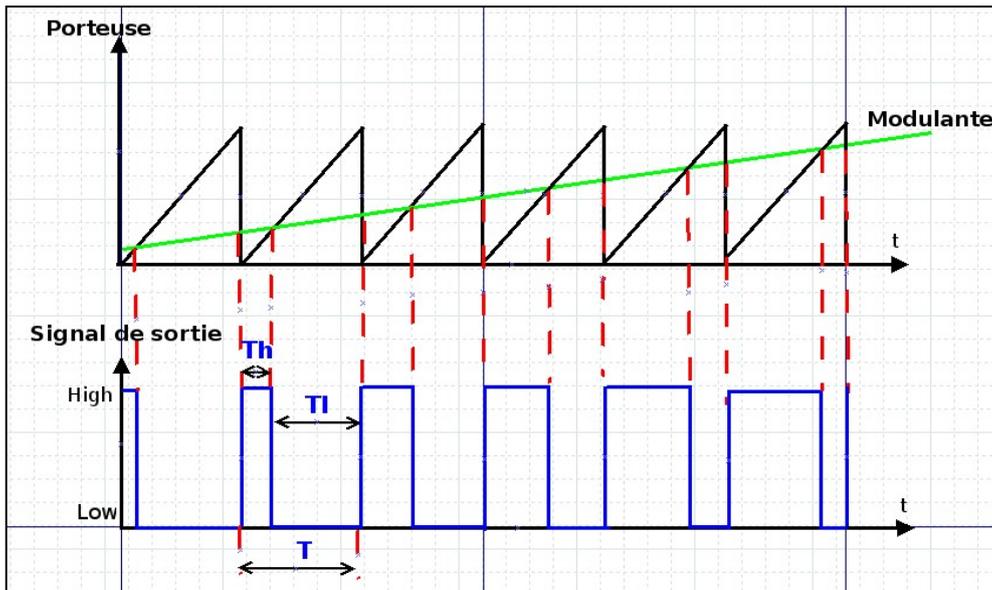


Figure III.11 Principe de la PWM

Comme représenté, le signal PWM peut être obtenu en comparant un signal avec un autre signal triangulaire. En effet, si le signal comparé est au dessus du signal triangulaire, la sortie est à l'état bas. Dans le cas contraire, la sortie est mise à l'état haut. Le signal continu généré à partir des deux valeurs distinctes prend la valeur moyenne sur les intervalles de temps.

On précise que dans la figure III.11 :

- Th représente la période de temps pendant laquelle le système est à l'état haut ;
- Tl représente la période de temps pendant laquelle le système est à l'état bas.

La PWM se caractérise par :

- La période PWM : $T_{pwm} = T_h + T_l$;
- Le rapport cyclique d'impulsion (Duty cycle) $\alpha = \frac{T_h}{T_{PWM}}$.

La fréquence PWM est toujours constante. Par contre. Le rapport cyclique peut varier. De plus, le signal PWM est un signal rectangulaire qui varie entre deux valeurs extrêmes, mais les systèmes sur lesquels on applique la PWM sont en général des filtres passe bas (moteurs, œil humain, etc.), ils ne ressentent que la valeur moyenne du PWM.

III.4.4.2 Modes de la PWM

Il existe deux modes sur la PWM :

➤ *edge-aligned*

Le signal est asymétrique par rapport au début de la période. Ce signal s'obtient en comparant un signal constant avec un signal triangulaire possédant une seule pente. Nous trouvons le mode *left edge-aligned* (à chaque début de période, on est à l'état haut) et le mode *right edge-aligned* (à chaque fin de période, on est à l'état haut).

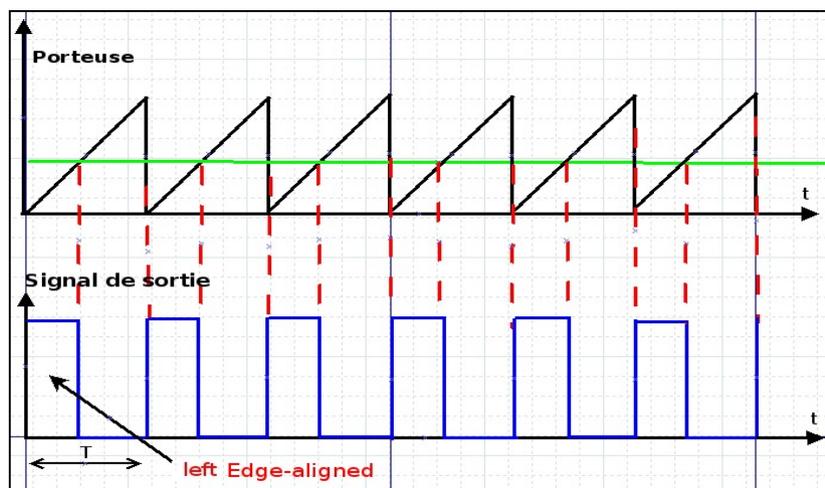


Figure III.12 edge-aligned PWM

➤ *center-aligned*

Le signal est une fois à l'état haut, et à la période suivante à l'état bas. Ce signal est donc symétrique par rapport au début d'une période. Ce signal s'obtient en comparant un signal constant avec un signal triangulaire à deux pentes.

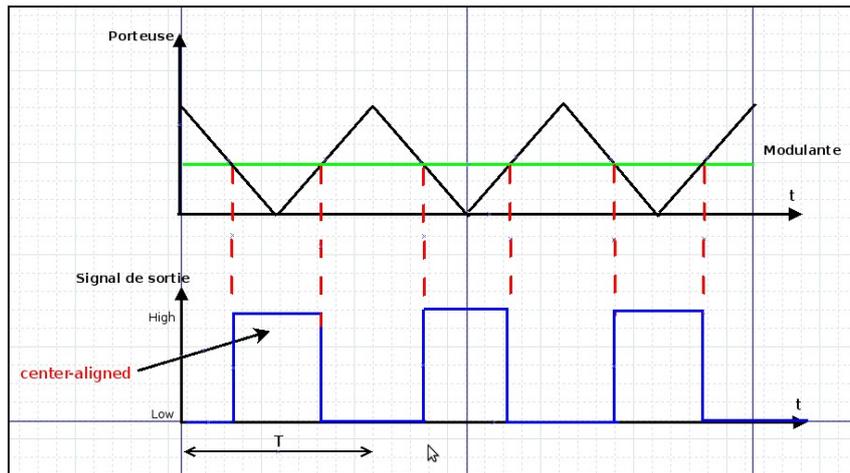


Figure III.13 center-aligned PWM

III.4.4.3 Choix de la fréquence de la PWM

Pour commander un moteur en continu, il y a un certain intervalle de fréquences PWM permise. D'une part, une fréquence minimale f engendre la sensibilité du moteur à la variation de tension. Pour se mettre au dessus de cette fréquence minimale, il faut suivre les étapes suivantes :

- Tout d'abord, on fixe le pourcentage de stabilité P (donne le degré de variation du courant) qu'on souhaite avoir ;
- Ensuite, on se réfère au datasheet du moteur pour obtenir la résistance R et l'inductance L de son schéma équivalent.

On additionne les résistances internes du pont H (MOSFET) et on applique la formule suivante [20]:

$$f = \frac{R}{-2L \cdot \ln\left(1 - \frac{P}{100}\right)} \quad (\text{III.5})$$

D'autre part, quand on utilise un pont en H, il ne faut pas dépasser une certaine fréquence pour limiter les pertes lors de la commutation des transistors et aussi pour respecter les limites d'utilisation spécifiées par le constructeur.

Mais si on choisit une fréquence de PWM qui appartient au spectre audible, nous allons entendre des sifflements au niveau du moteur et ce dernier pourrait éventuellement chauffer. Il faut alors trouver un compromis entre ces paramètres.

Pour notre application nous avons jugé empiriquement qu'une fréquence de 20kHz été satisfaisante.

III.4.5 Génération de la PWM

Le microcontrôleur contient un module destiné à générer des PWM basé sur un de ses timers 16bit. Le cas échéant, le timer3. Pour que le moteur filtre correctement un signal PWM, ce dernier doit avoir une fréquence de 20 kHz.

La fréquence du PWM se règle au moyen du registre OCR1A et de la valeur du prescaler. L'horloge du microcontrôleur est donnée par l'oscillateur interne qui a une fréquence de 16Mhz, la résolution de la PWM peut varier de 2 jusqu'à 10bit. Il nous a paru judicieux de choisir la résolution la plus élevée qui donne une valeur maximale du registre OCR1A de 1023.

On obtient la fréquence en utilisant l'équation suivante [15]:

$$f_{ocncPWM} = \frac{f_{clk_I/O}}{N \cdot (1 + TOP)} \quad (III.6)$$

Il s'agit de calculer à partir la valeur TOP du registre OCR1A. Pour régler le rapport cyclique on entre dans le registre OC1x une valeur de 0 à 1023 (100 %).

Le principe est illustré sur (la figure III.14) issue du manuel de référence de la famille de ce microcontrôleur [15].

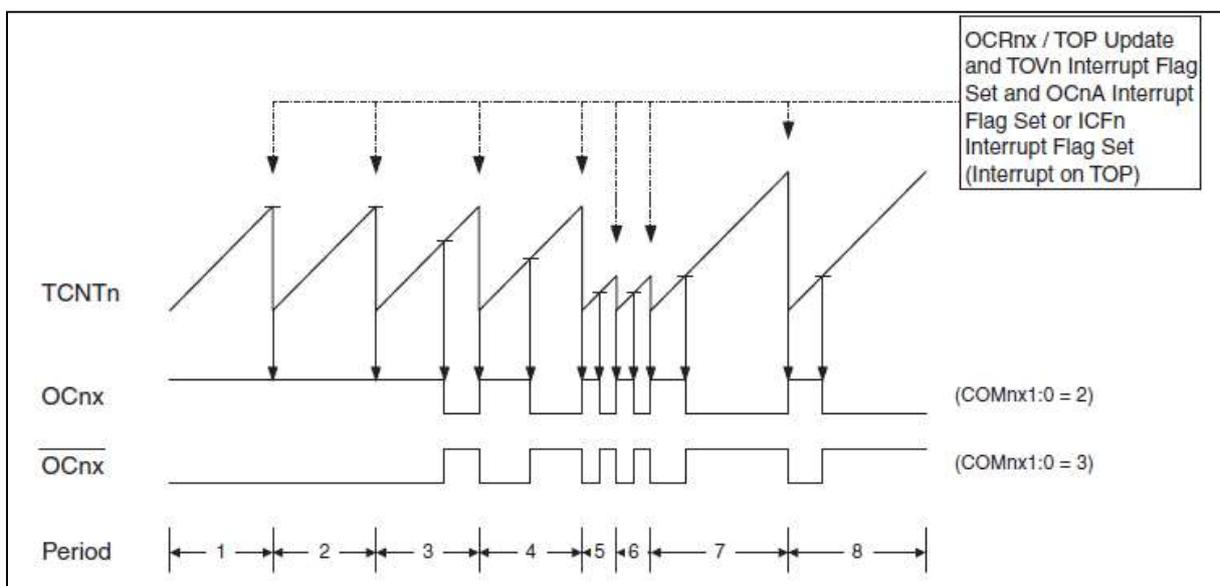


Figure III.14 Edge-aligned PWM Mode, Timing Diagram

III.5 Communication

Pour pouvoir régler l'asservissement, il a été nécessaire de récupérer l'historique d'évolution des diverses variables et d'en tracer la représentation graphique après adaptation des données brutes en un format plus exploitable :

1. Au début de chaque session de mesures, on télécharge sur le microcontrôleur les paramètres à utiliser : coefficients du/des PID, consigne à atteindre, vitesse maximale, accélération maximale et durée de session. Une fois ces paramètres renseignés, un message apparaissant sur le moniteur demandant à l'utilisateur d'appuyer sur une touche, et la session de mesures peut commencer ;
2. Dès lors, en plus de gérer l'asservissement du moteur, le microcontrôleur envoie à l'ordinateur (via une connexion série) les variables concernées, et ce à chaque boucle de correction : position, vitesse, erreur, correction demandée et tension appliquée ;
3. Une fois ces données enregistrées, elles sont confiées à un logiciel spécialisé dans l'exploitation de données et le tracé de courbes (KST2) afin d'en tracer la représentation graphique de manière facilitée ;
4. Une fois la session terminée, les variables internes (état de l'intégrale, position angulaire, ..) sont réinitialisées et une nouvelle session peut commencer.

III.6 Implémentation de l'asservissement

L'implémentation du régulateur a simplement consisté à écrire en C le code correspondant au régulateur choisi. Il a cependant fallu rester attentif aux temps d'exécution des différentes opérations, afin que l'on ne dépasse pas la période d'échantillonnage.

III.6.1 Choix de la fréquence d'échantillonnage

A intervalle de temps régulier le microcontrôleur lit la position angulaire donnée par le capteur et met à jour ses variables internes de position et de vitesse. La mesure des erreurs (erreur de position et erreur de vitesse) se fait par une simple différence entre la valeur demandée et la valeur mesurée.

C'est la fréquence de rafraîchissement des erreurs et de recalcul de la correction qui joue un rôle significatif dans la précision de l'asservissement :

- Si la fréquence de rafraîchissement est trop faible, le trop grand écart entre deux calculs de correction opère une discrétisation du temps trop grossière, ce qui amoindrira la précision de l'asservissement ;
- Si la fréquence de rafraîchissement est trop élevée, bien que l'on se rapproche du modèle de temps continu la précision de l'asservissement s'amoindra puisque le nombre de pas comptés par le capteur entre deux calculs de correction sera plus faible, donc moins précis.

Dans le choix de la fréquence d'échantillonnage, un point a été impérativement à respecter : c'est le théorème de Shannon [27]. Ce théorème stipule que la fréquence d'échantillonnage f_e doit valoir au moins deux fois la fréquence maximale f_{max} contenue du système à échantillonner.

$$f_e \geq 2f_{max} \quad \text{(III.7)}$$

Le modèle physique calculé au point II.3.3 a permis de tracer le diagramme de Bode et d'estimer que la fréquence de coupure $f_c = 15$ Hz.

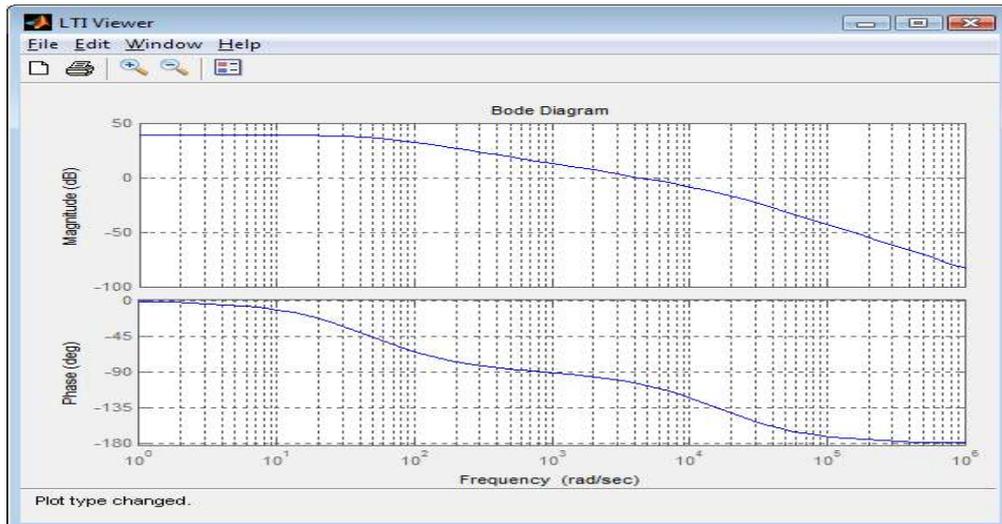


Figure III.15 diagramme de Bode du model physique du moteur réducteur.

Finalement, Une fréquence de 100Hz a été jugée empiriquement comme un assez bon compromis entre période d'échantillonnage et précision de chaque calcul lors des expériences.

III.6.2 Calculs et approximations

III.6.2.1 Calcul approximé de la dérivée

On estime le nombre dérivé en tenant compte des deux dernières mesures par une approximation au premier ordre de la fonction erreur à l'instant considéré. Étant donné que Δt (temps entre deux mesures, $T_e = 10\text{ms}$ à 100Hz) est assez petit par rapport à la vitesse de variation du système, on considère qu'il s'agit là d'une approximation convenable.[9]

$$\frac{d}{dt} \varepsilon(t) = \lim_{\Delta t \rightarrow 0} \frac{\varepsilon(t) - \varepsilon(t - \Delta t)}{\Delta t} \simeq \frac{\varepsilon(t) - \varepsilon(t - \Delta t)}{\Delta t} \quad (\text{III.8})$$

Remarque: Pour éviter de surcharger le microcontrôleur avec une division à 100Hz, on inclue le terme $\frac{1}{\Delta t}$ directement dans le terme k_d paramétré.

III.6.2.2 Calcul approximé de l'intégrale

On utilise la formule de Riemann pour obtenir une approximation de l'intégration de l'erreur par la somme discrète des erreurs mesurées [21]:

$$\int_a^b f(t) dt = \lim_{n \rightarrow \infty} \frac{b-a}{n} \sum_{k=0}^n f\left(a + k \cdot \frac{b-a}{n}\right) \Rightarrow \int_0^t \varepsilon(\tau) d\tau =$$

$$\lim_{\Delta t \rightarrow 0} \Delta t \sum_{k=0}^{t/\Delta t} \varepsilon(k \cdot \Delta t) \simeq \Delta t \sum_{k=0}^{t/\Delta t} \varepsilon(k \cdot \Delta t)$$

(III.9)

Remarque: $a = 0$, $b = t$, $n = \frac{t}{\Delta t}$

On se contente donc de sommer l'ensemble des erreurs mesurées, ce qui donne le terme $\sum_{k=0}^{t/\Delta t} \varepsilon(k \cdot \Delta t)$. Le terme Δt sera déjà inclus logiciellement dans le paramètre k_i pour éviter de surcharger le microcontrôleur avec une multiplication supplémentaire à 100Hz.

III.6.2.3 Calcul du PID

A chaque nouvelle mesure de l'erreur, on recalcule la consigne corrigée à envoyer aux moteurs après avoir effectué les calculs préliminaires de dérivation et d'intégration[9] :

$$u(t) = k_p \cdot \varepsilon(t) + k_i \cdot \int_0^t \varepsilon(\tau) d\tau + k_d \cdot \frac{d}{dt} \varepsilon(t)$$

(III.10)

$$u(p) = k_p \cdot \varepsilon(p) + k_i \cdot \frac{\varepsilon(p)}{p} + k_d \cdot p \cdot \varepsilon(p) = \varepsilon(p) \cdot [k_p + k_i \cdot \frac{1}{p} + k_d \cdot p] \quad \text{(III.11)}$$

III.7 Mise au point de l'asservissement – Résultats

Étant donné que les moteurs à notre disposition sont des moteurs-réducteurs hauts de gamme (Maxon) généralement utilisés pour les asservissements nécessitant de la précision, la plupart des paramètres (inductance, constante de couple, ...) sont donnés par le constructeur.

Nous avons donc appliqué le paramètre PID calculé grâce à la simulation sous Matlab effectuée au chapitre II. Cela dit, ces paramètres devront être optimisés empiriquement car certains paramètres n'ont pas été pris en compte.

Le choix du modèle de moteur peut conduire à certaines approximations :

- Le jeu mécanique des pièces notamment des réducteurs ;
- L'usure des pièces n'est pas prise en compte ;

- Le modèle n'est valable que pour une plage de tension et de courant au-delà de laquelle le comportement et la durée de vie du moteur sont incertains ;
- le temps de retard induit par les capteurs.

L'implantation logicielle du PID peut elle aussi induire certaines approximations :

- Approximation des calculs du nombre dérivé et de l'intégrale logicielle ;
- Approximation en considérant le temps comme discret et non continu (fréquence de calcul de la correction) ;
- Approximation au niveau des capteurs (nombre de pas des encodeurs, mauvais comptage de pas).

D'autant plus que la méthode choisie pour régler les paramètres du PID (méthode de Ziegler nicols) n'est pas des plus précises. Néanmoins, ces premiers réglages ont donné des résultats satisfaisants, et peuvent servir de point de départ. Les paramètres pouvant être ajusté selon le cahier des charges (précision, stabilité, rapidité).

Tableau III.1 coefficients des termes PID calculés à partir de la méthode ZN

kp	0.013
ki	0.9
kd	0.00004

Dans un asservissement PID, le terme Dérivation peut parfois poser problème. En effet, prenons le cas d'un système fortement bruité comme un asservissement de la vitesse d'un moteur. Si on dérive la vitesse on obtient l'accélération or celle-ci peut s'avérer très instable si la vitesse mesurée est trop bruitée. Comme illustré ci-dessous, l'asservissement est alors fortement altéré et inutile.

Deux solutions sont alors envisageables :

- La première consiste à effectuer un filtrage à l'aide d'un filtre passe-bas afin de limiter le bruit. Cette solution peut parfois poser problème car un filtrage entraîne une perte d'information qui peut nuire à l'asservissement ;
- La seconde est celle que nous avons retenue il s'agit d'enlever le paramètre Dérivé de l'asservissement, on obtient donc un asservissement PI. C'est cette dernière solution qui est la plus souvent utilisée. Il est intéressant de noter que ce type de problème

intervient essentiellement dans le cas d'asservissement en vitesse. Dans la plupart des autres systèmes, la dérivée est peu bruitée permettant un asservissement d'autant plus efficace.

Avant de présenter les résultats, il est important de préciser que les données recueillies ont été filtrées grâce à un filtre passe-bas de dimension quatre afin de procurer plus de clarté dans les résultats.

