الجمهورية الجزائرية الديمقراطية الشعبية

MINISTERE DE L'ENSEIGNEMENT SUPERIEUR

ET DE LA RECHERCHE SCIENTIFIQUE

UNIVERSITE DR. YAHIA FARES DE MEDEA

FACULTE DE LA TECHNOLOGIE

DÉPARTEMENTDUGÉNIEDELAMÉCANIQUE



Mémoire de Fin d'Etudes

Présenté pour l'obtention du Diplôme

Master en Génie Mécanique

Option : Mécatronique

Robot équilibriste commandé par supervision via WIFI

Proposé et dirigé par :

Prof. Abudura Salam

Co-promoteur :

Dr. Belhadj Abdelmonaim

Préparé et Présenté Par :

Mr. Saadoudi Abdelmadjid

Mr.Benrami youcef

Année Universitaire : 2016/2017

Dédicace

Je dédis ce mémoire a ma mère qui ma encouragé a aller de l'avant et qui m'a donné tout son amour pour reprendre mes études.

A la mémoire de mon père qui nous a quittés voilà vingt ans.

A mon grand père

A mon frère : Walid et ma Sour et son enfants : Abdou.

A toutes ma familleBenRami et Moussaoui.

A mon meilleur ami : larbi Mohamed et toutes mes chers amis qui j'ai partagé des moments des plus agréable.

A tout ceux qui sont chères, proche mon cœur, et a tout ceux qui m'aiment et qui aurait voulu partager ma joie….

Dédicace

Je dédis ce mémoire a ma mère qui ma encouragé à aller de l'avant et qui m'a donné tout son amour pour reprendre mes études.

Et mon père qui ma pousser de faire plus de plus pour mon carrière qui me support dans tout ma route

Àma famille Saadoudi

À mon frère Imed et mes sœurs

A mes meilleurs amis et collègues : Hadji, Krimo, Farouk, Izo, Yasser, et mon binôme Youcef

A tous ceux qui sont chères, proche à mon cœur, ettous ceux qui m'aiment et qui aurait voulu partager ma joie….

Remerciements

A mon encadrement Mr ABODORA SALEM

J'ai en le privilège de profiter de vous vastes connaissance, ansai que votre profond savoir faire, j'ai aussi apprécié vos qualités humaines et professionnelles qui ont suscité mon administration.

Croyezmonsieur, à mon profond respect a ma gratitude et mes sentiments de haute estimer.

A mon membre de jury.

Je tiens a remercie le chef de département de géniemécanique Mr ZIRARI MONIR, pour ses précieux conseils durant ces cinq années de formation.

Je terminerai en remerciant toute les enseignent se département science et technologieet surtout les professeurs de spécialité mécatronique, pour cette formation de qualité.

Merci et merci et mille mercis

ملخص

مشروع البحث العلمي المقدم يتمثل في دراسة الروبوت المتوازن على عجلتين ومعالجة نظام الميكاترونيك الذي يمكننا من التحكم المثالي في الروبوت وبمساعدة الواجهة الرسومية للما تلاب نستطيع متابعة الأداء. ونقترح في هذا المشروع تطبيق الأندر ويد على الهواتف النقلة لعملية التحكم في الروبوت، ونوع التحكم الذي ندرسه هو نوع من التواصل اللاسلكي عن طريق الوايفاي وهذا المقترح يعتمد على تطبيق الأندر ويد مما يسمح لنا في التحكم بالنظام عن بعد وبذالك تم وضع طريقة للتواصل، وكل هذا يعتمد على قطعة الأردو ينو والوايفاي المكمل لها

هذا نظام الميكاترونيك يتألف من البندول المقلوب المحمول ذو العجلتين للبقاء في توازن مع التحكم عن بعد في وجهته. تصميم هذا الروبوت قدم الكثير من التحديات ومن بينها أن نضامه غير مستقر مما يتطلب مصحح لاستقراره وبذالك الحفاظ على توازه