On prend comme consigne $v = 500\text{rad/s}$

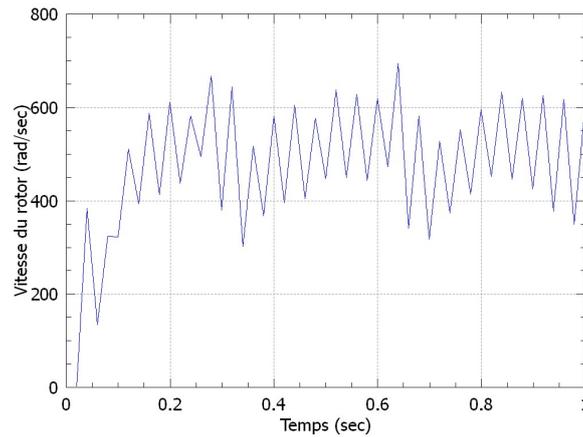


Figure III.16 évolution de la vitesse du rotor pour une correction PID

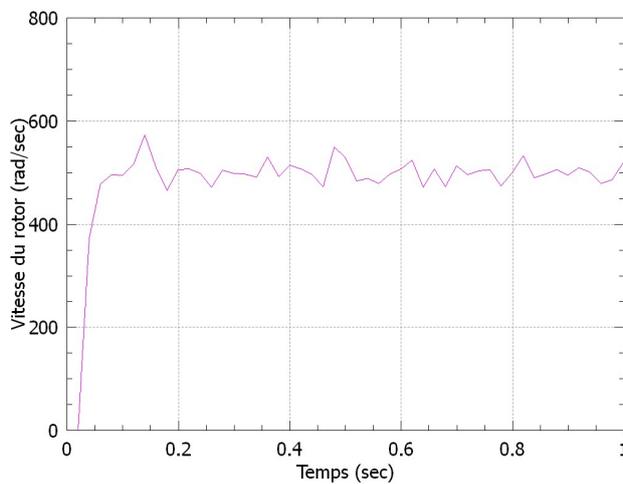


Figure III.17 évolution de la vitesse du rotor pour une correction PI

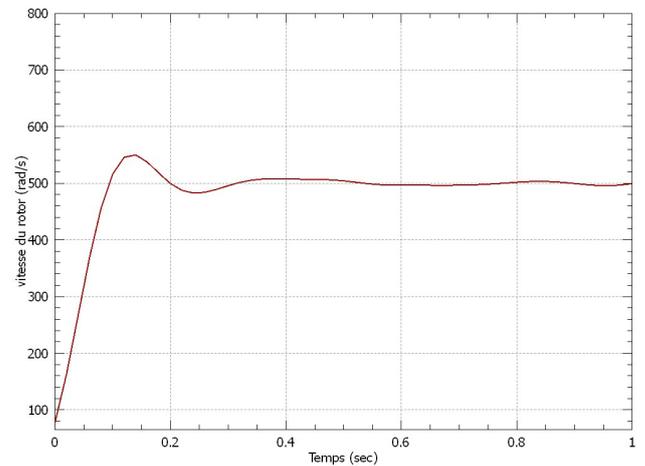


Figure III.18 résultat du filtrage de la vitesse pour une correction PI

Les courbes de résultats sont présentés en annexe 1 et elles présentent les résultats d'une série de tests pour différentes vitesses désirées. Les résultats sont jumelés aux résultats obtenus en simulation. Les gains des contrôleurs de moteurs utilisés pour ces essais étaient les suivants: $k_p = 0.015$ et $k_i = 0.4$.

Ces gains ont été ajustés par essais et erreurs puisque l'utilisation de gains calculés de façon analytique ne donnait pas les performances attendues. Cela s'explique principalement par les nombreuses non-linéarités qui sont présentes et qui tendent à détériorer la réponse de ce dernier. D'ailleurs, il est assez difficile d'observer une concordance évidente entre la réponse du modèle de simulation et la réponse du système réel puisque de nombreuses caractéristiques non-linéaires ne sont pas présentes dans le modèle de simulation (discrétisation des encodeurs, zone morte du moteur, imprécisions mécaniques, perturbations ; des gains retenus permettent néanmoins d'obtenir une réponse qui soit précise et rapide et qui comporte le moins de bruit possible.[22]

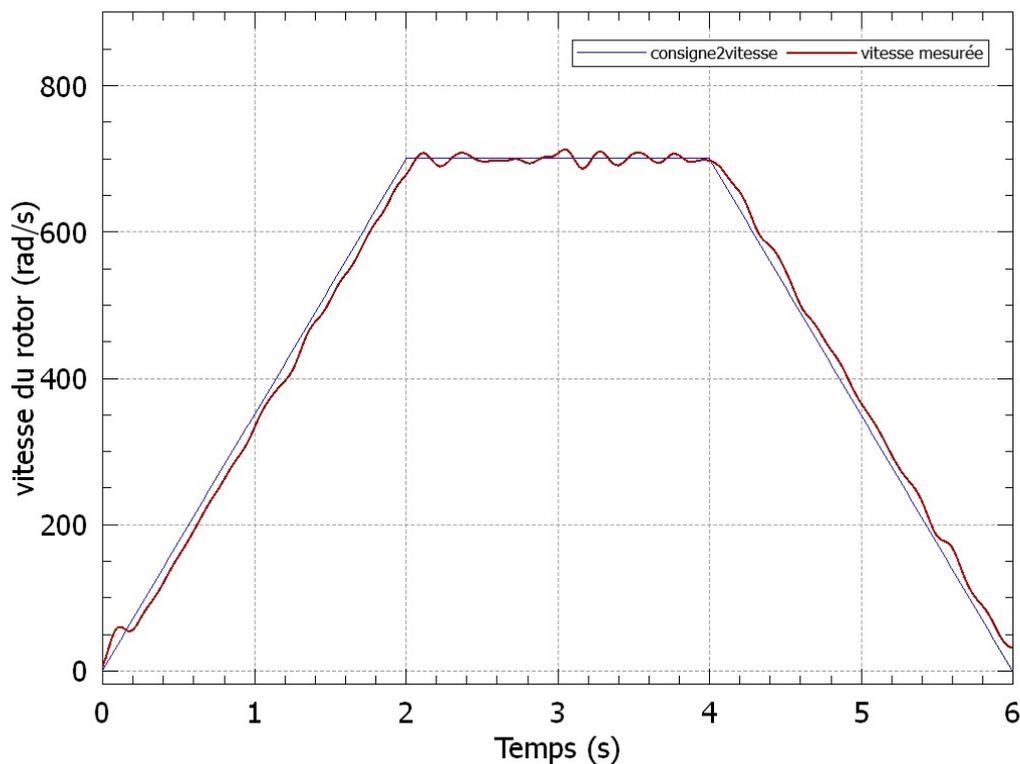


Figure III.19 asservissement de vitesse avec profil trapézoïdale

La (figure III.19) montre un asservissement de vitesse avec profil trapézoïdale. On parle dans ce cas d'asservissement en poursuite plutôt qu'en régulation. On remarque l'erreur dynamique appelée erreur de trainage entre la consigne et la vitesse du rotor. Cette erreur peut être diminuée en augmentant l'action de l'intégrale, mais celle-ci engendre moins de stabilité en régime établi. On constate donc que le réglage d'un PID en régulation n'est pas le même qu'en poursuite. Généralement, on tente de respecter les deux modes de fonctionnement et on choisit un réglage moyen.

III.8 Conclusion

Les résultats de la mise en œuvre de la commande PID d'un moteur à courant continu nous ont démontré que ce régulateur pouvait offrir de bonnes performances pour ce type de système (système linéaire), que ce soit au niveau de la précision, du temps de réaction, du dépassement, ou de la stabilité. Cependant le régulateur PID peut s'avérer moins efficace pour certains systèmes qui contiennent du bruit (coefficient dérivé) et il est essentiel de s'éloigner le moins possible du modèle théorique pour qu'il reste valable.

La méthode de réglage des paramètres du PID utilisée (Ziegler & Nicols) a permis d'obtenir un bon point de départ pour l'estimation des coefficients ces derniers pouvant être optimisés pour de meilleures performances. Cela dit le régulateur idéal n'existe pas, chaque optimisation d'un paramètre (précision, ...) se fait aux dépens d'un autre. Il convient alors de trouver le meilleur compromis en fonction des exigences initiales du cahier de charge. Divers éléments, tels que le choix de la méthode d'implantation du PID, peuvent aussi avoir des conséquences sur les performances.



Chapitre IV

Mise en œuvre du système de locomotion

IV.1 Introduction

Dans ce chapitre, le but est de proposer un module qui permettra de gérer les déplacements d'une base robotique du même type que celle utilisée pour le concours Eurobot.

Il s'agira essentiellement de la conception et la réalisation de notre système de locomotion avec la solution choisie dans le chapitre (I) : un robot unicycle enchainant des lignes droites et des rotations symétriques.

Le module se présentera comme une boîte noire servant d'interface entre un utilisateur et les moteurs (figure IV.1). Au niveau de l'utilisateur, la communication est effectuée au travers d'un protocole de communication standard.

Les commandes envoyées sont de haut niveau :

- Distance à parcourir ;
- Vitesse et accélération à imposer au robot ;
- Angle à effectuer.

De même, les informations reçues seront affichées :

- Position actuelle du robot ;
- Vitesse d'avance ;
- Vitesse de rotation.

L'utilisateur dont il est question pourra être de diverses natures. Il pourrait être dans le cas du concours un PC embarqué ou un microcontrôleur. Mais on pourrait également imaginer interagir avec le robot depuis le logiciel Matlab ou simplement envoyer des commandes et recevoir des données sur le terminal d'un ordinateur fixe.

Le module communique non seulement avec l'utilisateur, mais aussi avec les moteurs. Il doit appliquer des tensions adaptées à l'entrée des moteurs en fonction des ordres haut niveau reçus, et traduire les signaux des encodeurs (quadrature) en informations plus parlantes pour l'utilisateur (position actuelle, vitesse courante).

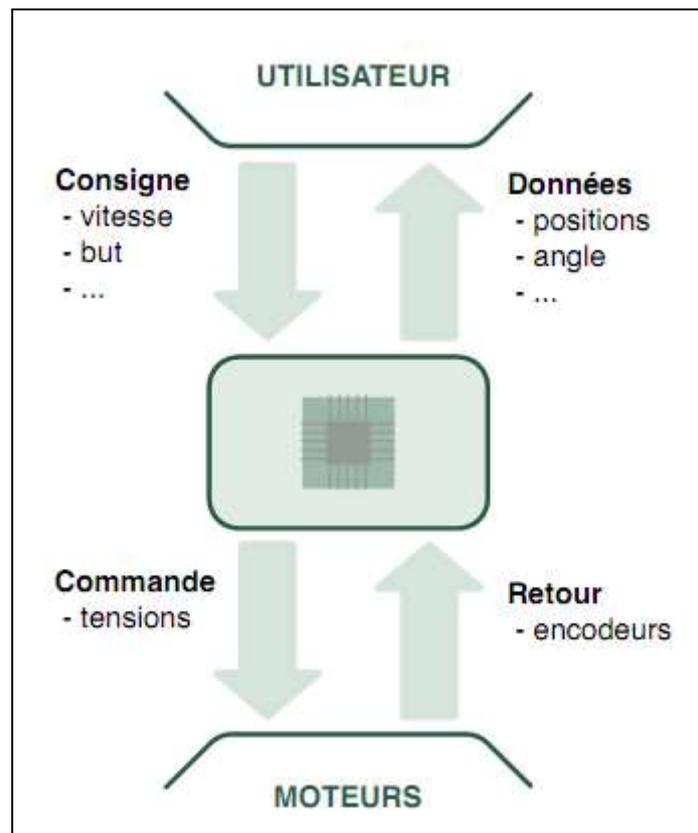


Figure IV.1 Le module développé servira d'intermédiaire entre l'utilisateur et les Moteurs

IV.2 Conception de la base Robotique

IV.2.1 Objectifs

Notre objectif est de concevoir une base robotique dédiée à la gestion des déplacements pour le développement de la commande que nous avons choisi d'appliquer et qui pourra aussi servir à tester d'autres algorithmes de commande dans le cadre d'un futur projet.

La partie hardware du projet a été développée en gardant à l'esprit les deux points suivants:

- **Adaptabilité au concours Eurobot :** La taille de la base de développement sera similaire à celle du robot disputant le concours, les moteurs seront similaires et les signaux renvoyés par les encodeurs devront être les mêmes, afin de pouvoir utiliser les mêmes schémas électroniques.

- **Simplicité et faisabilité** : La mécanique devra être le plus simple possible, afin d'être réalisable dans le temps à disposition. Les pièces devront pouvoir être fabriquées dans l'atelier de mécanique disposant de moyens classiques. Les composants électroniques seront choisis de telle sorte qu'ils soient disponibles sur le marché local et soudable à la main.

IV.2.2 Partie mécanique

Le robot utilisé est de type unicycle et la mécanique de sa plate-forme mobile est relativement simple.

La forme globale du robot est un cylindre d'une trentaine de centimètres de diamètre et la locomotion est assurée par deux roues motrices diamétralement opposées, elles sont placées dans un block pour garantir un meilleur alignement. Les roues motrices ont un diamètre de huit centimètres et sont directement fixées sur les réducteurs des moteurs. La stabilité est assurée par deux roues bille placées à l'avant et à l'arrière de la base du robot.

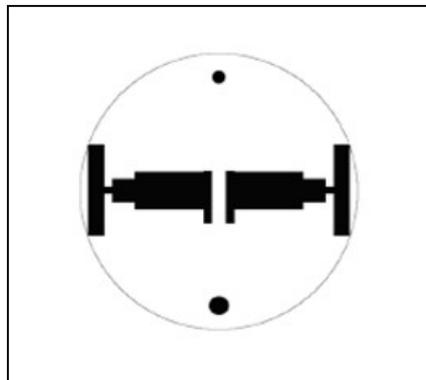


Figure IV.2 Vue de la disposition mécanique du robot utilisé

Il a fallu élaborer un système pour éviter que le robot ne se perde si les roues dérapent. En effet en utilisant des codeurs directement sur les axes moteurs, ces derniers pourraient renvoyer de fausses informations sur la vitesse et la position réelles du robot en cas de glissement.

L'idée est de placer les codeurs sur des axes différents montés sur ressort pour coller au sol même si une des roues du robot décolle un peu.

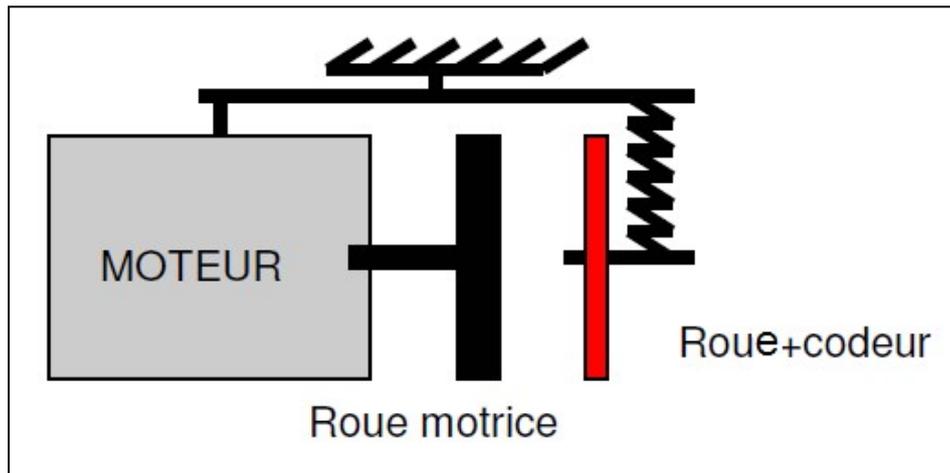


Figure IV.3 Mise en place d'une roue codeuse

Les deux moteurs utilisés sont identiques, à courant continu 12V de 6W et le facteur de réduction est de 1 : 84 avec un rendement de 72%.

Ces moteurs sont de grande qualité provenant du constructeur Maxon (110160). On voit dans le datasheet (annexe 2) que le couple nominal du moteur est de 6.77 mNm et que sa vitesse nominale est de 10200 rpm. Avec des roues de 8cm de diamètre et un robot de 5 kilos, on pourra donc espérer :

$$V_{max} = \frac{10200rpm}{84*60} * \pi * 8cm = 0.5 m/s \quad (IV.1)$$

$$C_{roue_{max}} = 72\% * 84 * 6.77mNm = 0.409Nm \quad (IV.2)$$

$$F_{roue_{max}} = \frac{C_{roue_{max}}}{r} = \frac{0.409Nm}{4cm} = 10.23N \quad (IV.3)$$

$$Acc_{max} = \frac{2F_{roue_{max}}}{masse_{robot}} = \frac{2*10.23N}{5kg} = 4.09 m/s^2 \quad (IV.4)$$

V_{max} : vitesse maximale de la roue a vide $F_{roue_{max}}$: force maximale exercée par la roue
 Acc_{max} : accélération maximale du robot $C_{roue_{max}}$: couple maximal de la roue

IV.2.3 Partie électronique

IV.2.3.1 Puissance

L'électronique de puissance doit être en mesure de commander un moteur en boucle ouverte ou en boucle fermée (régulation en vitesse ou en position), en utilisant le principe du pont H comme présenté dans la section précédente. Le moteur à commander est un moteur à courant continu 12V de 6W, ses caractéristiques donnent 2.09A comme courant de démarrage et 0.7A comme Courant nominal.

Le schéma général est connu, il convient de sélectionner correctement les composants nécessaires à sa réalisation.

L'utilisation d'un pont H intégré (tout en un) a été retenue. Trois critères ont dicté ce choix :

- Possibilité de trouver des composants intégrés permettant des courants nominaux au dessus de la limite de consommation de nos moteurs ;
- Une liberté sur la commande des éléments du pont ;
- Une bande passante relativement élevée.

Le pont H choisi est le L298 . Il a été retenu pour les caractéristiques suivantes[18] :

- Support des PWM (jusqu'à 40 kHz);
- Gamme de puissance adaptée à l'application;
- Deux pont H sur un seul circuit intégré.

Pour commander les sorties OUT1 et OUT2 du L298, nous devons agir sur les entrées ENABLE, IN1 et IN2.

Tableau IV.1 Effet des entrées sur le pont H

ENABLE	Cette entrée permet d'activer (1) ou désactiver (0) le pont H
IN1	Cette entrée commande la sortie OUT1
IN2	Cette entrée commande la sortie OUT2

Nous voulons contrôler la vitesse du moteur au moyen d'une PWM. Pour ce faire, il y a deux possibilités:

1. Nous pouvons envoyer les PWM sur le bit ENABLE. De cette façon l'alimentation du moteur est coupée quand ce bit est à zéro. Le sens est, dans ce cas, contrôlé par les bits IN1 et IN2, l'un devant être à un niveau logique haut, et l'autre à zéro.

2. Le bit ENABLE est toujours à VCC (le pont-H est toujours activé), et les PWM sont envoyés soit sur l'une des entrées (IN1 ou IN2) l'autre étant à zéro. Le sens est contrôlé en choisissant l'entrée à laisser à zéro. Ici, c'est la deuxième solution qui a été choisie. Le microcontrôleur utilisé est prévu pour ce type de fonctionnement en proposant deux sorties pour un même signal PWM et la possibilité de choisir facilement sur laquelle des deux sorties la PWM doit être générée. On peut ainsi facilement inverser le sens de rotation du moteur en modifiant la sortie PWM. Mais par sécurité, il faut adjoindre des diodes.

En effet, dans le mode *Locked Anti-phase*, les diodes de roue libre ne doivent conduire du courant que pendant les phases de commutation des transistors. Les diodes devront réagir le plus rapidement possible pour limiter au maximum les pics de tension inverses aux bornes du moteur pendant ces courtes périodes.

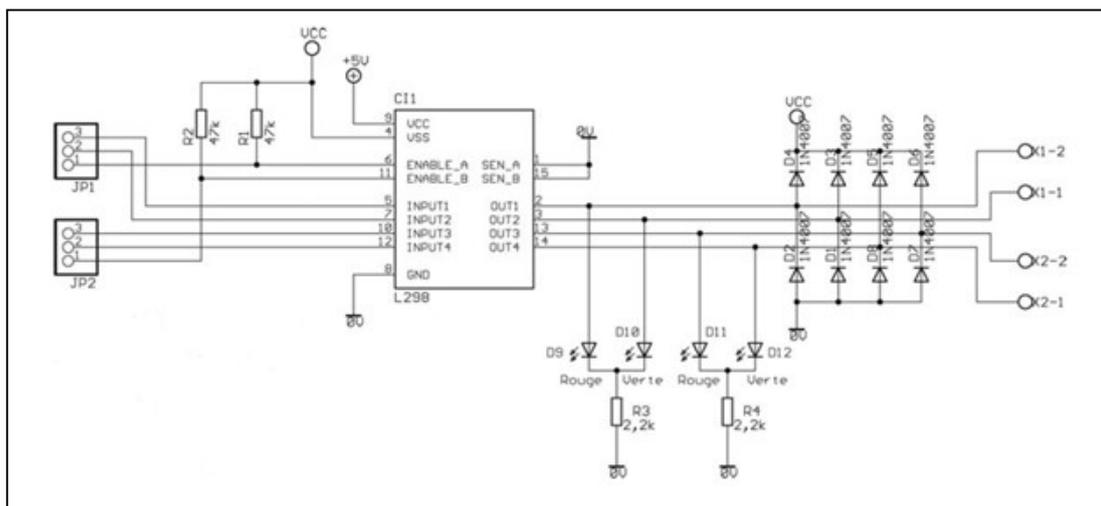


Figure IV.4 schéma de la carte de puissance

IV.2.3.2 Logique

Une fois le design de la partie puissance réalisé, il reste à élaborer celui de la partie logique qui va être chargée de commander le pont H. Ceci passera par la sélection d'un microcontrôleur et d'un certain nombre de capteurs. La partie logique doit être en mesure d'effectuer un certain nombre de tâches :

- Générer des signaux PWM en mode *edge_aligned* à une fréquence de l'ordre d'une vingtaine de KHz avec une résolution acceptable du rapport cyclique;
- Gérer un signal d'arrêt d'urgence;
- Effectuer des conversions analogiques-numériques pour la mesure du courant consommé et de la tension de l'alimentation ou de mesure de distance avec des télémètres infrarouges ou ultrasons;
- Récupérer les informations d'un encodeur de quadrature;
- Effectuer une régulation numérique de type PID;
- Gérer le protocole I2C un Protocol de communication série UART.

IV.2.3.3 Le microcontrôleur

Pour effectuer toutes les tâches nécessaires, notre choix s'est porté sur un microcontrôleur de chez AVR de la famille Atmel : l'atmega2560.[15]

Les caractéristiques de ce microcontrôleur répondent à toutes nos attentes. Nous citerons par exemple :

- 54 pins d'entrées sortie numérique/analogique;
- un module PWM de plusieurs canaux (14);
- un module pour la conversion analogique-digital sur 10 bits de 16 canaux, pouvant effectuer plusieurs conversions simultanées;
- 4 unités de capture;
- un module I2C maître/esclave et 4 ports de communications série UART;
- de nombreux Timers ;
- une gestion des interruptions interne et externe;
- 256 bytes de ROM et 8 kbytes de SRAM;
- Le microcontrôleur effectue 16MIPS (millions d'instructions par seconde).

Le microcontrôleur est utilisé sur un système Arduino qui procure une grande souplesse d'utilisation.

Introduction au module Arduino

Le système Arduino est une plateforme open-source d'électronique programmée et basée sur une simple carte à microcontrôleur (de la famille AVR) et un logiciel. Ce dernier est un véritable environnement de développement intégré pour écrire, compiler et transférer le programme vers la carte à microcontrôleur. Arduino peut être utilisé pour développer des objets interactifs, pouvant recevoir des entrées d'une grande variété d'interrupteurs ou de capteurs, et pouvant contrôler une grande variété de lumières, moteurs ou toutes autres sorties matérielles. Les projets Arduino peuvent être autonomes comme ils peuvent communiquer avec des logiciels tournant sur un ordinateur (tels que Flash, Processing, MaxMSP). Les cartes électroniques peuvent être fabriquées manuellement ou être achetées pré-assemblées; le logiciel de développement open-source peut être téléchargé gratuitement. Le langage de programmation Arduino est une implémentation de Wiring, une plateforme de développement similaire, qui est basée sur l'environnement multimédia de programmation Processing.[23]

IV.2.3.4 Les capteurs

En robotique mobile, on classe traditionnellement les capteurs en deux catégories, selon qu'ils mesurent l'état du robot lui-même ou l'état de son environnement. Dans le premier cas, à l'image de la perception chez les êtres vivants, on parle de proprioception et donc de capteurs proprioceptifs. On trouve par exemple dans cette catégorie les capteurs de position ou de vitesse des roues et les capteurs de charge de la batterie. Les capteurs renseignant sur l'état de l'environnement, donc de ce qui est extérieur au robot lui-même, sont eux appelés capteurs extéroceptifs. Il s'agit de capteurs donnant la distance du robot à l'environnement, la température signalant la mise en contact du robot avec l'environnement... etc.[2]

IV.2.3.5 Les encodeurs en quadrature

Procédons à une description du capteur. L'encodeur en quadrature montée sur nos roues codeuse est un HEDL-5540 de 500 pas. Celui-ci permet d'obtenir une information sur la position du moteur de manière incrémentale. En effet, si on connaît le nombre d'impulsions par tour, il suffit de compter le nombre d'impulsions données par le capteur pour connaître la position relative du moteur par rapport à sa position de départ.

L'encodeur est dit de quadrature, parce qu'il possède deux sorties A et B déphasées de 90° . Nous avons quatre états différents pour les deux signaux, qui s'enchainent dans un certain ordre suivant le sens de rotation.

On peut décoder ces deux signaux pour générer les impulsions et pour connaître le sens de rotation du moteur. La résolution lors du décodage sera quatre fois supérieure au nombre de pas du capteur (pour chaque pas nous avons quatre transitions d'état), et on pourra espérer pour le HEDL-5540 une précision au 1/2000ème de tour.

IV.2.3.6 Courant consommé

Afin d'être en mesure de détecter une situation dangereuse pour le moteur, comme son blocage, nous allons faire une mesure du courant consommé.

Le circuit L298 possède une sortie (sens) qui fournit le courant consommé par le moteur en plaçant une résistance de puissance de faible valeur (1 ohm) entre cette sortie et la masse, nous aurons une tension proportionnelle au courant le traversant. La sortie est ensuite connectée à l'entrée d'un convertisseur analogique-digital du microcontrôleur. L'information fournie par le capteur sera largement suffisante pour nous permettre de vérifier le courant consommé par le moteur.

Pour éviter d'endommager le microcontrôleur, une diode Zener d'une valeur de 5 V est placée en parallèle avec la résistance.

IV.2.3.7 Mesure de proximité et de distance

On appelle télémétrie toute technique de mesure de distance par des procédés acoustiques, optiques ou radioélectriques. L'appareil permettant de mesurer les distances est appelé télémètre. De même qu'il existe différentes techniques de mesure de distance (mesure du temps de vol d'une onde, triangulation), il existe différentes technologies pour

réaliser des télémètres. Sur notre robot, nous en avons utilisé deux types de capteur: infrarouge et ultrason.[2]

➤ Capteurs infrarouges

Les capteurs infrarouges sont constitués d'un ensemble émetteur/récepteur fonctionnant avec des radiations non visibles, dont la longueur d'onde est juste inférieure à celle du rouge visible. La mesure des radiations infrarouges étant limitée et, en tout état de cause, la qualité très dégradée au delà d'un mètre, ces dispositifs ne servent que rarement de télémètre. On les rencontrera le plus souvent comme détecteurs de proximité. Il faut noter que ce type de détection est sensible aux conditions extérieures, notamment à la lumière ambiante, à la spécularité des surfaces sur lesquelles se réfléchissent les infrarouges, à la température et même à la pression ambiante. Ces capteurs ne sont pas complètement directionnels et leur caractéristique présente une zone de détection conique à l'origine d'incertitudes. Enfin, l'alternance de phases d'émission et de réception impose une distance de détection minimale.

Les capteurs infrarouges Sharp de la famille GP2 (voir figure IV.5) sont des télémètres de portée réduite. Ces modules permettent de mesurer des distances absolues ou relatives par détection infrarouge et disposent, selon les modèles, de sorties numériques ou analogiques.

La mesure est établie à partir de l'inclinaison du rayon réfléchi : le rayon est en effet dévié à l'aide d'un système optique sur une matrice de photodiodes (PSD), comme cela est représenté dans la figure IV.5. Sur des robots mobiles, ces capteurs ne sont généralement pas utilisés pour repérer l'ensemble des corps présents dans l'environnement car leur portée est trop faible. Ils servent plutôt à mesurer la proximité de corps proches du robot.[2]

Sur notre robot nous disposant de capteur Sharp GP2Y0A02YK, qualifié de télémètre longue distance, il travaille entre 10 et 150 cm, grâce à une carte de prétraitement nous pouvons déclencher un signal d'arrêt d'urgence pour différentes distances de proximité d'un obstacle grâce à un simple potentiomètre.

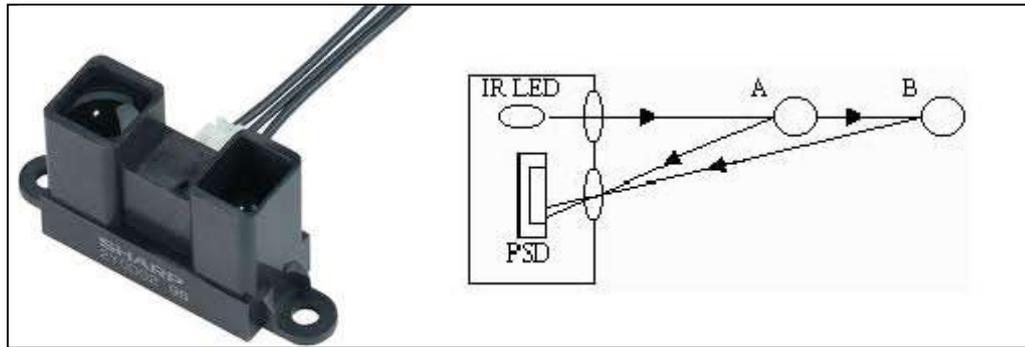


Figure IV.5 capteur infrarouge Sharp GP2Y0A02YK

IV.2.3.8 Isolation galvanique

Il est nécessaire de protéger la partie logique de l'étage de puissance par une isolation galvanique. Cette isolation entre deux circuits électroniques fait que le courant qui circule dans le premier circuit, ne circule pas dans le deuxième circuit. L'utilité d'une pareille isolation arrive quand on veut protéger un circuit (comme un microcontrôleur) d'une surtension qui pourrait provenir d'un deuxième circuit. Dans notre cas, nous avons d'un côté le microcontrôleur, et de l'autre des cartes électroniques contrôlant des moteurs et nécessitant une alimentation externe.

L'isolation galvanique nous protège de deux sources de problèmes principaux:

- Si l'alimentation externe qui donne la puissance aux moteurs n'est pas isolée du réseau, en connectant la masse de l'alimentation à celle de microcontrôleur, on court un risque. L'isolation galvanique permet de ne pas devoir connecter les masses ensembles et ainsi protéger le microcontrôleur.
- Les moteurs sont une source de parasites qui peuvent être très importants. L'isolation galvanique nous permet d'éviter que ces parasites remontent dans le microcontrôleur.

La méthode la plus répandue pour protéger un circuit d'un autre sans bloquer la transmission de signaux est d'utiliser un optocoupleur. Dans notre cas, un TLP521-4. L'optocoupleur possède d'un côté une diode émettrice (non visible, elle est intégrée dans la puce). De l'autre côté, nous trouvons un phototransistor (ou équivalent). Grâce à ce jeu optique la présence d'un signal est transmise sans mélanger les électrons.

Les entrées de l'optocoupleur sont relié aux sorties PWM et ENABLE du microcontrôleur et leurs sorties sont reliées aux entrées de commande du circuit L298.

IV.2.3.9 Afficheur LCD

Pour afficher différents messages renseignant sur l'état du système comme la charge de la batterie ou des variables comme la position ou la vitesse, un afficheur LCD 2*16 est placé sur la partie supérieure du robot.

IV.2.3.10 L'alimentation

L'énergie est fournie par une batterie en Pb de 12V. Une carte distribue la tension d'entrée aux autres cartes, après l'avoir régulée. L'étage de puissance nécessite du 12V, la logique et l'afficheur LCD fonctionnent à 5V. Pour ce faire, des régulateurs sont utilisés pour passer du 12V au 5V, des fusibles sont placés avant chaque régulateur pour protéger les cartes d'une éventuelle court circuit et des LCD sont placés pour montrer l'état des cartes (on/off).

Avec notre pont en H, la tension aux bornes du moteur est proportionnelle au rapport cyclique de la PWM. Cela nous permet de fixer un pourcentage de la tension d'alimentation aux bornes du moteur. La réponse de notre contrôleur PID aurait été en fonction de la charge de la batterie, ce qui aurait été gênant. Pour effectuer notre mesure, nous allons utiliser un simple diviseur potentiométrique en entrées d'un convertisseur analogique-digital du microcontrôleur. Grâce à cette mesure, et connaissant l'état de la batterie, un message pourra par exemple afficher si elle vient à se décharger.

IV.2.3.11 Résultat

L'électronique étant constituée de plusieurs éléments, ils sont regroupés sur un seul circuit imprimé sur deux couches (carte mère). Cette solution est très efficace pour éviter les pannes et les erreurs inutiles dues à un mauvais câblage. Cette carte est présentée en l'annexe 2.

IV.3 Mise en œuvre du control

IV.3.1 Démarche

La régulation du robot est un point central de ce travail. La commande du robot est continuellement ajustée en fonction d'un retour d'informations sur le comportement réel du robot. C'est le calcul de la vitesse instantanée de chaque roue du robot et l'estimation de la position du robot par odométrie qui constitueront ces informations. L'approche adoptée pour déterminer un régulateur a été de séparer le plus possible ce problème en plusieurs blocs moins complexes. Ceci facilite l'implémentation et favorise la réutilisabilité du travail effectué. Les principaux étages sont le calcul des vitesses et de la position, la régulation en vitesse, la régulation en position avec un profil de vitesse.

Cette partie commence par décrire le système à commander et propose une modélisation physique de ce dernier. C'est ensuite la régulation elle-même, les choix effectués sont traités. Finalement, la section consacrée à l'odométrie détaillera la méthode utilisée pour l'estimation de la position du robot.

IV.3.2 Système à commander

IV.3.2 .1 Approche multivariable

Le système à commander est un système MIMO à deux entrées et deux sorties. Les entrées sont les tensions appliquées sur les moteurs, et les sorties sont les vitesses des roues:

$$\mathbf{u} = \begin{pmatrix} u_1 \\ u_2 \end{pmatrix} \quad \mathbf{y} = \begin{pmatrix} \Omega_1 \\ \Omega_2 \end{pmatrix}$$

Cependant, l'application d'une commande sur l'une des roues a très peu d'influence sur la position de la deuxième roue, car les masses et les inerties sont faibles. C'est la raison pour laquelle il est possible de considérer le système comme deux systèmes SISO.

IV.3.2 .2 Modèle physique

Cette modélisation ne reflète pas le système réel car elle est basée sur la connaissance de paramètres physiques difficiles à déterminer avec exactitude de plus de nombreux phénomènes ont été négligés notamment des phénomènes non-linéaires.

Le système à commander est un moteur à courant continu à aimants permanents. L'entrée est une tension et la sortie une vitesse.[24]

Hypothèses

- L'inductance du circuit d'induit est nulle ;
- Le frottement de l'air est négligeable ;
- Le frottement de roulement est négligeable ;
- Le poids du robot est également réparti sur les deux roues.

Équation du moteur à courant continu :

$$u(t) = R_i(t) + e(t)$$

(IV.5)

$$u(t) = R_i(t) + K_e \omega(t)$$

(IV.6)

Avec :

R: Résistance du circuit d'induit ;

i(t) : Courant d'induit ;

K_e: Constante du moteur ;

Ω(t) : vitesse angulaire.

Le couple du moteur est donné par :

$$\mathbf{C}_u(\mathbf{t}) = K_c \mathbf{i}(\mathbf{t}) \quad (\text{IV.7})$$

La loi de Newton conduit à l'équation suivante :

$$(k^2 J_{rotor} + J_{roue}) \dot{\Omega}(\mathbf{t}) = k \mathbf{C}_u(\mathbf{t}) - \mathbf{C}_r(\mathbf{t}) - \mathbf{C}_p(\mathbf{t}) \quad (\text{IV.8})$$

Avec :

J_{rotor} : Inertie du rotor ;

J_{roue} : Inertie de la roue ;

k : Rapport de réduction;

$\mathbf{C}_r(\mathbf{t})$: Couple résistant dû à la réaction du sol sur la roue lors des déplacements ;

$\mathbf{C}_p(\mathbf{t})$: Couple résistant parasite.

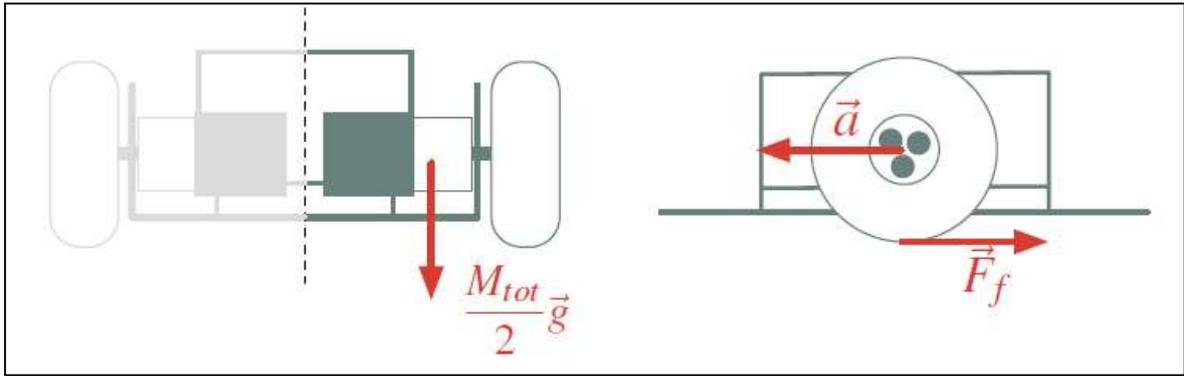


Figure. IV.6 On cherche à calculer la force F_f .

Le couple résistant $C_r(t)$ est obtenu en considérant la moitié du robot, comme illustré sur la figure IV.6. La deuxième loi de Newton permet d'obtenir la norme de la force F_f :

$$F_f = \frac{M_{tot}}{2} * a = \frac{M_{tot}}{2} * \dot{\Omega}(t) \cdot r \quad (\text{IV.9})$$

Le couple résistant C_r vaut donc :

$$C_r = F_f \cdot r = \frac{M_{tot}}{2} * \dot{\Omega}(t) \cdot r^2 \quad (\text{IV.10})$$

Pour obtenir C_u , on isole $i(t)$ dans l'équation (IV.6) que l'on insère dans (IV.7) :

$$C_u(t) = K_c \frac{u(t) - K_e \Omega(t)}{R} \quad (\text{IV.11})$$

C_p est un couple résistant parasite, il englobe par exemple le frottement de roulement, le frottement de l'air, les frottements dans la transmission, etc. Nous le considérons comme

une perturbation que le régulateur devra rejeter. C'est pourquoi nous posons pour notre modélisation :

$$\mathbf{C}_p = \mathbf{0} \quad (\text{IV.12})$$

On substitue les résultats (IV.10), (IV.11) et (IV.12) dans l'équation (IV.8) :

$$\dot{\Omega}(t) + \frac{kK_e^2}{\left(k^2J_{rotor} + J_{roue} + \frac{M_{tot}}{2} \cdot r^2\right)R} \Omega(t) = \frac{kK_c}{\left(k^2J_{rotor} + J_{roue} + \frac{M_{tot}}{2} \cdot r^2\right)R} u(t) \quad (\text{IV.13})$$

La fonction de transfert $G(s)$ du système s'obtient en effectuant la transformée de Laplace de (IV.13) :

$$\begin{aligned} G(s) = \frac{\Omega(s)}{U(s)} &= \frac{\frac{kK_c}{\left(k^2J_{rotor} + J_{roue} + \frac{M_{tot}}{2} \cdot r^2\right)R}}{s + \frac{kK_e^2}{\left(k^2J_{rotor} + J_{roue} + \frac{M_{tot}}{2} \cdot r^2\right)R}} \\ &= \frac{\frac{1}{K_c}}{\frac{\left(k^2J_{rotor} + J_{roue} + \frac{M_{tot}}{2} \cdot r^2\right)R}{kK_e^2} s + 1} = \frac{b}{\tau s + 1} \end{aligned} \quad (\text{IV.14})$$

Les valeurs numériques sont les suivantes :

$$J_{rotor} = 4.26 \cdot 10^{-7} \text{ [kg} \cdot \text{m}^2]$$

$$J_{roue} = 12 \cdot 10^{-5} \text{ [kg} \cdot \text{m}^2]$$

$$k = 84$$

$$K_c = 10.9 \cdot 10^{-3}$$

$$K_e = 10.9 \cdot 10^{-3}$$

$$M_{tot} = 5 \text{ [kg]}$$

$$r = 40 \text{ [mm]}$$

$$R = 5.74 \text{ [\Omega]}$$

La fonction de transfert devient donc :

$$G(s) = \frac{91.74}{4.098s+1} \quad (\text{IV.15})$$

IV.3.3 Principe de la commande

L'objectif de la commande de la base robotique est d'imposer un déplacement au robot aussi proche que possible de celui demandé.

Il est intéressant de séparer la commande du robot en deux niveaux distincts :

- Régulation de la vitesse des roues ;
- Régulation du déplacement du robot.

Les deux régulateurs sont placés en « cascade » comme illustré par la figure IV.7. Cette séparation simplifie l'implémentation du régulateur haut niveau, car on peut ainsi imposer au système bas niveau des consignes de la forme (v, ω) , correspondant à la vitesse d'avance et la vitesse angulaire du robot.

Cette approche simplifie également les développements futurs. On peut, en effet, changer de régulateur haut niveau, tout en réutilisant le travail fait au niveau de la régulation en vitesse.

De plus, lors du développement de la commande haut niveau, on pourra dans une certaine mesure, faire l'hypothèse que les consignes en vitesses sont appliquées directement sur les roues.

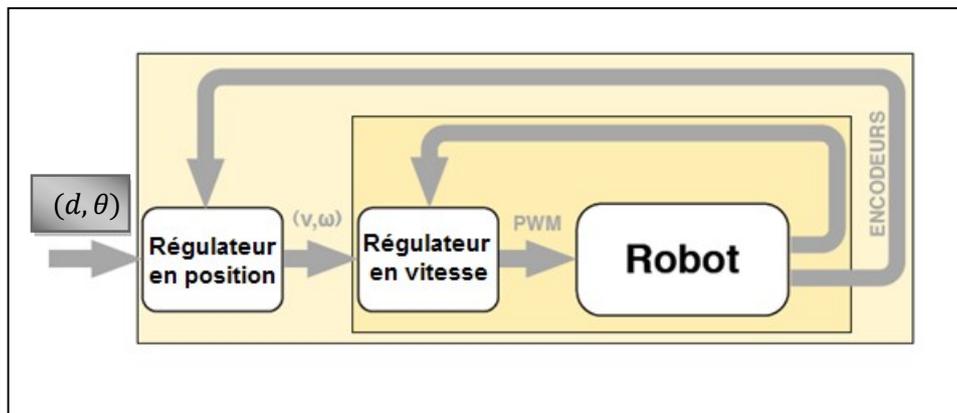


Fig. IV.7 Régulateurs haut et bas niveau en cascade.

D'après le modèle cinématique du robot unicycle présenté dans le chapitre (I), il s'agit d'appliquer un PID sur l'erreur d'orientation et l'erreur de distance pour générer la vitesse de consigne des roues [25].

En utilisant juste le coefficient P pour les 2 asservissements :

$$v_{\text{ref}} = DK_p * \text{Erreur_distance} \quad (\text{IV.16})$$

DK_p : Coefficient proportionnel du PID distance

Puis on écrête ces vitesses par :

v_{max} : La vitesse maximale en ligne droite.

A_{dmax} : L'accélération maximale en ligne droite

$$\omega_{\text{ref}} = \theta K_p * \text{Erreur_orientation} \quad (\text{IV.17})$$

θK_p : Coefficient proportionnel du PID orientation

Puis on écrête ces vitesses par :

ω_{\max} : La vitesse maximale en rotation.

$A_{\theta\max}$: L'accélération maximale en rotation.