ولكي نستطيع تصميم متحكم، تم تحليل ديناميكية الروبوت وتطوير نموذج رياضي الى شكل نماذج الحالات الخطية. وخوارزميات التصفية تسمح للحصول على نتائج مقدرة على الرغم من وجود تشويش على الإشارات

الكلمات المفتاحية :الروبوت المتوازنعلى عجلتين, الميكاترونيك, الأردو ينو والوايفاي, نماذج الحالات الخطية, التواصل اللاسلكي

RÉSUMÉ

Le projet de recherche présenté dans ce document est étudié le robot équilibriste et traiter le système mécatroniques qui géré le robot parfaitement que nous espérons et à l'aide d'une interface graphique sous Matlabe on peut suivie sa performance. Et proposer une application Android pour faire contrôle le robot, ce type de commande qu'on a étudié est un exemple de communication par WI-FI. Ce ci proposée est basée sur une application Android

Afin de pouvoir commander à distance des montages mécatroniques, une architecture de communication a été mise en place, Et avec l'appareil Arduino et son complément wifi, notre communication sera réalisée pour contrôler le robot.

Ce système mécatronique consistant en un Pendule inversé mobile à deux roues a été conçu pour rester en état équilibre générée à distance par des command de control sa destination.

La conception du robot équilibriste présente plusieurs défis dont le fait que celui-ci est un système Instable et requiert donc un correcteur pour le stabiliser en le maintenant en équilibre.

Afin de pouvoir concevoir ce contrôleur, la dynamique du robot a d'abord été analysée et un modèle mathématique de celle-ci a été développé sous la forme de modèles d'états linéaires. Des algorithmes de filtrage permettant d'obtenir de bons estimés des états malgré la présence de bruits dans les signaux.

- Mots clés :robot équilibriste, interface graphique, application Android, WI-FI, command à distance, modèles d'états linéaires, Arduino,

ABSTRAC

The research project presented in this paper studies the balancing robot and treats the mechatronic systems that generate perfectly the robot, and using a graphical interface under Matlabe. Its objective is to follow the performance of this mechatronic system, and propose an Android application to control the robot remotely; this type of control that has been studied is an example of communication via Wi-Fi. This one proposed is based on an Android application

In order to be able to remotely control mechatronic assemblies, a communication architecture has been set up, and with the Arduino device and its Wi-Fi complement, our communication will be carried out to control the robot.

This mechatronic system consisting of a two-wheeled reversed mobile pendulum was designed to remain in equilibrium state generated remotely by controllers to control its destination.

The designed balancing robot presents several challenges, the fact that it is an unstable system and therefore requires a corrector to stabilize it by keeping it in equilibrium.

In order to design this controller, the dynamics of the robot were first analyzed and a mathematical model of the latter was developed in the form of linear state models. Filtering algorithms for obtaining good state estimates despite the presence of noise in the sensor signals have also been implemented.

- KEY WORD : balancing robot, mechatronic, graphical interface, Android application, remote control, Wi-Fi, linear state models, Arduino

TABLE DES MATIÈRES

ملخص. . . . . .

IRÉSUMÉ. . . . . .
II

ABSTRAC. III

TABLE DES MATIÈRES. IV

LISTE DES TABLEAUX. VIII

LISTE DES FIGURES. IX

LISTE DES NOTATIONS ET DES SYMBOLES. XI

LISTE DES ACRONYMES. XIII

INTRODUCTION GÉNÉRALE. XIV

CHAPITRE 1 DÉFINITION DU PROBLÈME

1 Introduction 1

1.1 Robot équilibriste. 2

1.1.1 Revue de la littérature 2

1.1.2 Conception 4

1.1.3 Modélisation Mathématique 4

1.2 Commande via Internet 4

1.2.1 Application web 4

1.3 Commande via WIFI 5

1.3.1 Application Android 5

1.4 Interface graphique sous Matlabe 6

1 Conclusion 6

CHAPITRE 2 CONCEPTION DU ROBOT ÉQUILIBRISTE

2 Introduction 8

2.1 Mécanique 8

2.1.1 Actionneurs 9

2.2 Électronique 9

2.2.1 Alimentation 10

2.2.2 Capteurs 11

2.2.2.1 Gyroscope 12

2.2.2.2 Accéléromètre 13

2.2.2.3 Encodeurs Optiques 14

2.2.3 Pont H 15

2.2.4 Communication sans fil 16

2.2.5 Arduino 16

2.2.5.1 Principe de fonctionnement des interruptions 17

2.2.5.3 Temporisateurs 19

2.2.5.4 Convertisseur analogique numérique (ADC) 19

2.2.5.5 Modulation de largeur d'impulsion (PWM) 20

2 Conclusion 21

CHAPITRE 3 MODÉLISATION MATHÉMATIQUE DU ROBOT

3 Introduction 22

3.1 Inclinaison et déplacement linéaire 23

3.1.1 Dynamique d'un moteur. 24

3.1.2 Couple appliqué à une roue 24

3.1.3 Dynamique d'inclinaison 25

3.1.4 Dynamique des roues et du déplacement linéaire 26

3.1.5 Modèle d'état de la dynamique linéaire 27

3.2 Angle de direction. 28

3.2.1 Modèle d'état de la dynamique de direction 31
3.3 Perturbations externes 32
3.3.1 Dynamique d'inclinaison et de déplacement linéaire 33
3.3.2 Dynamique de direction 33
3 Conclusion 34
CHAPITRE 4 LE SYSTÈME EN TEMPS RÉEL
4 Introduction 35
4.1 Filtrage 35
4.1.1 Angle d'inclinaison 36
4.1.1.1 Estimé du gyroscope 36
4.1.1.2 Estimé de l'accéléromètre. 37
4.1.1.3 Filtres complémentaires 37
4.1.2 Filtres pour positions des roues 40
4.2 Correcteur 40
4.2.1 Références et intégrateurs 40
4.2.2 Structure du correcteur 43
4.2.3 Placement de pôles 44
4.2.3.1 Dynamique linéaire 44
4.2.3.2 Dynamique de direction 45
4.3 Fréquence d'échantillonnage 46
4.3.1 Filtre du gyroscope 47
4.3.2 Filtre de l'accéléromètre 47
4 Conclusion 48
CHAPITRE 5 COMMANDE VIA WIFI
5 Introduction 49
5.1 Architecture de communication 49
5.1.1 Émetteur-récepteur Wi-Fi 50

5.1.2 Robot 51
5.1.2.1 Protocol TCP/IP 51
5.1.2.2 Protocol SPI 51
5.2 Android App 52
5.2.1 La configuration de l'application avec l'Arduino 52
5.2.2 Explication de programmation 53
5.2.2.1 Le code 55
5.3 Interface graphique 56
5.3.1 Développement sous Matlabe (Guide) 58
5 Conclusion 58
Conclusion générale 59
Référence 60