Enfin on somme les 2 consignes calculées précédemment pour chaque roue :

$$v_{\text{droite}} = v - \frac{1}{2}L \omega \quad (\text{IV.18})$$

$$v_{\text{gauche}} = v + \frac{1}{2}L \omega \quad (\text{IV.19})$$

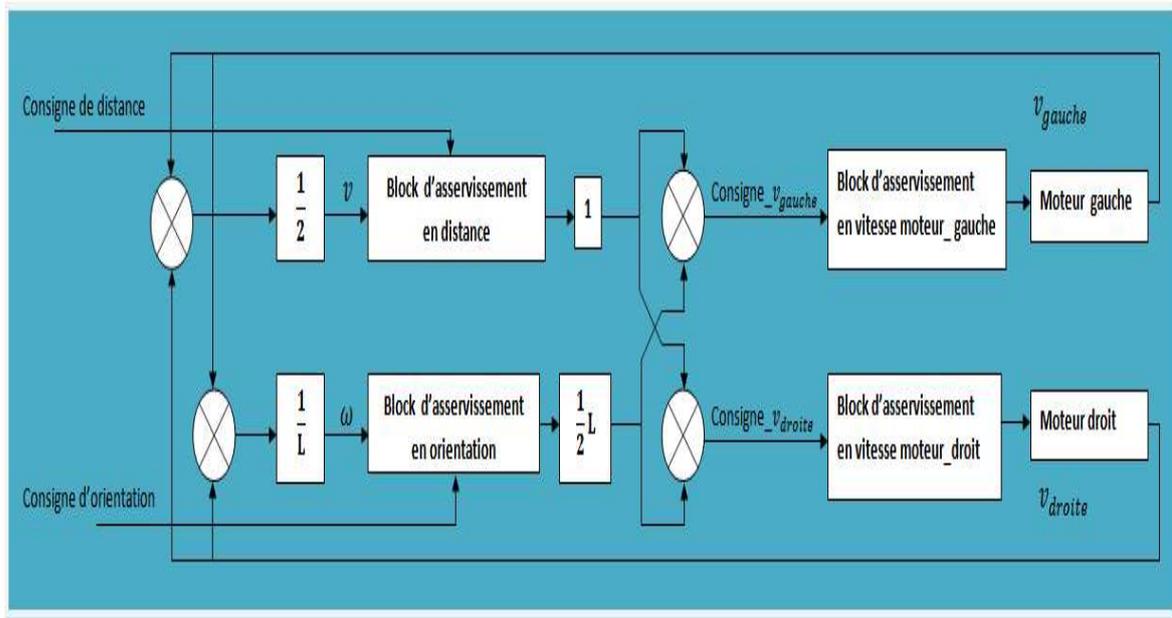


Figure IV.8 schéma bloc de la commande.

Si L'asservissement est bien réglé, le robot va réagir selon la combinaison de consigne appliquée :

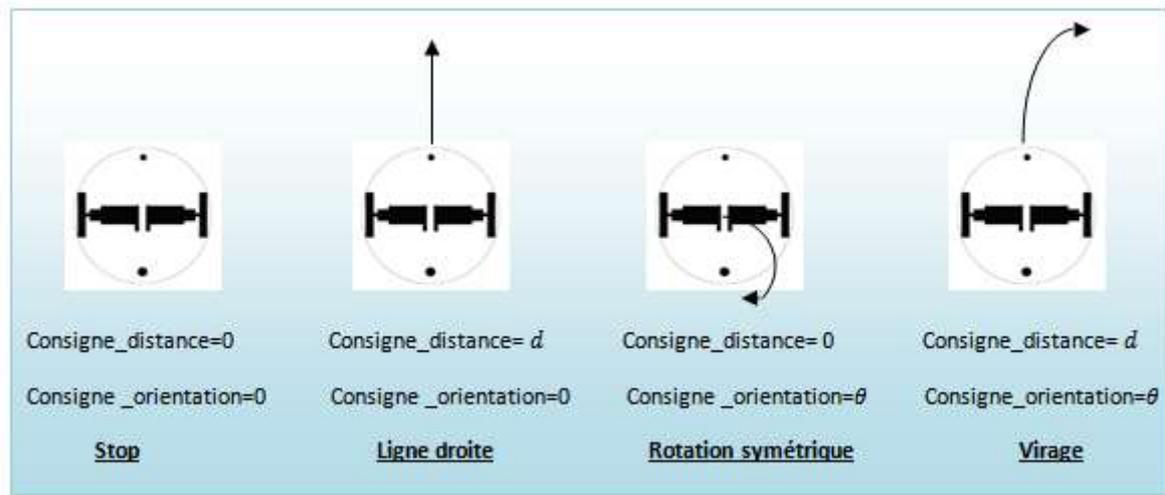


Figure IV.9 Les quatre Différentes combinaisons de déplacement possible.

Comme nous l'avons déjà souligné, la solution retenue pour notre système de locomotion est suite aux rotations et lignes droites, la quatrième configuration sera donc écartée de notre étude.

IV.3.3.1 Synthèse des régulateurs

➤ Fréquence d'échantillonnage

Pour respecter le théorème Shannon, il a été impératif de choisir une fréquence d'échantillonnage supérieure à 0.04 Hz d'après le diagramme de Bode. En pratique, l'idéale est de choisir une fréquence d'échantillonnage supérieure à 20 Hz. Cependant, il faut considérer les limitations du microcontrôleur qui doit pouvoir effectuer tous les calculs nécessaires à chaque période d'échantillonnage. Nous avons pu déterminer que le temps nécessaire à ces calculs était inférieur à 20 ms. Cela correspond à une fréquence de 50 Hz, Cela n'est pas problématique et permet même d'augmenter la précision du calcul de la position par odométrie.

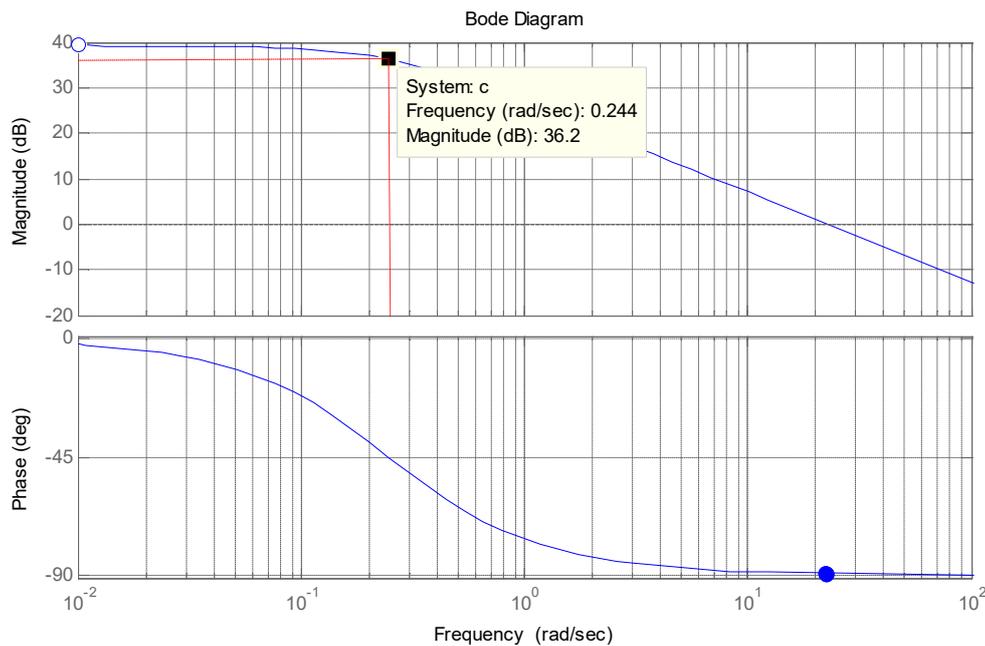


Figure IV.10 diagramme de Bode du modèle physique du robot

➤ Réglage des paramètres PID

La première étape consiste à régler les coefficients du PID, la vitesse de chaque moteur $G(z)$ étant désormais connue, on procède à la synthèse du régulateur PID avec la période d'échantillonnage de 20 ms choisie. En utilisant la méthode de Ziegler Nichols, on choisit les valeurs des paramètres visibles dans le tableau IV.2.

Tableau IV.2 Paramètres PID choisis

VKp	0.013
VKi	0.005
VKD	0

Lors de l'implémentation sur la base robotique, les paramètres du régulateur ont été ajustés empiriquement : Un K_p plutôt élevé a été choisi. En effet, lorsque les moteurs ne commandent que les roues, le système est différent du cas où toute la masse du robot doit être déplacée (frottements avec le sol). Ceci rendant la commande trop molle si le K_p est trop faible. Une autre modification avec le choix d'un terme intégrateur plus faible, qui réagit vivement lorsqu'une erreur s'accumule, ce qui provoque de petites oscillations lorsque le robot est sur sa cible.

Dans la deuxième étape, il s'agit de régler le coefficient P distance et orientation. le choix De ces coefficients n'est pas totalement libre : étant soumis au seuillage de l'accélération, si le régulateur est réglé pour monter très vite en vitesse et décélérer tard avant d'accoster à la position finale il subira l'action du seuillage d'accélération. Le moteur ne ralentira pas assez vite et dépassera sa position finale, engendrant petit à petit des oscillations divergentes. Il faut donc choisir les coefficients suffisamment faibles pour ralentir assez tôt, ce qui est d'autant plus contraignant que l'accélération maximale choisie est faible.

L'action intégrale et dérivée n'étant pas utilisé ($K_i = K_d = 0$), il faudra ajuster d'abord le coefficient K_{DP} en mettant $K_{\theta P} = 0$ et effectuer une batterie de teste pour trouver une valeur qui garantira la plus petite erreur statique possible sans pour autant être gêné par la limitation de l'accélération maximale lors des phases de décélération. De la même manière, on ajuste le coefficient $K_{\theta P}$ du terme proportionnel du PID orientation.

Finalement après avoir déterminé et implémenter toutes les valeurs des coefficients l'asservissement sera opérationnel.

IV.3.4 Odométrie

En se basant sur le model du robot unicycle du chapitre (I), on peut calculer la position du robot au moyen d'une formule basée sur la vitesse angulaire et la vitesse d'avance du robot. Cette méthode a été choisie pour les raisons suivantes :

- Vitesses de rotation des roues mesurables, et par conséquent vitesses angulaire et d'avance faciles à calculer ;
- Bonne précision, si la position est mise à jour à une fréquence élevée ;
- Calculs réalisables dans le temps à disposition ;
- Facile à implémenter.

Les vitesses angulaires et d'avance ne sont pas directement mesurables. C'est la vitesse de chaque roue qui est mesurée à chaque instant d'échantillonnage. On en déduit facilement les vitesses angulaires et d'avance.

La formule suivante permet d'obtenir la vitesse de rotation du robot :

$$\omega = \frac{v_{droite} - v_{gauche}}{L} \quad (IV.20)$$

Où L est la distance entre les deux roues du robot. Une vitesse positive correspond à une rotation dans le sens trigonométrique positif.

Nous estimons ensuite la vitesse d'avance du robot en faisant la moyenne entre les deux vitesses des roues :

$$v = \frac{v_{droite} + v_{gauche}}{2} \quad (IV.21)$$

Les coordonnées du robot sont prises au milieu de l'essieu des roues, l'orientation θ représentant l'angle orienté formé par l'axe longitudinal du robot par rapport à l'axe des Y .

L'odométrie permet de calculer la distance parcourue sur la trajectoire, donc la valeur de la vitesse est la dérivée de la distance. L'odométrie permettant par ailleurs de connaître l'orientation du robot θ , l'orientation du vecteur vitesse, est par conséquent connu (représenté par exemple par le vecteur en trait gras sur la figure IV.12). Il suffit alors d'intégrer le vecteur vitesse à chaque période d'échantillonnage pour connaître la position du robot.

Vu que Nous sommes en numérique. La dérivation s'obtient par une simple soustraction et l'intégration par une simple addition.

Représentons en pointillé la trajectoire du robot sur le terrain (figure IV.12). Soit T_e la période d'échantillonnage. Plaçons nous à l'instant présent $n.T_e$ (point M sur la figure). Entre l'instant présent $n.T_e$ et l'instant suivant $(n+1).T_e$, le robot s'est déplacé de la distance d sur sa trajectoire en suivant son orientation θ_n . [26]

Les calculs permettant de mettre à jour la position sont présentés ci-après. La figure IV.12 permet de comprendre intuitivement la méthode utilisée.

$$\begin{pmatrix} x_{new} \\ y_{new} \\ \theta_{new} \end{pmatrix} = \begin{pmatrix} x_{old} \\ y_{old} \\ \theta_{old} \end{pmatrix} + \begin{pmatrix} \cos\theta \cdot d \\ \sin\theta \cdot d \\ \omega \cdot T_e \end{pmatrix} \quad (IV.22)$$

Avec d et θ estimés par :

$$d = v \cdot T_e \quad (\text{IV.23})$$

$$\theta = \omega T_e \quad (\text{IV.24})$$

Nous obtenons finalement pour l'estimation de la nouvelle position absolue :

$$\begin{pmatrix} x_{new} \\ y_{new} \\ \theta_{new} \end{pmatrix} = \begin{pmatrix} x_{old} \\ y_{old} \\ \theta_{old} \end{pmatrix} + \begin{pmatrix} v \cdot T_e \cos \theta \\ v \cdot T_e \sin \theta \\ \omega \cdot T_e \end{pmatrix} \quad (\text{IV.25})$$

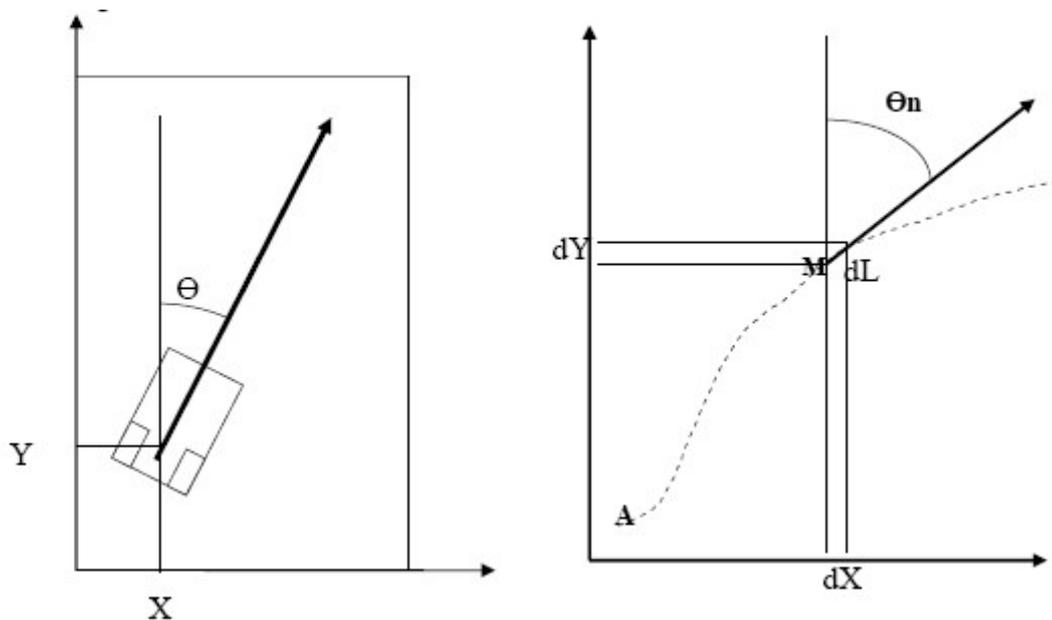


Figure IV.11 Estimation de la position absolue du robot.

L'odométrie par codeurs incrémentaux procédant par intégration est soumise à une dérive, c'est-à-dire à une incertitude sur le positionnement qui croît avec le temps. La valeur de cette dérive dépend de différents facteurs qui vont influencer sur la précision du résultat :

- La qualité mécanique de la réalisation du système de roues codeuses (Absence de jeu, alignement parfait avec les roues motrices, parallélisme parfait des 2 roues codeuse avec l'axe longitudinal du robot).
- La résolution des codeurs (Nombre de pas par tours). En pratique une résolution d'au moins 500 points semble nécessaire.
- Le choix du diamètre des roues codeuses (Trop petites, elles vont subir les irrégularités du terrain. Trop grandes elles font perdre en résolution).
- La qualité du module de comptage décomptage (Filtrage numérique des signaux, prise en compte des fronts montants et descendants des 2 canaux au front près).
- Le choix de la fréquence d'échantillonnage. Une fréquence trop grande va conduire à des erreurs de quantifications excessives. Une fréquence trop faible va allonger les arcs de trajectoire donc augmenter l'erreur dans l'approximation.

Si toutes ces conditions ne sont pas réunies, des recalages en cours de déplacement sont à prévoir avec une balise de localisation par exemple ou une méthode plus simple à mettre en œuvre comme les recalages sur les bordures de l'aire du déplacement du robot.

IV.3.5 Implémentation sur le microcontrôleur

Le microcontrôleur effectuera en continu un cycle de 20 ms durant lequel les opérations suivantes seront effectuées :

- Les vitesses des roues gauches et droites seront calculées à partir de leur position (la position des roues étant mesuré avec les encodeurs par un autre microcontrôleur et transmit via le bus I2c) ;
- la vitesse angulaire et la vitesse d'avance du robot seront calculées ;
- la position absolue du robot sera mise-à-jour ;
- en fonction de la dernière consigne donnée et des vitesses obtenues aux points précédents, la nouvelle vitesse angulaire et la nouvelle vitesse d'avance seront calculées (deux régulateur PID sur la distance et l'orientation du robot) ;
- les nouvelles vitesses pour chaque roue seront calculées ;
- les nouvelles vitesses pour chaque roue seront fournies comme consigne à deux régulateurs PID assurant la commande des moteurs.

Le reste du temps sera réservé à la communication par l'UART (transmission de la position actuelle, vitesses, messages, etc.) Le timer 1 du microcontrôleur permettra de contrôler la durée du cycle.

Le programme devra aussi être capable d'émettre des requêtes et recevoir des données via le bus I2C, ou encore arrêter les moteurs en cas de situation dangereuse (blocage par exemple).

IV.3.5.1 Contraintes et solution

Pour la réalisation de ce programme, nous allons devoir faire face à un certain nombre de contraintes.

La boucle de régulation numérique doit être exécutée à intervalles réguliers avec une fréquence importante. Le temps d'exécution de celle-ci devra donc être le plus court possible.

Afin d'effectuer les calculs numériques des boucles de régulation le plus rapidement possible, plusieurs astuces ont été utilisées :

- Utiliser un calcul en virgule seulement si c'est absolument nécessaire, tous les autres calculs étant effectués en nombres entiers.
- éviter le plus possible les divisions très coûteuses en temps, et les remplacer par des multiplications lorsque c'est possible. On remplacera par exemple une division par 5 par une multiplication par 0.2.
- Le microcontrôleur ayant une architecture de 8 bits, il faut veiller à choisir correctement les types des variables. On préférera utiliser par exemple le type particulier short de 16 bits plutôt que le type long de 32 bits, si 16 bits sont suffisants pour stocker notre variable.
- Essayer d'utiliser au maximum des opérations binaires (décalage, ...) plutôt que des opérations arithmétiques lorsque c'est possible.
- Il convient de faire attention à la modification des variables pouvant être lues dans d'autres fonctions.
- Un temps optimal entre chaque cycle devra être laissé pour pouvoir émettre des requêtes et recevoir des données (position de chaque roue codeuse) via le bus I2C.

IV.4 Résultats et évaluation des performances

IV.4.1 Résultats

Nous allons ici monter deux manipulations simple qui nous ont aidé à évaluer une partie des performances de notre système dans ces expériences on prendra:

$$\omega_{\max} = 4.85 \text{ rad/s} \quad (\text{IV.25})$$

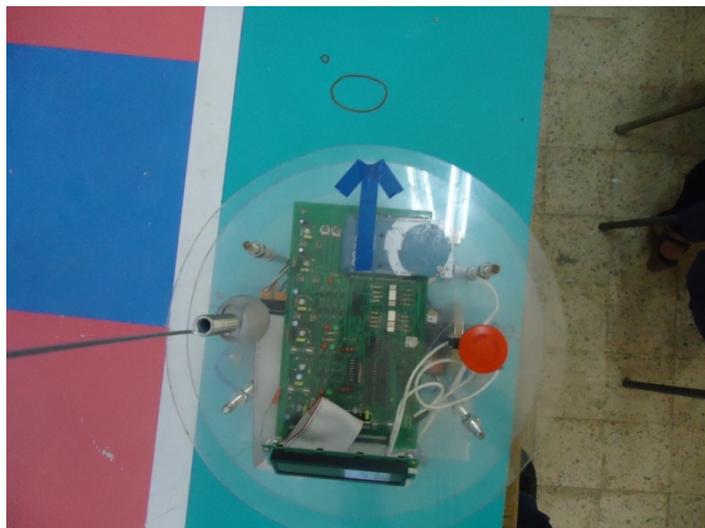
$$A_{\theta \max} = 3.63 \text{ rad/s}^2 \quad (\text{IV.26})$$

$$v_{\max} = 0.4 \text{ m/s.} \quad (\text{IV.27})$$

$$A_{d \max} = 0.6 \text{ m/s} \quad (\text{IV.28})$$

IV.4.1.1 Manipulation rotation symétrique

Cette manipulation consiste a déplacer le robot d'une configuration de départ A ($v = 0$, $\omega = 0$, $x = 0$, $y = 0$, $\theta = 0$) vers configuration B ($v = 0$, $\omega = 0$, $x = 0$, $y = 0$, $\theta = 180^\circ$).



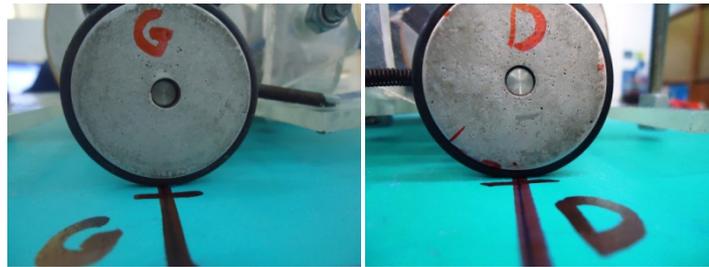


Figure IV.12 position du robot à la configuration initiale A

La figure IV.12 montre le robot sur sa configuration initiale A, la roue droite et le côté droit de la table sont repérés par le signe D, la roue gauche et le côté gauche de la table sont repérés par le signe G.

En donnant une consigne de rotation de 180° et une consigne de distance de 0 le robot se déplace en effectuant une rotation symétrique.

La figure IV.13 montre le robot sur sa configuration finale B.