LISTE DES TABLEAUX

2.1 Composantes du système 10

2.2 Fonctionnement d’un moteur à partir de L293D 15

3.1 Paramètres du système. 23

LISTE DES FIGURES

1.1 : JOE …… 4

1.2 : robot de Rich Chi Ooi ...…… 4

2.1 : robot équilibriste …… 8

2.2 : Architecture du système numérique …… 10

2.3 : Circuit d'alimentation ……...11

2.4 : Gyroscope IDG-300 ……. 12

2.5 : accéléromètre LIS3LV02DQ .……13

2.6 : signaux produits par un encodeur optique ……14

2.7 : pont H L293D ……...16

2.8 : WIFI module ESP8266 ……...16

2.9 : plaque Adruino UNO ……...17

2.10 : analogique / numérique ……..20

3.1 Inclinaison et déplacement linéaire du robot …….. 24

3.2 Diagramme de corps libre de la dynamique d'inclinaison ……..25

3.3 Diagramme de corps libre d'une roue ……. 26

3.4 Déplacements des roues lors d'un changement de direction……28

3.5 Diagramme de corps libre de la dynamique d'angle de direction ……28

4.1 Traitement de signaux et estimation des états …….36

4.2 : Accéléromètre utilisé en tant qu'inclinomètre …….37

4.3 : Diagramme de Bode du filtre passe-bas G_a (s) ……..38

4.4 : Diagramme de Bode du filtre passe-haut G_g (s) …….39

4.5 : Diagrammes de Bode des filtres complémentaires G_a (s) G_g (s) …….39

4.6 : Structure globale du correcteur par retour d'état…….. 44

4.7 : Schéma de blocage pour le contrôleur Pole-placement …… 44

4.8 : Réponse du système avec le contrôle de placement de pôles…….. 48

5.1 :l'interface de commande et configuration ……..49

5.2 : architecture de communication ……. 50

5.3: architecture d'UE WIFI interface …….50

5.4 : lire /écrire depuis SD mémoire ……..51

5.5 : schéma de contrôle ……52

5.5 : diagramme de test robot montée un dos d'âne …….. 57

5.6 : interface graphique ……57

LISTE DES NOTATIONS ET DES SYMBOLES

$\delta(t)$: angle de direction

$\theta(t)$: angle d'une roue

$\theta_i(t)$: angle de l'arbre moteur

$\theta_l(t)$: angle de la roue gauche

$\theta_o(t)$: angle de l'arbre de la boîte de réduction

$\theta_r(t)$: angle de la roue droite

$w_x(t)$: perturbation externe linéaire

$w_h(t)$: perturbation externe de direction

$\psi(t)$: angle d'inclinaison

ζ: facteur d'amortissement

$F(t)$: force appliqué au sol par une roue

$F_l(t)$: force appliqué au sol par la roue gauche

$F_r(t)$: force appliqué au sol par la roue droite

$T(t)$: couple livré à une roue

$T_i(t)$: couple livré à la boîte de réduction par un moteur à courant continu

$T_l(t)$: couple livré à la roue gauche

$T_r(t)$: couple livré à la roue droite

S: distance entre les roues

$u_h(t)$: portion de la tension contrôlant la direction

$u_h(t)$: tension envoyée au moteur gauche

$u_r(t)$: tension envoyée au moteur droit

$u_x(t)$: tension moyenne envoyée aux moteurs

$x(t)$: position du robot

$x_l(t)$: position de la roue gauche

$x_r(t)$: position de la roue droite

r_w : rayon des roues

M : masse de la moitié robot, y compris une roue

m_b: masse de la moitié du corps du robot

m_w : masse de l'une des roues

J_b : moment d'inertie de la moitié du corps du robot

J_d : moment d'inertie du robot autour de l'axe Z

J_w : moment d'inertie de l'une des roues

d : distance entre l'arbre du moteur et le centre de gravité

K_t : constante de couple des moteurs

K_e : constante de force électromotrice des moteurs

r_a : résistance d'armature des moteurs

r_g : rapport de la boîte de réduction

η : efficacité de la boîte de réduction

C_f : constante de friction