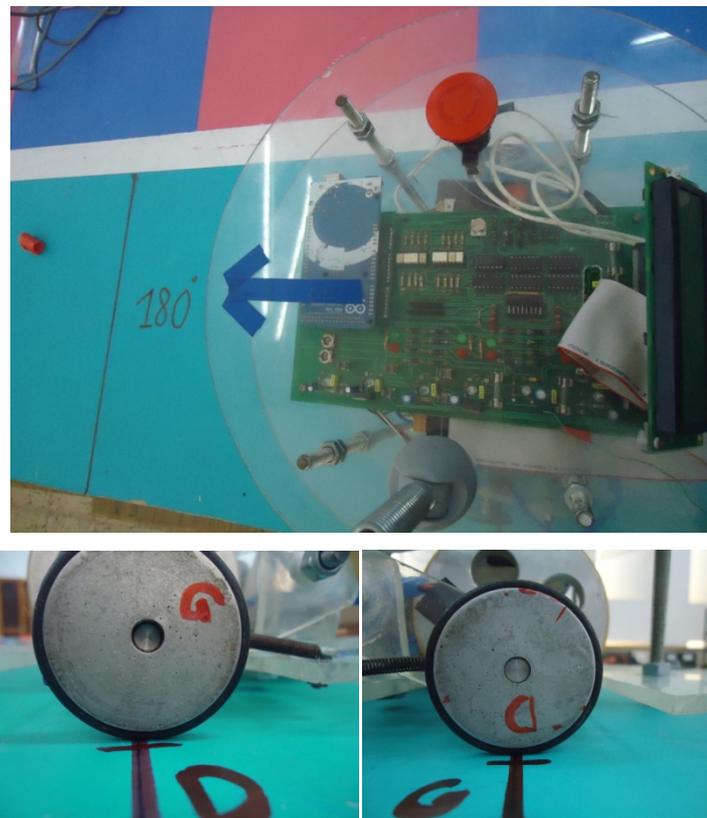


Figure IV.13 la position du robot à la configuration finale B

L'évolution de la vitesse d'avance et angulaire ainsi que la distance et l'orientation du robot sont présentés sur les figures suivantes.

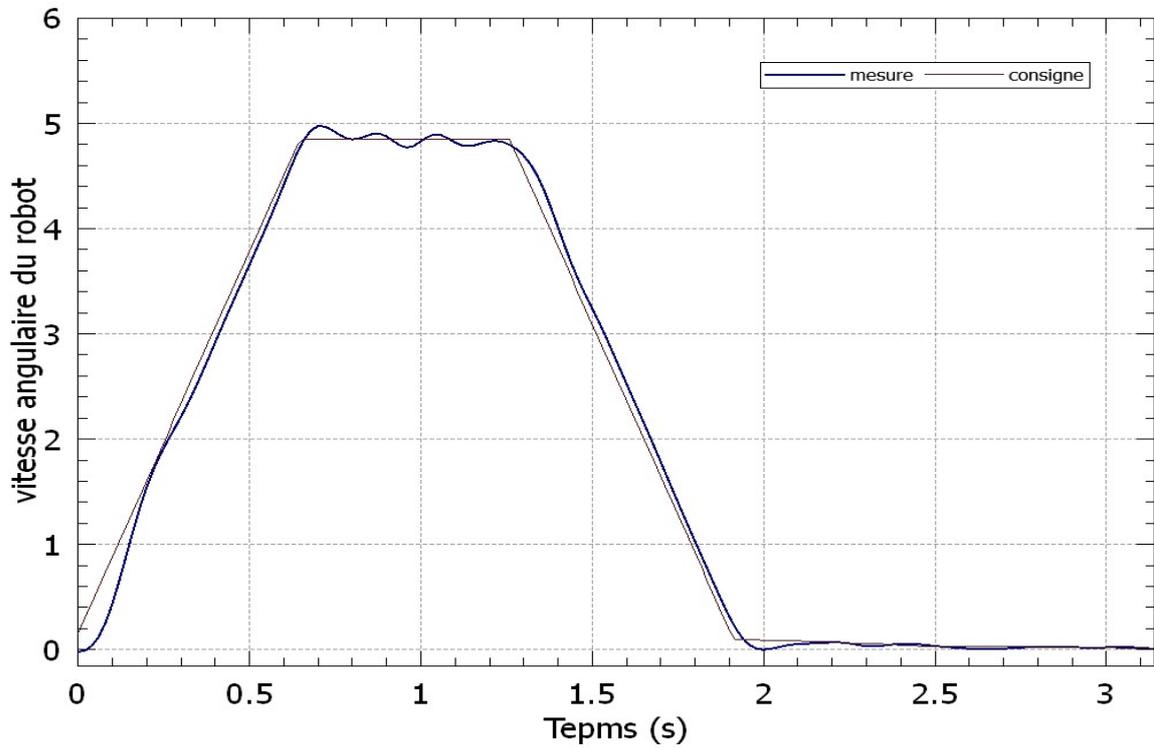


Figure IV.14 évolution de la vitesse angulaire du robot

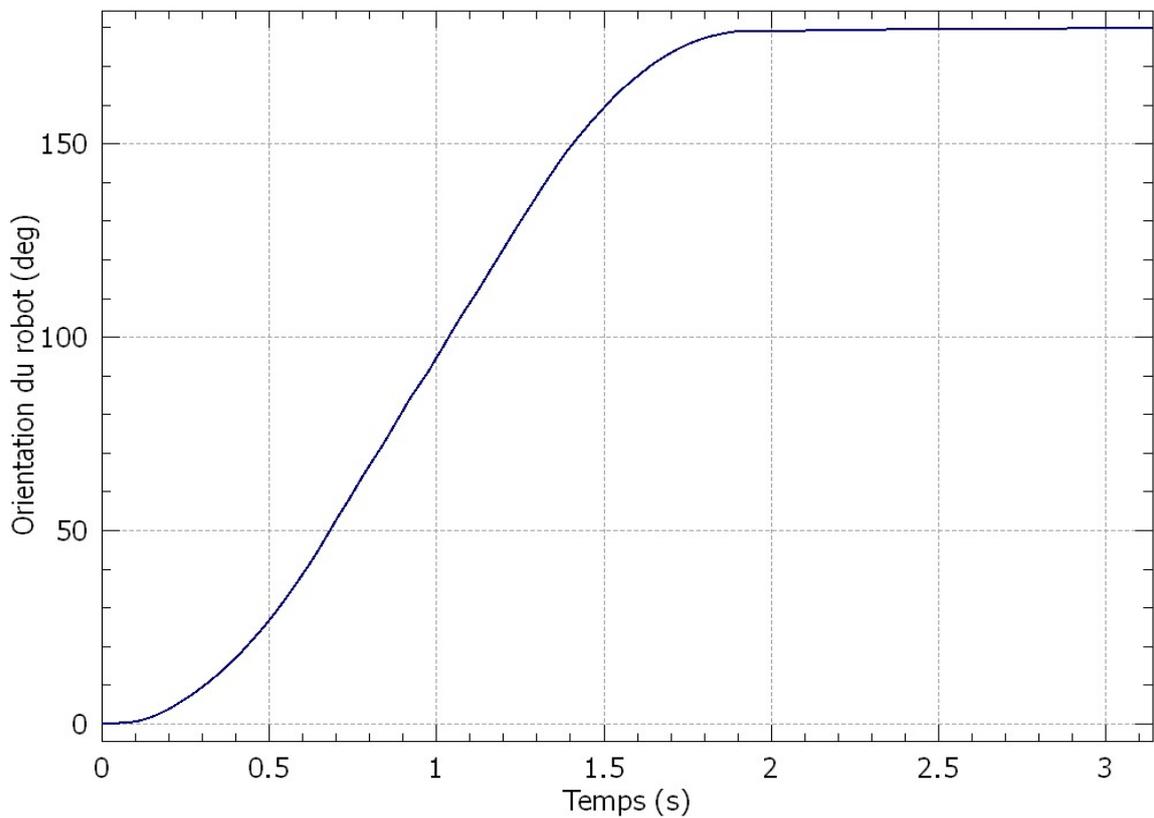
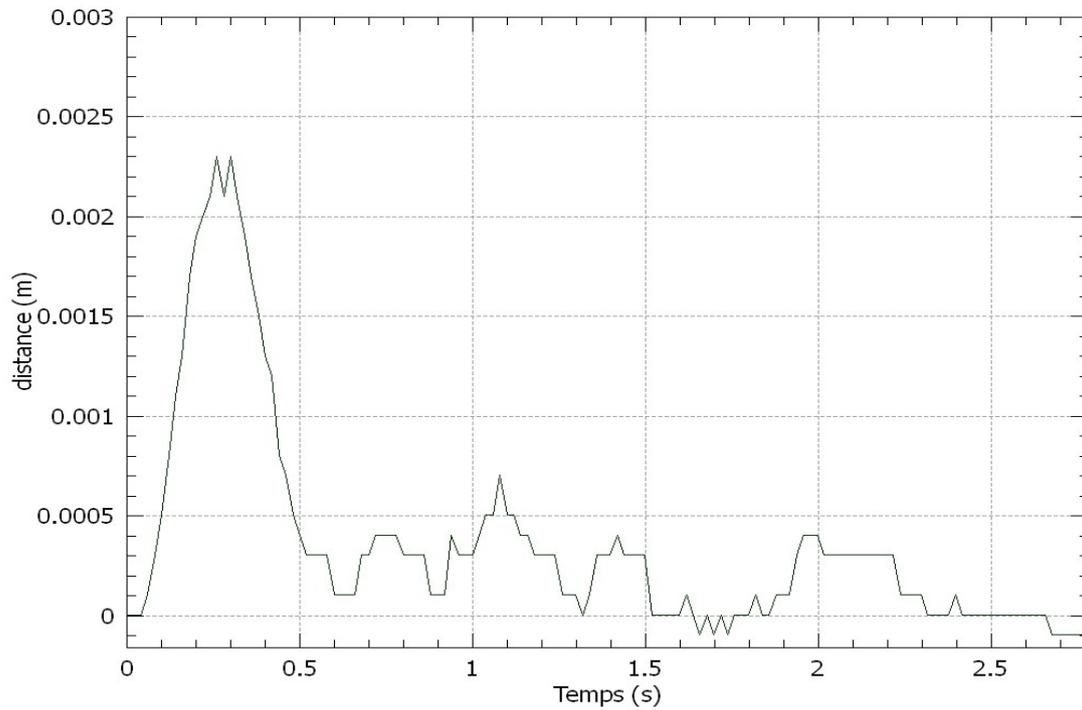
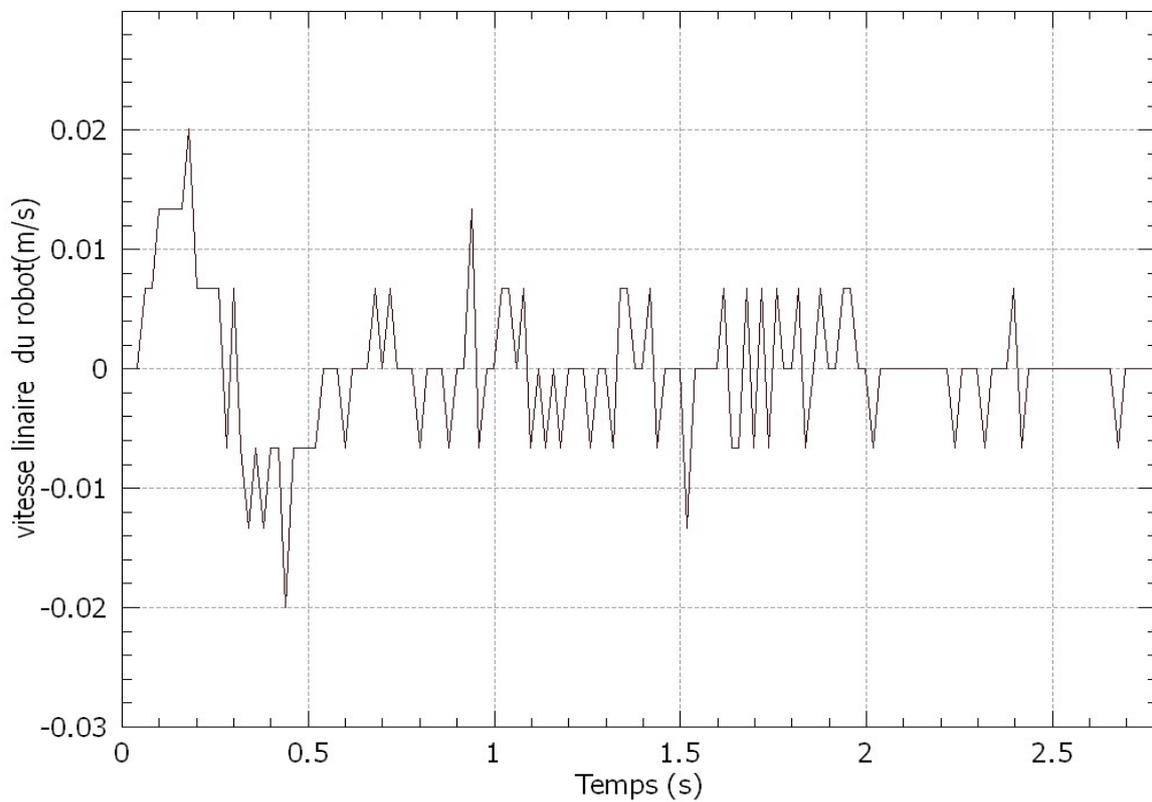


Figure IV.15 évolution de l'orientation du robot**Figure IV.16** évolution de la distance du robot**Figure IV.17** évolution de la vitesse linéaire du robot

On remarque un profil trapézoïdal de la vitesse angulaire, le robot accélère puis reste sur sa vitesse maximale avant de décélérer pour accoster à l'orientation final avec une vitesse nulle. Les graphes montrent aussi le bon suivi de la consigne même si une petite erreur de trainage existe et elle est visible surtout au début en effet cette erreur est due au frottement du robot sur le sol et l'orientation du robot est asservie avec une très bonne précision, on note une erreur statique de 0.5° soit 0.3%. La vitesse longitudinale et l'avance du robot sont pratiquement nulles.

IV.4.1.2 Manipulation ligne droite

cette manipulation consiste a déplacé le robot d'une configuration de départ A ($v = 0, \omega = 0, x = 0, y = 0, \theta = 0$) vers configuration B ($v = 0, \omega = 0, x = 0, y = 1000mm, \theta = 0$).

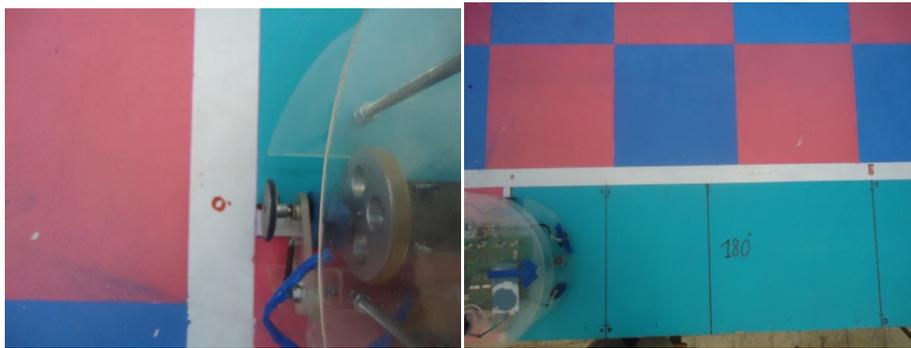


Figure IV.18 la position du robot à la configuration initiale A

La figure IV.18 montre le robot sur sa configuration initiale A, le signe 0 montre la ligne de départ et le signe 1000 montre la ligne d'arrive placée a un mètre et en parallèle avec la dernière ligne.

En donnant une consigne de rotation de 0° et une consigne de distance de 1m le robot se déplace en effectuant une ligne droite.

La figure IV.19 montre le robot sur sa configuration finale B.



Figure IV.19 la position du robot à la configuration finale B

L'évolution de la vitesse longitudinale et angulaire ainsi que la distance et l'orientation du robot sont présentés sur les figures suivantes.

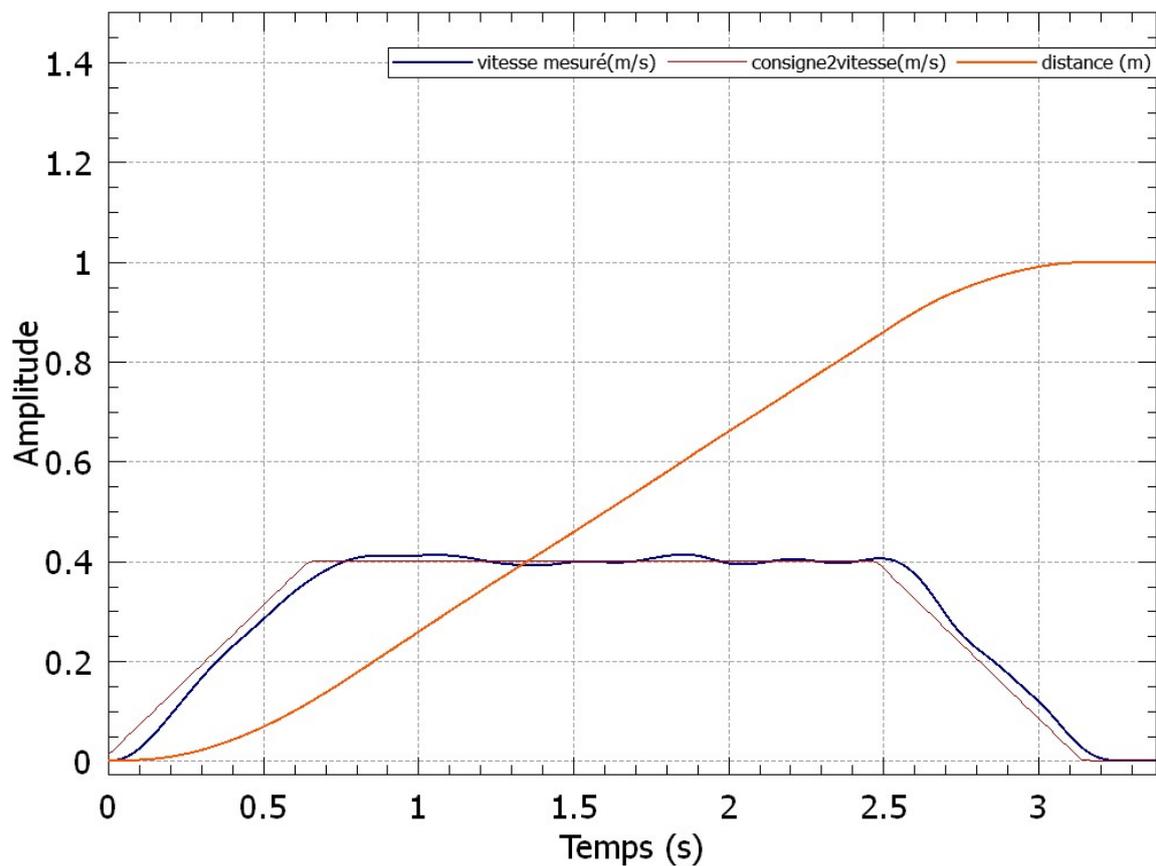


Figure IV.20 évolution de la vitesse d'avance et la distance parcourue par le robot

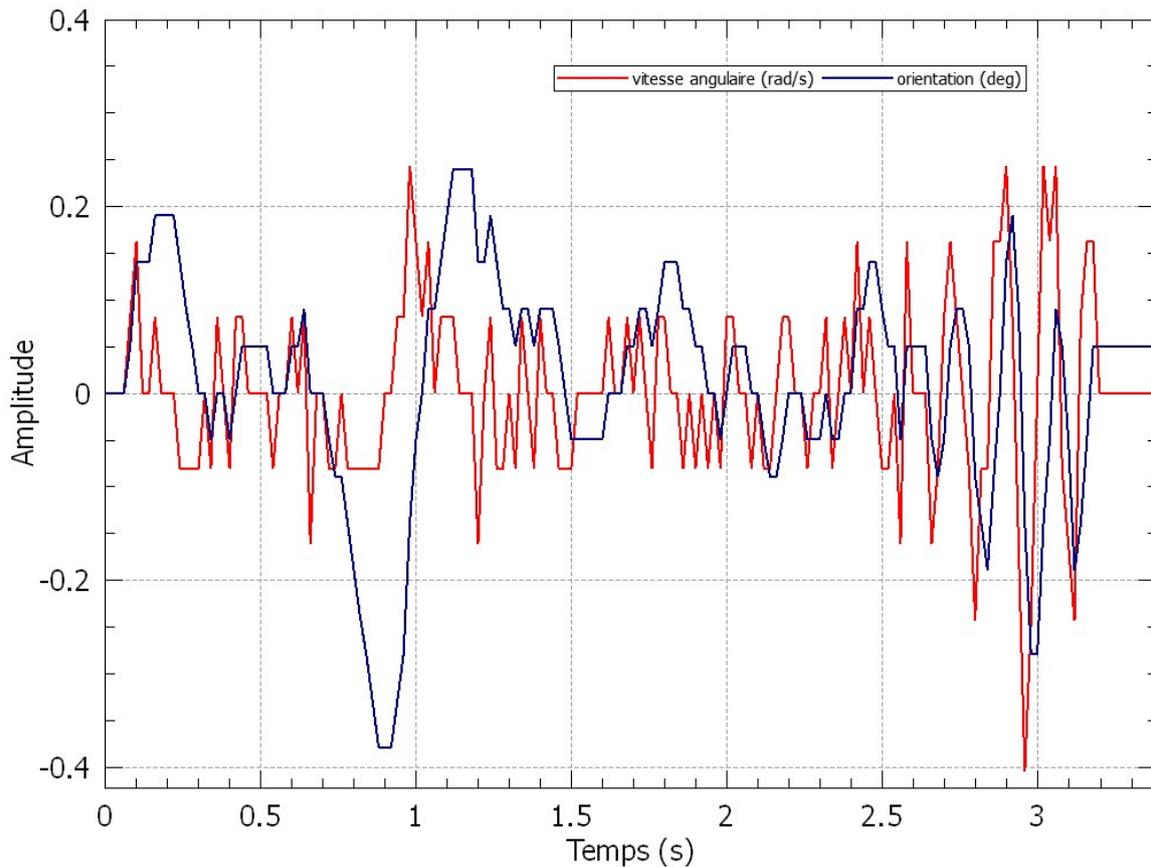


Figure IV.21 évolution de la vitesse angulaire et de l'orientation du robot

On remarque un profil trapézoïdale de la vitesse longitudinale, le robot accélère puis reste sur sa vitesse maximale avant de décélérer pour accoster à la distance finale avec une vitesse nulle. Les graphes montrent aussi le bon suivi de la consigne de vitesse même si une erreur de trainage et la distance du robot est asservie avec une très bonne précision, on note une erreur statique de 1mm soit 0.1%. La vitesse angulaire et l'orientation du robot sont presque nulles.

On observe que le robot dévie à l'arrivée de la ligne droite d'environ 1 à 2mm de sa cible mais cela reste acceptable.

L'odométrie nous permet de localiser le robot sur la table et le suivi de la ligne droite est présenté sur la figure IV.22 les coordonnées initiale sont $x = 250\text{mm}$, $y = 250\text{mm}$, $\theta = 0^\circ$.

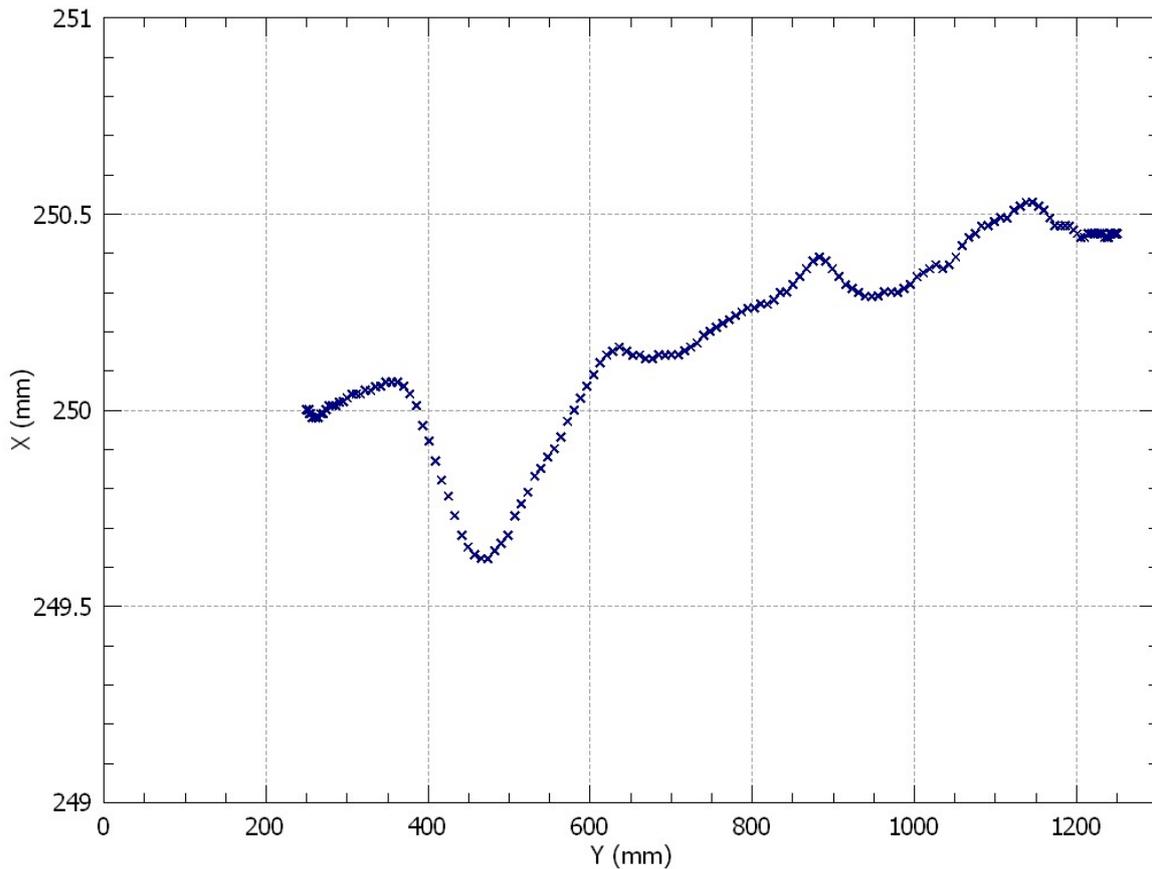


Figure IV.22 évolution de la localisation par odométrie

En comparant le résultat de la localisation par odométrie avec la position réelle du robot on s'aperçoit que l'erreur n'est pas vraiment visible.

Remarque : Pour les deux manipulations les graphes d'évolution d'autres variables comme les tensions de commande appliquées au moteur sont présentées en annexe 1.

IV.4.2 Évaluation des performances

IV.4.2.1 La Partie mécanique

Le choix d'un robot de type unicycle s'est avéré judicieux. En effet, le terrain étant limité et rempli d'obstacles pour ce concours, le robot peut s'y déplacer avec une grande mobilité et se sortir de situations contraignantes. Par exemple, si il est plaqué contre la bordure de la table de jeux notamment grâce à sa forme cylindrique.

Malgré tous nos efforts, on ne peut pas dire que la réalisation et le montage des pièces mécaniques sont de très grande qualité à défaut de moyens nécessaires et de compétences

requis pour effectuer cette tâche. Néanmoins le résultat est satisfaisant et on ne remarque pas un très grand impact sur la qualité du contrôle. Pour de meilleures performances, cette partie devra être améliorée et confiée plutôt à des spécialistes du domaine.

IV.2.2 La partie électronique

Cette partie nous permet d'exécuter toutes les tâches prédéfinies. Grâce à la carte de puissance, nous pouvons piloter efficacement les moteurs. Le pont H en circuit intégré avait tendance à dégager un peu de chaleur lors des utilisations de longues durées. Un refroidisseur a été placé pour une meilleure dissipation de la chaleur. La détection d'obstacles et la mesure du courant consommé s'est avérée très utile car en détectant une situation dangereuse, on peut décider de la manœuvre à effectuer (mettre les moteurs hors tension par exemple).

Le microcontrôleur est un élément central de la partie logique. Son choix a été soumis à plusieurs contraintes. Le microcontrôleur utilisé (atmega2560) a répondu au cahier des charges imposé. Grâce à des astuces de programmation évoquées dans la section précédente, la boucle de commande est effectuée en un cycle relativement court.

En plus de la performance que propose ce microcontrôleur, notre choix a été guidé par l'utilisation de la plateforme Arduino particulièrement adaptée à ce type d'application. Elle offre une grande modularité et souplesse d'utilisation tout en ayant un aspect open source et beaucoup de sources qui facilitent le partage de connaissances. Pour plus de performances, on peut utiliser des microcontrôleurs plus puissants ou d'autres types de systèmes comme les FPGA ou les DSP. Mais les prix et la difficulté de mise en œuvre seront revus à la hausse.

IV.4.2.3 Partie contrôle

On peut dire que l'implémentation de la méthode de contrôle choisie est réussie. En enchaînant des mouvements de rotation symétrique et de ligne droite, le robot passe d'une configuration A vers une configuration B avec une bonne précision. Cependant, nous avons remarqué que si une perturbation d'amplitude relativement grande est appliquée sur l'orientation du robot, cette dernière était corrigée. Toutefois, le robot dévie de sa trajectoire n'atteignant pas sa cible. Notons aussi que la méthode d'asservissement, qui concerne les deux régulateurs PID montés en cascade, donne de bonnes performances.

L'avance et l'orientation du robot sont régulées avec précision et sans dépassement ou oscillations, grâce à un profil de vitesse trapézoïdal on note l'évolution parabolique de la distance au cours des deux phases accélération et freinage qui permet des mouvements en douceur.

La régulation de la vitesse d'orientation et d'avance est, quant à elle, moins précise étant soumise à plus de contraintes (perturbation de la mesure). Cela dit, ses performances sont assez acceptables, d'autant plus que l'orientation et l'avance du robot sont les deux variables sur lesquelles la commande doit être la plus précise que possible. Plusieurs autres actionneurs et systèmes de régulations existent pouvant donner des résultats différents, on peut citer le Correcteur RST, Logique floue, Réseaux de neurones artificiels etc...

IV.4.2.4 Localisation par odométrie :

Cette méthode donne une assez bonne estimation de la position absolue du robot (coordonnées, orientation) néanmoins elle manque de précision cela est dû à plusieurs facteurs déjà évoqués au point IV.3.4 de plus elle nécessite la connaissance exacte du diamètre des roues codeuse et de l'entraxe et cette tâche s'avère un peu délicate néanmoins on peut calibrer le robot en comparant la position calculée avec la position réelle, il faut aussi faire attention à le bien positionner par rapport à son repère initial. Même si le terrain permet l'odométrie cette méthode reste imprécise il sera plus judicieux d'utiliser un autre type de système de localisation en complément comme des balises par exemple. Ce point fera sans doute l'objet d'une étude approfondie lors de notre prochaine réalisation pour la participation au concours Eurobot.

IV.5 Conclusion

Ce chapitre représente la partie principale de notre travail après avoir effectué le choix de la solution à appliquer au chapitre I nous nous sommes lancés à sa mise en œuvre. Au début nous avons fixé les objectifs à atteindre puis nous avons fait la conception et la réalisation des différentes parties de notre module à savoir la mécanique, l'électronique et la partie contrôle, enfin la localisation par odométrie, après l'implémentation et le réglage des différents paramètres nous avons conclu sur les performances de notre système grâce aux résultats obtenus et affichés sous forme de graphes représentatifs. Les améliorations possibles ont été présentées pour chaque partie. Au final nous pouvons dire que les

résultats obtenus sont concluants ; le module réalisé permet de se localiser et de générer des déplacements, notamment de passer d'une configuration A vers une configuration B dans un périmètre fixé avec une manière rapide précise et fiable.

Liste des figures

Chapitre I

Généralités sur les robots

Figure I.1 convoyeur automatisés AGV	5
Figure I.2 Table Eurobot	7
Figure I.3 déplacement en ligne droite entre 3 positions	8
Figure I.4 Robot de type unicycle	9
Figure I.5 Robot de type tricycle	9
Figure I.6 Robot de type voiture	10
Figure I.7 Robot de type omnidirectionnel	11

Chapitre II

Application de la théorie du PID sur le modèle du moteur à courant continu

Figure II.1 Schéma bloc d'un système en boucle ouverte	19
Figure II.2 Schéma bloc d'un système en boucle fermée avec un retour unitaire	20
Figure II.3 Système possédant une erreur statique.	21
Figure II.4 Système a erreur de trainage.	21
Figure II.5 Temps de réponse a 5%.	22
Figure II.6: Schéma bloc d'un asservissement avec une perturbation (Retour unitaire)	23
Figure II.7 Schéma bloc d'un système avec correcteur (Retour unitaire)	24
Figure II.8 Schéma d'un moteur à courant continu	25
Figure II.9 Schéma équivalent d'un moteur à courant continu	26
Figure II.10 Schéma bloc du Modèle de moteur électrique en vitesse	29
Figure II.11 Modélisation sous Matlab de la réponse de vitesse	30
Figure II.12 Schéma bloc du Modèle de moteur électrique en position	30
Figure II.13 Modélisation sous Matlab de la réponse de position	31
Figure II.14 correcteur proportionnel dans un sservissement en vitesse	33
Figure II.15 correcteur PI dans asservissement en vitesse	34
Figure II.16 Schéma bloc du correcteur PID en régulation de vitesse	35
Figure II.17 Schéma bloc du correcteur PID en régulation de position	35

Figure II.18 réponse indicielle d'un processus du deuxième ordre	37
Figure II.19 Control and estimation tools manager	40
Figure II.20 Réponse du modèle moteur a une vitesse de 500 rad/s	41
Figure II.21 Diagramme de Bode modèle du moteur	42

Chapitre III

Mise en œuvre de la commande PID d'un moteur a courant continu

Figure III.1 Schéma de la mise en œuvre de l'asservissement	45
Figure III.2 Moteur à courant continue équipé d'un encodeur et d'un réducteur mécanique ..	46
Figure III.3 Signaux en quadrature	47
Figure III.4 Détection du sens de rotation à l'aide d'une bascule D	47
Figure III.5 Schéma de l'unité de capture de l'atmega2560	48
Figure III.6 Schéma de mise en œuvre de la mesure de vitesse	49
Figure III.7 Schéma de mise en ouvre de la mesure de position	50
Figure III.8 Schéma du montage du pont H	51
Figure III.9 Diagramme temporel Sign – Magnitude	53
Figure III.10 Diagramme temporel Locked Anti-phase	54
Figure III.11 Principe de la PWM	56
Figure III.12 edge-aligned PWM	57
Figure III.13 center-aligned PWM	58
Figure III.14 Edge-aligned PWM Mode, Timing Diagram	59
Figure III.15 diagramme de Bode du model physique du moteur réducteur.	62
Figure III.16 évolution de la vitesse du rotor pour une correction PID	65
Figure III.17 évolution de la vitesse du rotor pour une correction PI	65
Figure III.18 résultat du filtrage de la vitesse pour une correction PI	65
Figure III.19 asservissement de vitesse avec profil trapézoïdale	66

Chapitre IV

Mise en œuvre du système de locomotion

Figure IV.1 Le module développé servira d'intermédiaire entre l'utilisateur et les moteurs	69
Figure IV.2 Vue de la disposition mécanique du robot utilisé	70
Figure IV.3 Mise en place d'une roue codeuse	71
Figure IV.4 schéma sous porteuse de la care de puissance	73
Figure IV.5 capteur infrarouge Sharp GP2Y0A02YK	78
Figure. IV.6 On cherche à calculer la force F_f	82
Figure. IV.7 Régulateurs haut et bas niveau en cascade.	85
Figure IV.8 schéma bloc de la commande.	86
Figure IV.9 Les quatre Différentes combinaisons de déplacement possible.	87
Figure IV.10 diagramme de Bode du modèle physique du robot	88
Figure IV.11 Estimation de la position absolue du robot.	91
Figure IV.12 position du robot à la configuration initiale A	95
Figure IV.13 la position du robot à la configuration finale B	96
Figure IV.14 évolution de la vitesse angulaire du robot	96
Figure IV.15 évolution de l'orientation du robot	97
Figure IV.16 évolution de la distance du robot	97
Figure IV.17 évolution de la vitesse linéaire du robot	98
Figure IV.18 la position du robot à la configuration initiale A	99
Figure IV.19 la position du robot à la configuration finale B	99
Figure IV.20 évolution de la vitesse linéaire et la distance du robot	100
Figure IV.21 évolution de la vitesse angulaire et de l'orientation du robot	101
Figure IV.22 évolution de la vitesse angulaire et de l'orientation du robot	102

Liste des tableaux

Chapitre I Généralités sur les robots

Tableau I.1 comparaison des différents types de robot a roues	11
---	----

Chapitre II application de la théorie du PID sur le modèle du moteur a courant continu

Tableau II.1 Valeurs numériques d'un moteur à courant continu	29
Tableau II.2 tableau récapitulatif des paramètres PID	36
Tableau II.3 Paramètres PID obtenus à partir d'une réponse indicielle (ZN)	37
Tableau II.4 Réglage du contrôleur par la méthode d'oscillation de Ziegler-Nichols	39
Tableau II.5 les coefficients PID trouvés	40

Chapitre III Mise en œuvre de la commande PID d'un moteur a courant continu

Tableau III.1 coefficients des termes PID calculés à partir de la méthode ZN	64
--	----

Chapitre IV Mise en œuvre du système de locomotion

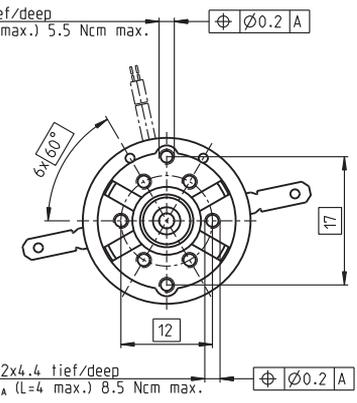
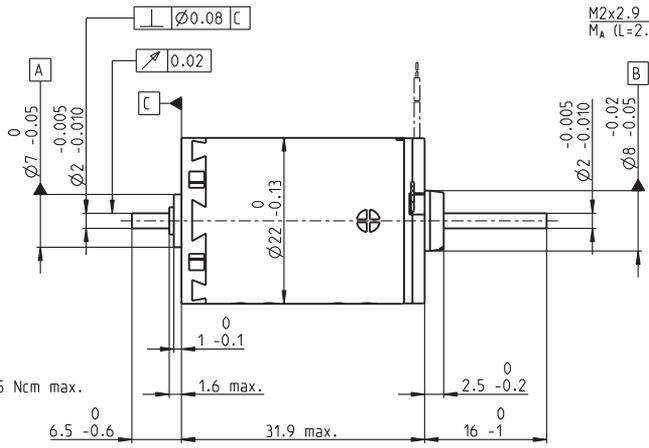
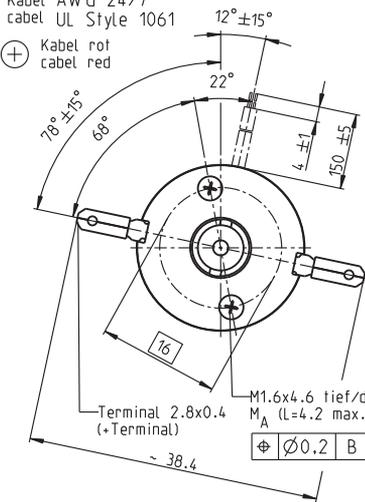
Tableau IV.1 Effet des entrées sur le pont H	72
Tableau IV.2 Paramètres PID choisis	88

A-max 22 Ø22 mm, Commutation Graphite, 6 Watt

maxon A-max

Kabel AWG 24/7
cabel UL Style 1061

⊕ Kabel rot
cabel red



M 1:1

- Programme Stock
- Programme Standard
- Programme Spécial (sur demande)

Numéros de commande

avec bornes	110156	110158	110159	110160	110161	110162	110163	110164	110165	110166	110167	110168
avec câbles	139848	353023	353024	231171	353025	353026	231174	353027	353028	353029	316659	353603

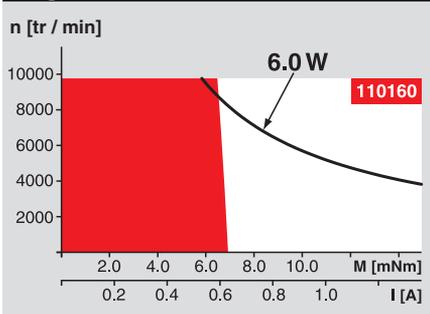
Caractéristiques moteur

Valeurs à la tension nominal																								
1 Tension nominale	V	6.0	9.0	9.0	12.0	12.0	15.0	18.0	24.0	24.0	36.0	48.0	48.0											
2 Vitesse à vide	tr / min	9250	9710	8530	10200	9200	10100	9800	10500	8500	9650	9130	8220											
3 Courant à vide	mA	83.2	57.9	49.7	45.9	40.5	36.0	29.0	23.7	18.4	14.2	10.0	8.85											
4 Vitesse nominale	tr / min	5550	6370	5240	6990	5960	6880	6630	7430	5340	6500	5920	5020											
5 Couple nominal (couple permanent max.)	mNm	5.82	6.52	6.76	6.77	6.82	6.87	6.94	6.97	7.07	7.00	6.91	7.02											
6 Courant nominal (courant permanent max.)	A	1.06	0.816	0.741	0.664	0.602	0.529	0.433	0.350	0.287	0.214	0.150	0.138											
7 Couple de démarrage	mNm	16.1	20.4	18.7	22.8	20.4	22.7	24.3	24.3	19.5	21.9	20.1	18.5											
8 Courant de démarrage	A	2.73	2.38	1.92	2.09	1.69	1.64	1.30	1.14	0.745	0.631	0.411	0.340											
9 Rendement max.	%	65	70	69	72	71	72	72	73	71	72	71	70											
Caractéristiques																								
10 Résistance aux bornes	Ω	2.20	3.78	4.69	5.74	7.12	9.15	13.8	21.0	32.2	57.1	117	141											
11 Inductivité	mH	0.106	0.222	0.288	0.362	0.445	0.584	0.890	1.37	2.10	3.68	7.29	8.95											
12 Constante de couple	mNm / A	5.90	8.55	9.73	10.9	12.1	13.9	17.1	21.2	26.2	34.8	48.9	54.3											
13 Constante de vitesse	tr / min / V	1620	1120	981	875	790	689	558	450	364	274	195	176											
14 Pente vitesse / couple	tr / min / mNm	604	494	473	461	465	455	451	445	447	450	466	458											
15 Constante de temps mécanique	ms	25.2	21.8	21.2	20.6	20.3	19.9	19.4	19.1	19.0	18.9	18.9	18.8											
16 Inertie du rotor	gcm ²	3.98	4.22	4.28	4.26	4.17	4.17	4.11	4.11	4.07	4.00	3.88	3.92											

Spécifications

- Données thermiques**
- 17 Résistance therm. carcasse/air ambiant 20 K / W
 - 18 Résistance therm. bobinage/carcasse 6.0 K / W
 - 19 Constante de temps therm. bobinage 10.1 s
 - 20 Constante de temps therm. du moteur 540 s
 - 21 Température ambiante -30 ... +85°C
 - 22 Température max. de bobinage +125°C
- Données mécaniques (paliers lisses)**
- 23 Nombre de tours limite 9800 tr / min
 - 24 Jeu axial 0.05 - 0.15 mm
 - 25 Jeu radial 0.012 mm
 - 26 Charge axiale max. (dynamique) 1 N
 - 27 Force de chassage axiale max. (statique) 80 N (statique, axe soutenu) 440 N
 - 28 Charge radiale max. à 5 mm de la face 2.8 N

Plages d'utilisation



Légende

- Plage de fonctionnement permanent**
Compte tenu des résistances thermiques (lignes 17 et 18) la température maximum du rotor peut être atteinte au valeur nominal de couple et vitesse et à la température ambiante de 25°C.
= Limite thermique.
- Fonctionnement intermittent**
La surcharge doit être de courte durée.
- Puissance conseillée**

- Données mécaniques (roulement à billes)**
- 23 Nombre de tours limite 9800 tr / min
 - 24 Jeu axial 0.05 - 0.15 mm
 - 25 Jeu radial 0.025 mm
 - 26 Charge axiale max. (dynamique) 3.3 N
 - 27 Force de chassage axiale max. (statique) 45 N (statique, axe soutenu) 440 N
 - 28 Charge radiale max. à 5 mm de la face 12.3 N
- Autres spécifications**
- 29 Nombre de paires de pôles 1
 - 30 Nombre de lames au collecteur 9
 - 31 Poids du moteur 54 g

Construction modulaire maxon

Réducteur planétaire
Ø22 mm
0.1 - 0.6 Nm
Page 219 / 220

Réducteur planétaire
Ø22 mm
0.5 - 2.0 Nm
Page 221 / 222

Réducteur à pignons droits
Ø24 mm
0.1 Nm
Page 226

Entraînement vis/écrou
Ø22 mm
Page 247 / 248

Codeur MR
32 Imp.,
2 / 3 canaux
Page 258

Codeur MR
128 / 256 / 512 Imp.,
2 / 3 canaux
Page 260

Codeur Enc
22 mm
100 Imp., 2 canaux
Page 265

Codeur MEnc
Ø13 mm
16 Imp., 2 canaux
Page 274

Electronique recommandée:
LSC 30/2 Page 282
EPOS2 Module 36/2 304
Informations 18

Aperçu à la page 16 - 21

Les caractéristiques moteur du tableau sont des valeurs nominales.
Explications des chiffres page 49.

Option
Roulements à billes au lieu des paliers lisse

CONCLUSION GENERALE

Ce projet présentait deux aspects très distincts. Premièrement, il s'agissait de développer une base robotique destinée à tester des algorithmes de commande. Il était donc principalement question d'électronique et de mécanique.

Il a fallu faire les choix les plus judicieux concernant les composants à utiliser, au risque de ne jamais pouvoir traiter la deuxième partie du projet.

Le second aspect du projet était plus tourné vers la théorie, des algorithmes ont été choisis, étudiés et implémentés, il s'agissait entre autre du régulateur PID pour la commande du déplacement et de l'odométrie pour la localisation du robot.

Finalement, l'implémentation a permis de lier théorie et pratique, pour atteindre un résultat tout à fait concluant. En effet, les objectifs ont été atteints et le travail effectué sera utilisé pour le concours Eurobot de l'année prochaine.

Les problèmes rencontrés ont cependant été nombreux. Les plus importants se sont manifestés au niveau du calcul de la vitesse et de la position, à tel point que c'est cet aspect du projet qui a demandé le plus d'investissement. L'implémentation sur le microcontrôleur a posé quelques problèmes au niveau vitesse d'exécution. Ces quelques soucis ont pu être résolus ou contournés.

En résumé, le projet a permis un contact avec beaucoup de domaines différents, tels que l'électronique, la mécanique, l'automatique et la programmation.

La confrontation entre la théorie et la pratique était un point intéressant et la diversité des tâches à accomplir de même la gestion du projet ont permis l'acquisition de beaucoup de connaissances.

Consigne=200 rad/s

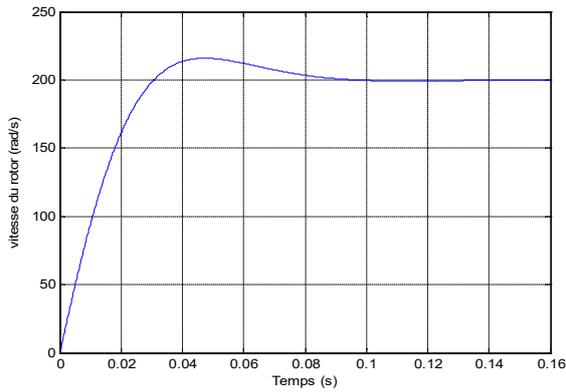


Figure 1 réponse du moteur en simulation

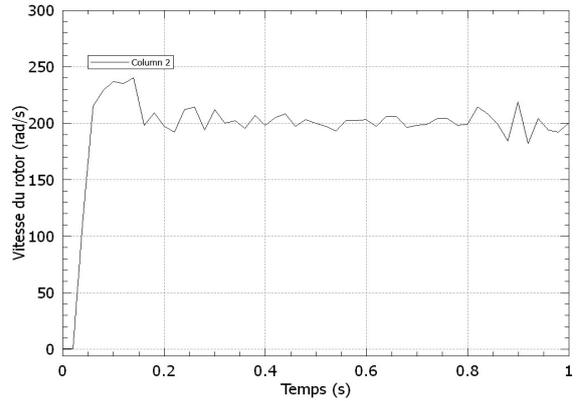


figure 2 réponse du moteur en pratique

Consigne=300 rad/s

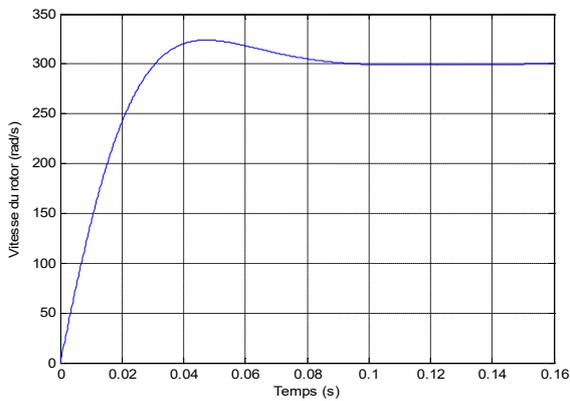


Figure 3 réponse du moteur en simulation

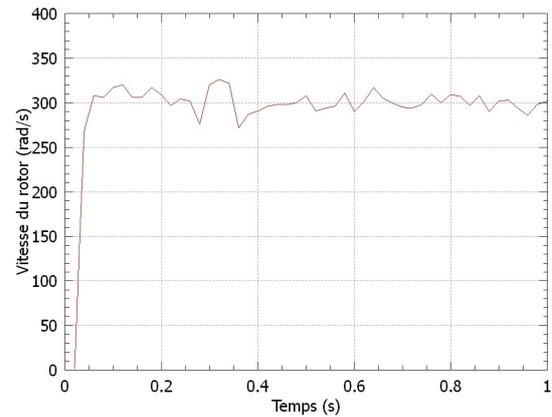


figure 4 réponse du moteur en pratique

Consigne=500 rad/s

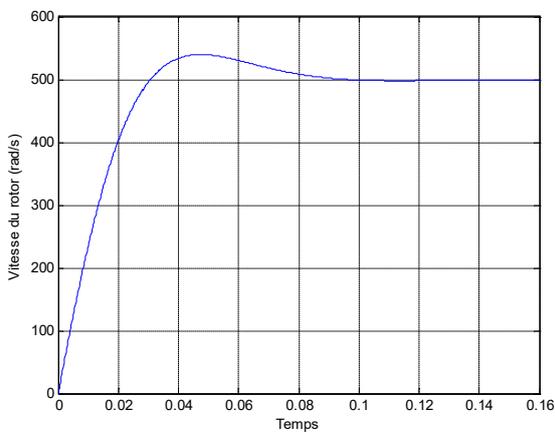


Figure 5 réponse du moteur en simulation

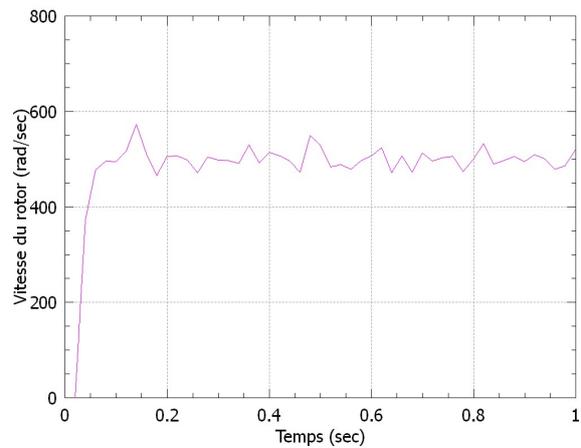


figure 6 réponse du moteur en pratique

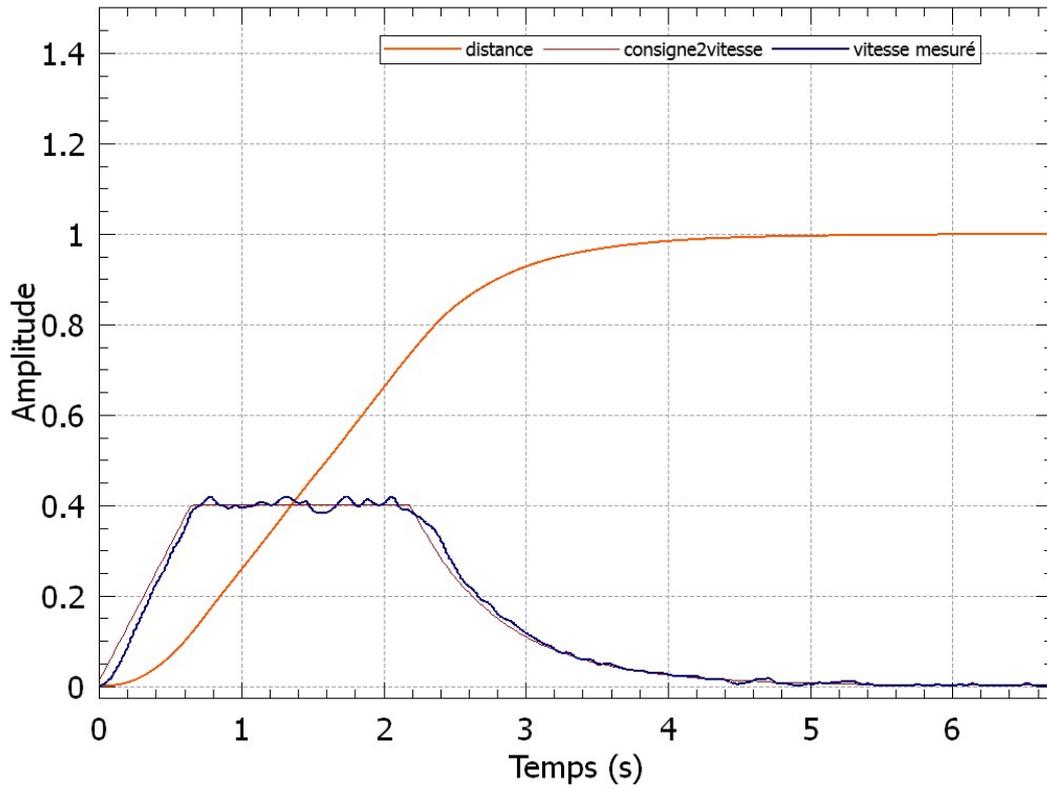


Figure 7 évaluations de la vitesse et de la position avec K_p faible

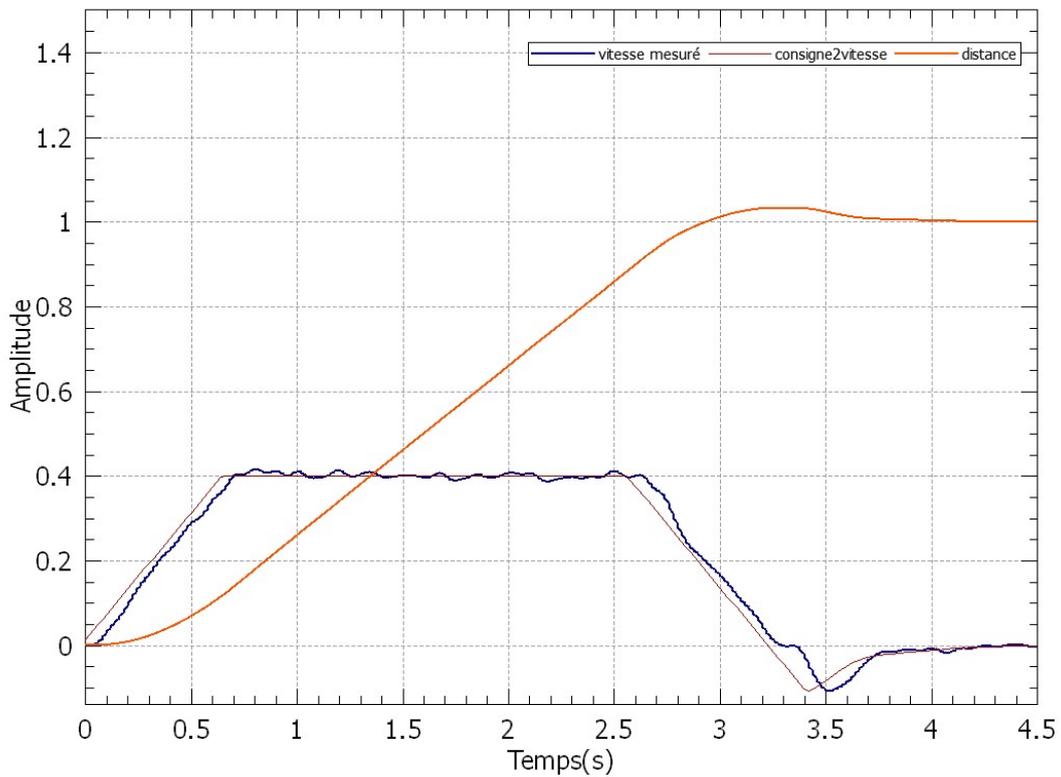


Figure 8 évaluations de la vitesse et de la position avec K_p élevé.

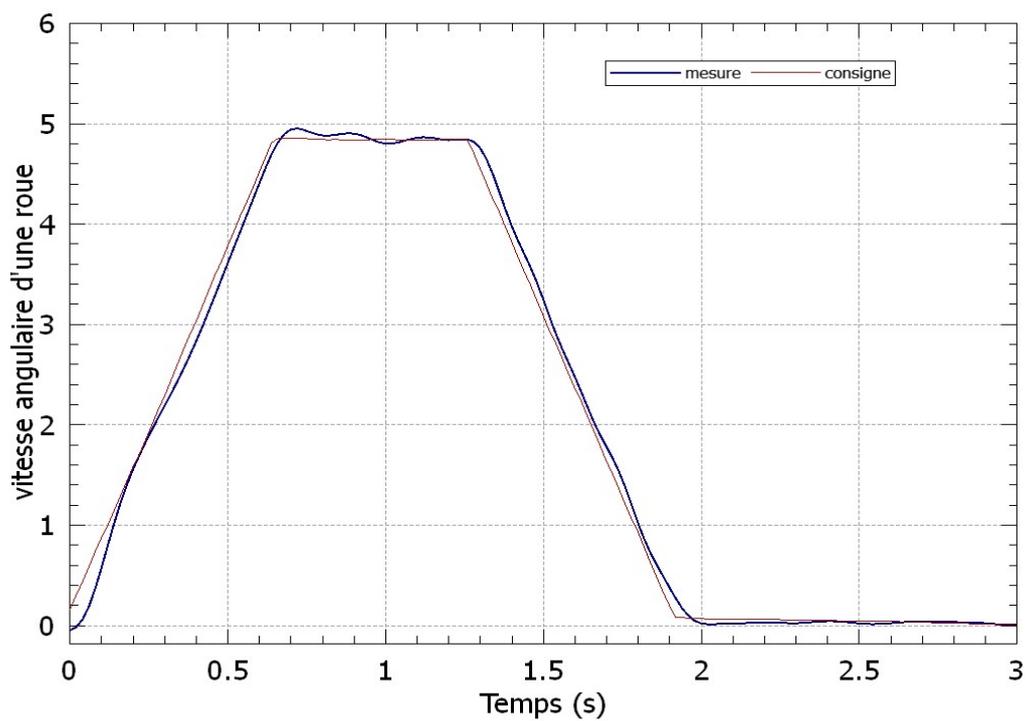


Figure 9 vitesse de la roue droite pour une rotation de 180°

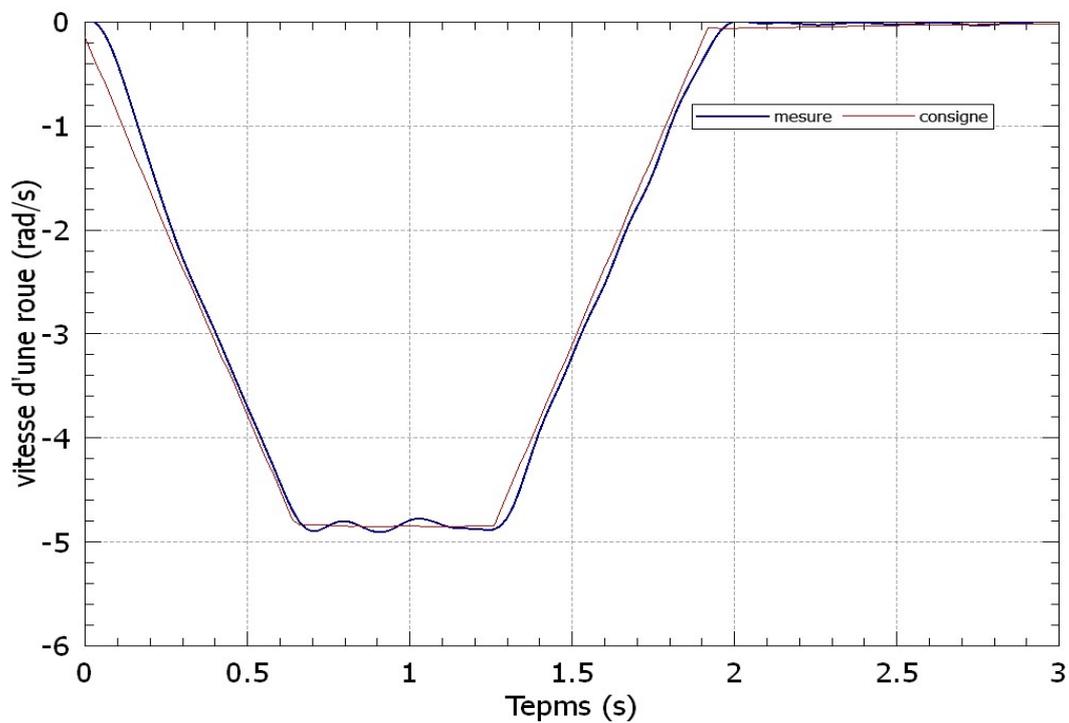


Figure 10 vitesses de la roue gauche pour une rotation symétrique de 180°

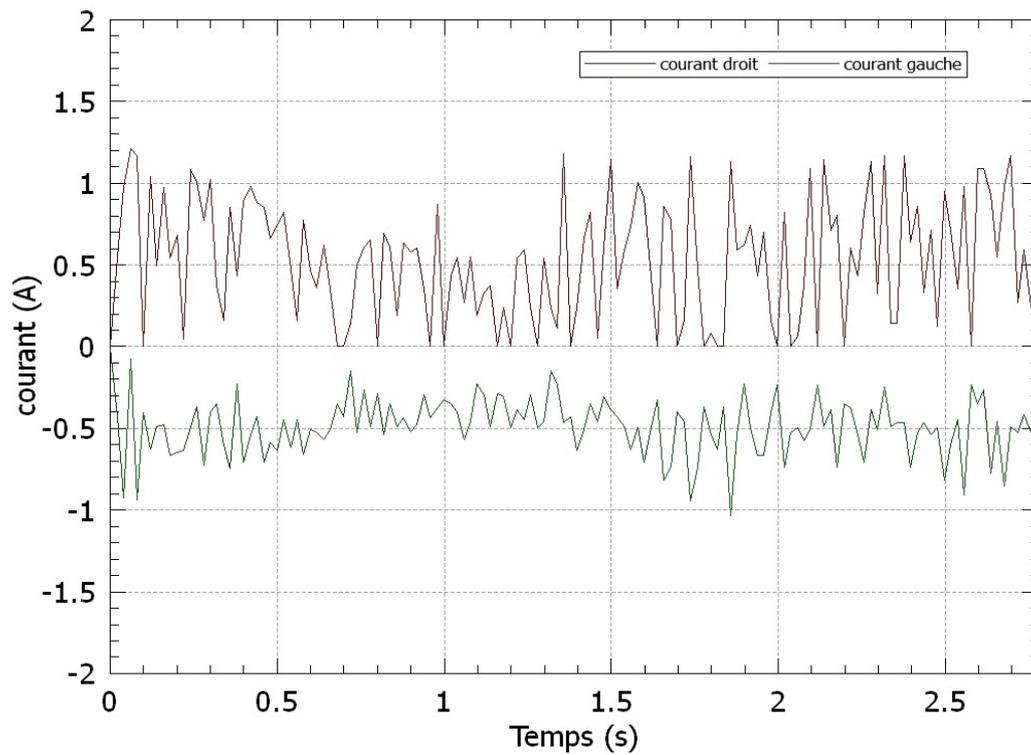


Figure 11 courants consommés par les deux moteurs pour une rotation symétrique de 180°

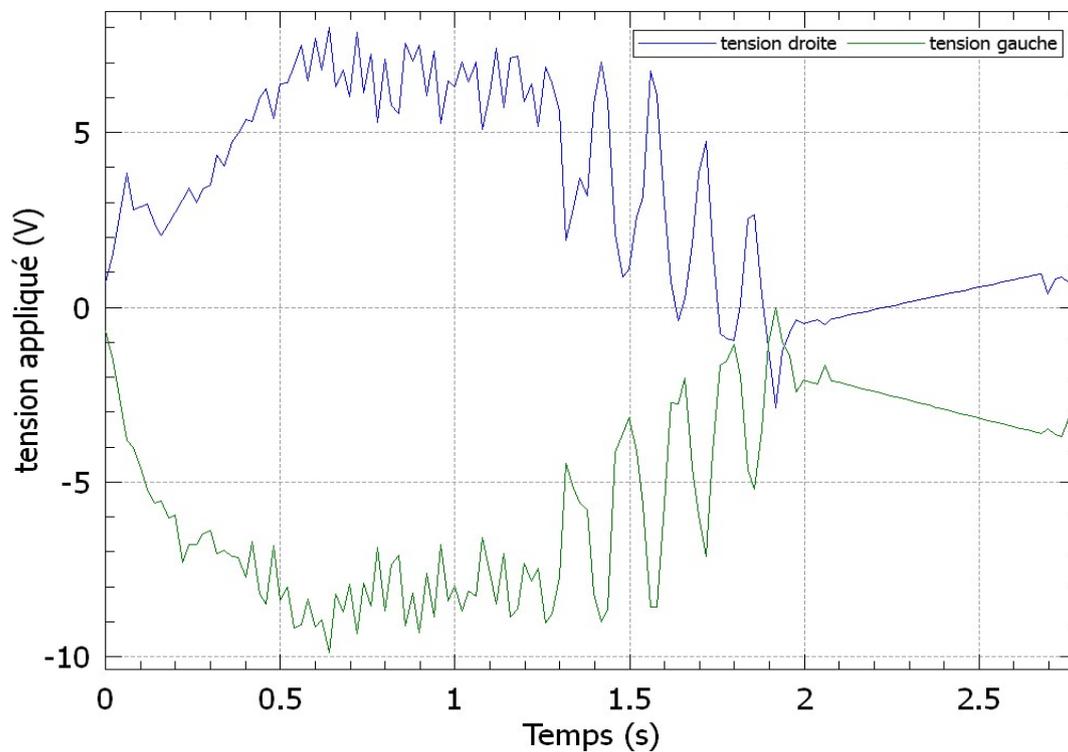


Figure 12 commandes appliquées aux moteurs pour une rotation symétrique de 180°

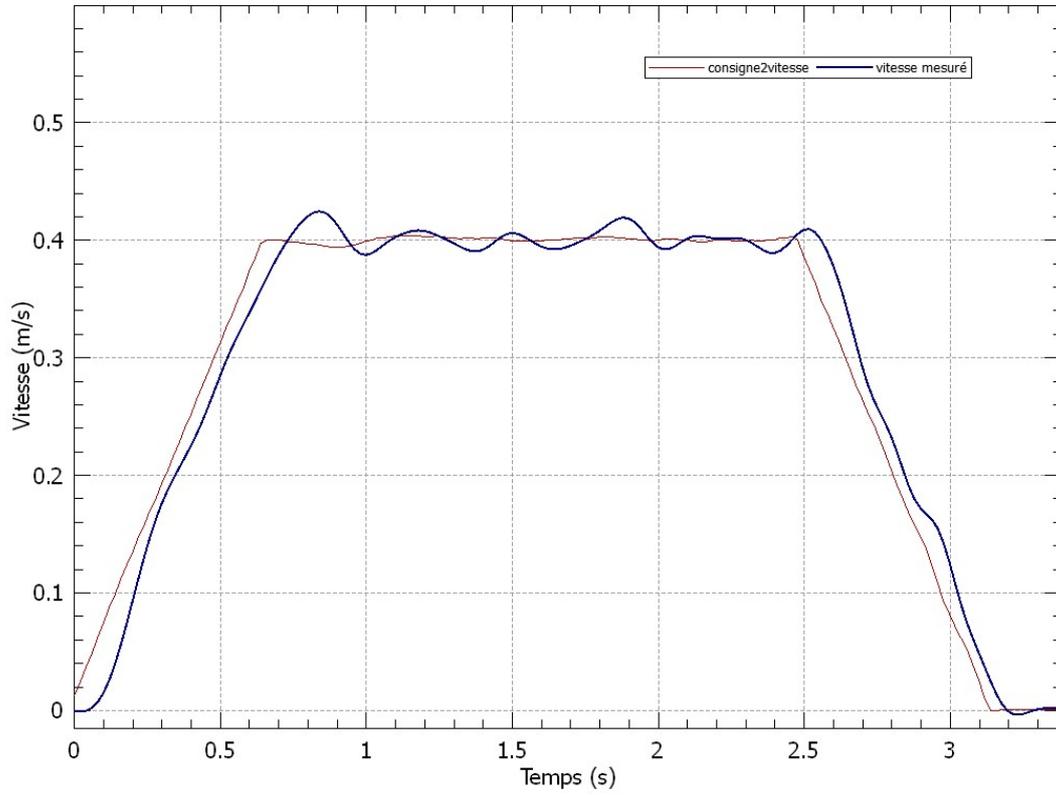


Figure 13 vitesses de la roue droite pour une distance de 1000 mm.

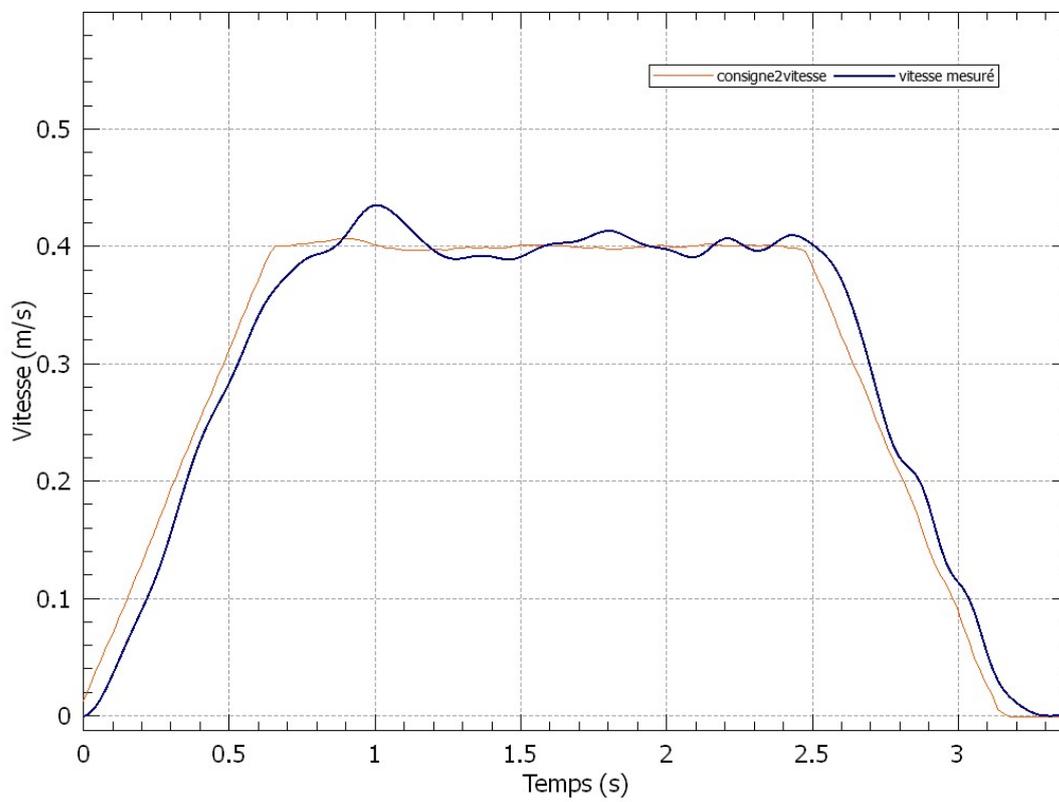


Figure 14 vitesses de la roue gauche pour une distance de 1000 mm

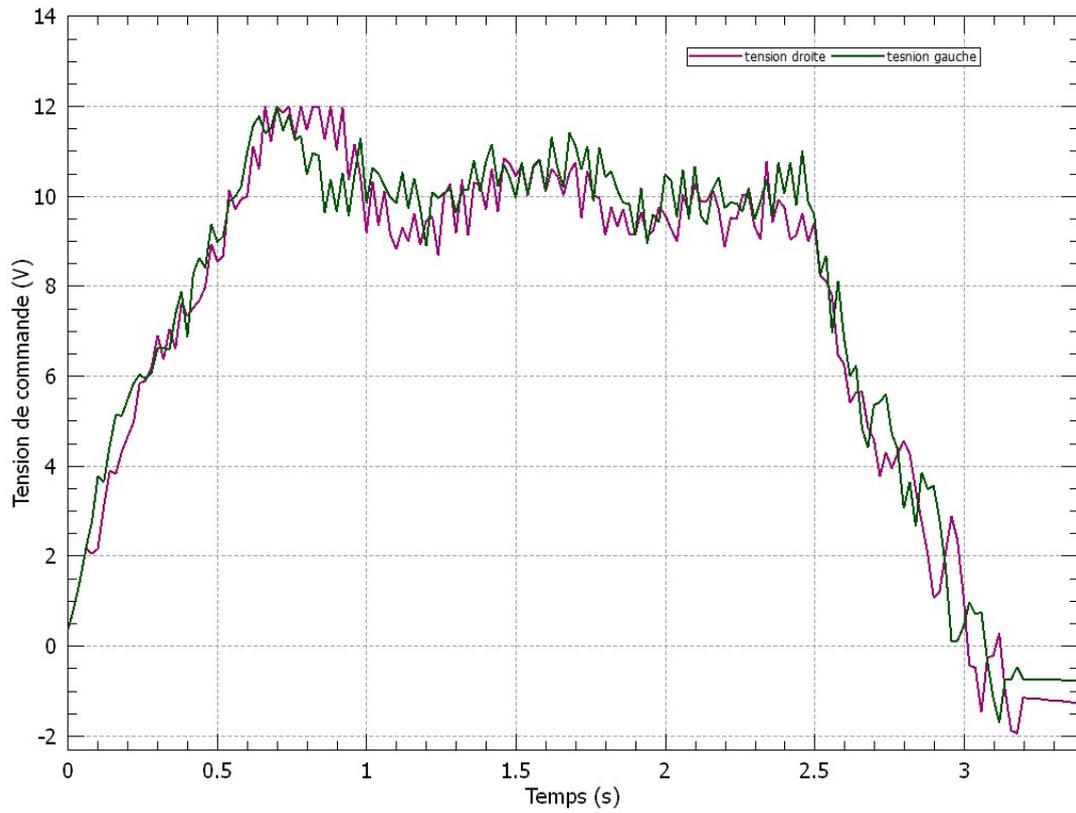


Figure 15 commandes appliquées aux moteurs pour une distance de 1000mm

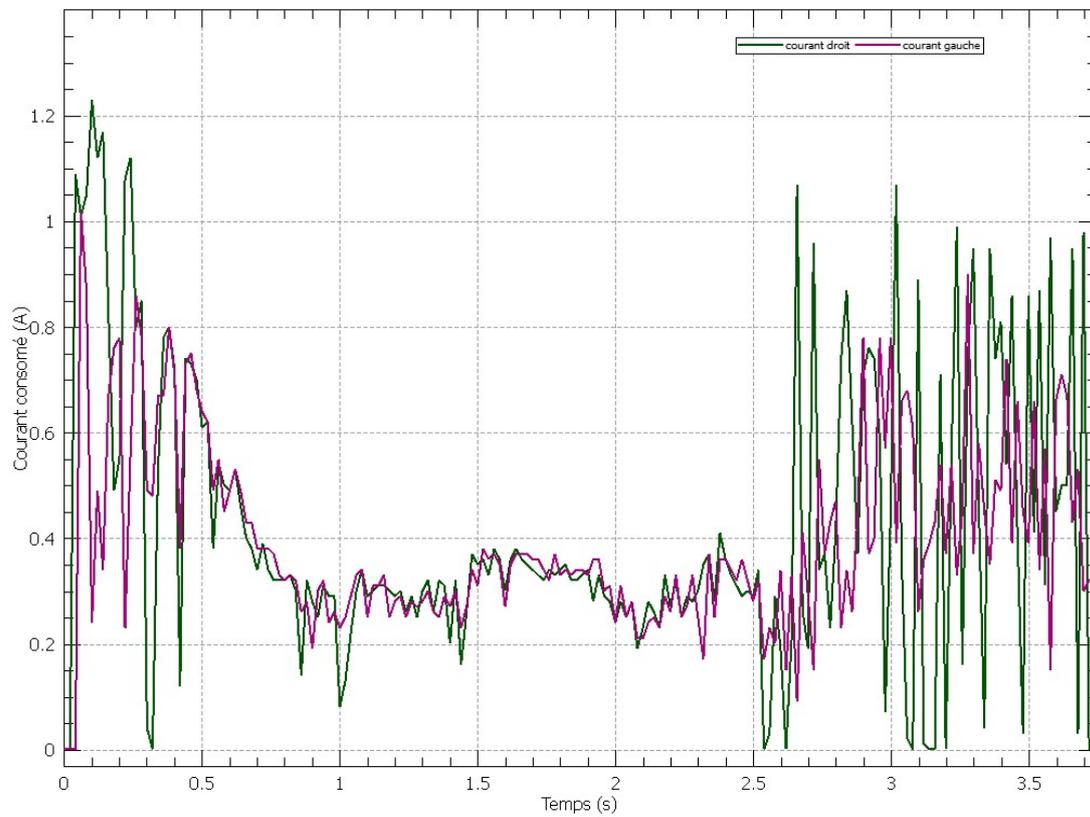


Figure 16 courant consommé par les deux moteurs pour une distance de 1000 mm

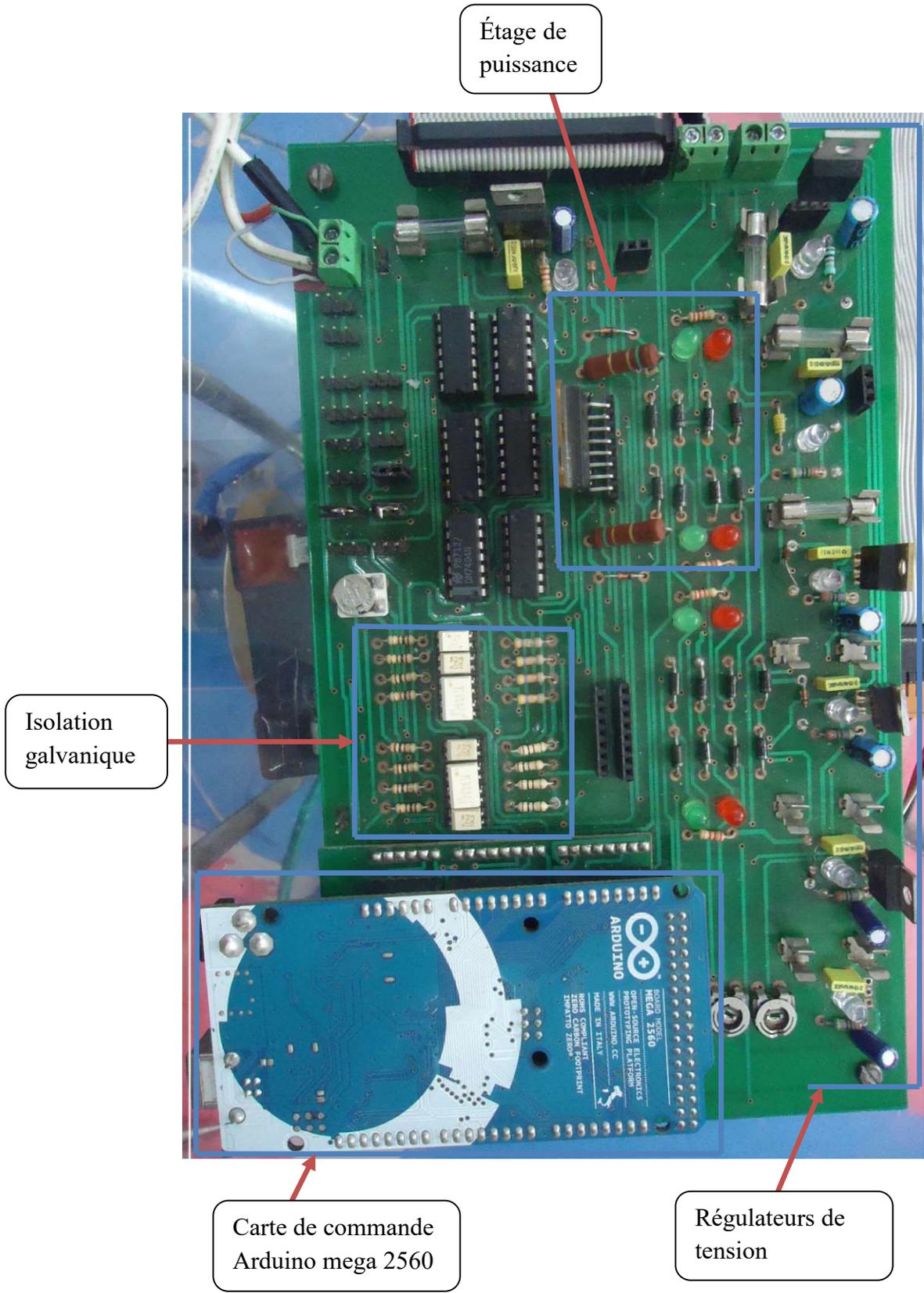


Figure 1 photo de la carte réalisée

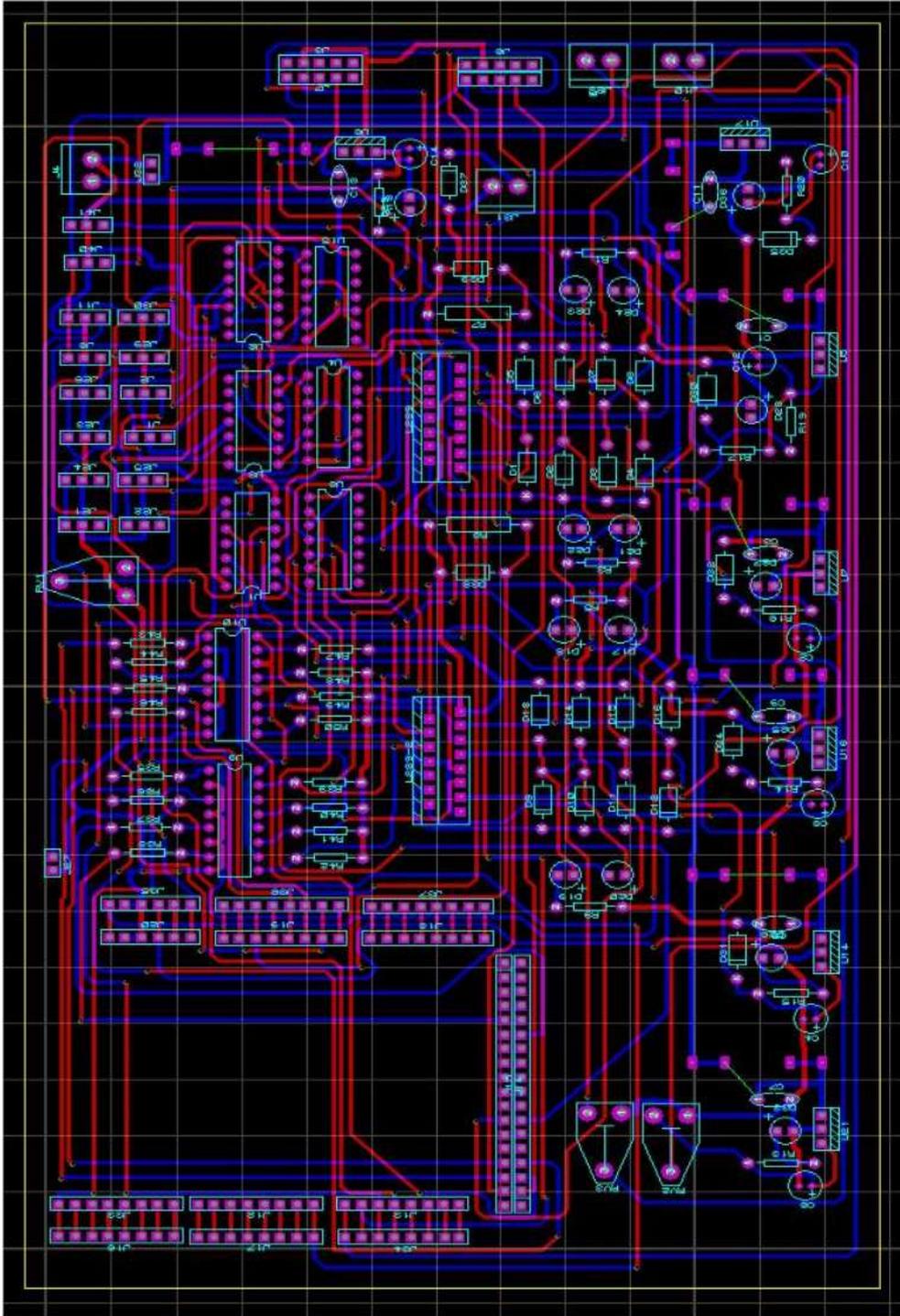


Figure 2 schéma de la carte réalisée

BIBLIOGRAPHIE

- [1] <http://www.roboticus.org/electronique/29-introduction-a-la-robotique>
- [2] B.BAYLE, « Robotique mobile », Ecole Nationale Supérieure de Physique de Strasbourg Université Louis Pasteur, année 2008–2009
- [3] <http://www.eurobot.org>
- [4] P. Rousseau ,«Asservissement linéaires continus», Ellipses, Paris, 2004.
- [5] H.Egon, M.Marie, P.Porée «Traitement du signal et automatique», Hermann, Paris, 2000.
- [6] P. Clerc « Automatique Continue & Echantillonnée », Masson, Paris, 1996.
- [7] http://fr.wikipedia.org/wiki/Machine_%C3%A0_courant_continu
- [8] B. Deforge & Q. David «Asservissement en position d'un axe linéaire», projet d'automatique, 2008.
- [9] C. le Lann «Le PID utilisé en régulation de position et/ou de vitesse de moteurs électriques», Projet de fin d'étude, 2007.
- [10] M. A. Johnson & Mohammed H.Moradi «PID Control (New identification and design methods) », Springer, 2005
- [11] K. J. Astrom & T. Hagglund «Advanced PID Control», ISA, New York, 2006.
- [12] Prof. F. Mudry Ajustage des Paramètres d'un Régulateur PID note et application mars 2006.
- [13] B. C. Kuo & F. Golnaraghi «Automatic Control Systems», John wiley and sons, New York, 2003.
- [14] Matlab tutorial — Designing a PID Compensator for a Single-Loop Feedback System
- [15] mega2560 datasheet
- [16] G.Séguier, R.Bausière, F.Labrique « Electronique de puissance : structures, fonctions de base, principales applications », Paris, Dunod, 2004.
- [17] <http://.en.wikipedia.org/wiki/H-bridge>
- [18] l298 datasheet
- [19] Siwar Rais,Cédric Le Ninivin, Samuel Mokrani « Asservissement PID, PWM et Ponts en H»,projet de fin d'etudes,24 mars 2011

[20] <http://homepages.which.net/paul.hills/SpeedControl>

[21] http://fr.wikipedia.org/wiki/Int%C3%A9grale_de_Riemann

[22] J.Beaudry, projet spinos « conception et contrôle d'un robot mobile a vitesse différentielle », projet de fin d'études, école polytechnique de Montréal, décembre 2001

[23] <http://www.arduino.cc>

[24] S.Tinguely, « Odométrie et suivi de trajectoires appliqués à une base robotique roulante », Projet de Semestre, école polytechnique fédérale de Lausanne, 2006

[25] J.Pinkasfeld, S.Morvan, P.Morelle, « Rapport de TX Projet de Robotique », université de technologie Compiègne.2007

[26] <http://www.RCVA.fr>

[27] k.kara, support du cours « traitement numérique du signal », master automatique, université Saad Dahleb-Blida, 2009