LISTE DES ACRONYMES

WIFI	Wireless Fidelity
UWA	Université of Western Australia

CGI Common Gateway Interface

LQRLinear Quadratic Regulator

HTMLHyperText Transfer Protocol

TCP-IP Transmission Control Protocol / Internet Protocol

AJAXAsynchronous JavaScript and XML

J2EEJava 2.0 Enterprise Edition

IBSSIndependent Basic Set service

JDK Java Development Kit

SDKSource Developing Kit

IDE Integrated development environment

API application programming interface

CAOConception Assistée par Ordinateur

DC direct courante

PDIPProgrammable Indicator Data Processor

TX/RX Transmitter/Receiver

UEUser Equipment

PWMPulse Width Modulation

ISR Interrupt Service Routine

ADCAnalog-to-Digital Converter

DELled

FPGAField Programmable Gate Array

D2DDevice-to-Device

INTRODUCTION GÉNÉRALE

Notre projet a été adopté au domaine de mécatronique quand on parle sur un robot, n'importe quel robot c'est une à couplage mécanique-électronique dans un seul système qui nous donne plus contrôle possible automatique ou manuelle, ce domaine est tellement vaste touche tous les sections industriels figure 1 peut expliquer les différent part pour un tout assemblage mécatronique.

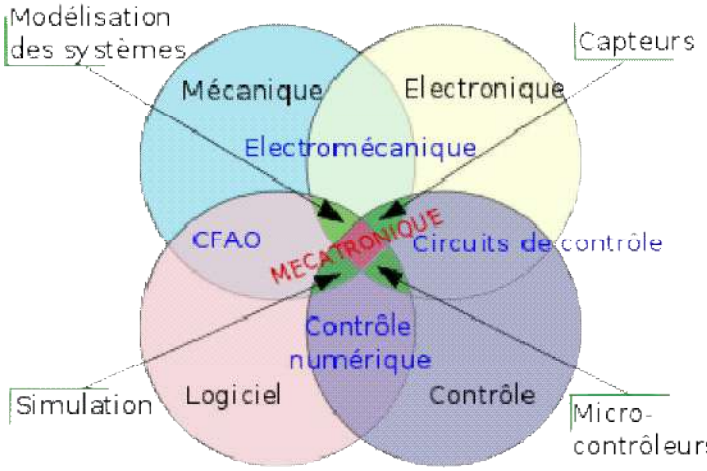


Figure 1 : Un diagramme de Venn montrant la mécatronique

L'ingénierie de tels systèmes mécatroniques nécessite la conception simultanée et pluridisciplinaire de 3 sous-systèmes :

-une partie opérative (squelette et muscle du système à dominante Mécanique et Électromécanique).

-une partie commande (intelligence embarquée du système à dominante Électronique et Informatique Temps Réel).

-une partie interface Homme/Machine (forme géométrique et dialogue du système à dominante Ergonomique et Esthétique).

Une approche globale permet aussi de réduire les coûts, d'augmenter la fiabilité et la modularité.

Le terme mechatronics a été introduit par un ingénieur de la compagnie japonaise « Yaskawa Electric Corporation »

La norme NF E 01-010 (2008) définit la mécatronique comme une « démarche visant l'intégration en synergie de la mécanique, l'électronique, l'automatique et l'informatique dans la conception et la fabrication d'un produit en vue d'augmenter et/ou d'optimiser sa fonctionnalité »

Les robots équilibristes Ont attiré de nombreux chercheurs intéressésla dernière décennie dans un certain nombre de robotiques laboratoires à travers le monde, cette recherche décrit la

conception de matérielle d'un robot équilibriste pour une bonne résulta do mouvement et de communication

Les systèmes de commande sont présents partout autour de nous. Ceux-ci sont utilisés aussi bien dans les produits commerciaux que dans les applications industrielles et militaires.

Traditionnellement, ces systèmes étaient analogiques et leurs différentes composantes telles que des capteurs, actionneurs et circuits de commande, étaient inter-reliées par des connexions physiques sur lesquelles des signaux analogiques étaient transmis.

De nos jours, avec l'évolution des ordinateurs et des technologies de l'information, la plupart des systèmes de commande sont conçus en tant que systèmes numériques à base de microprocesseur. Ceci permet donc à leurs différentes composantes d'être reliées par des connexions logiques sur un réseau de communication. On parle alors de système de commande par réseau (*Networked Control System* - NCS).

Le domaine de la commande par réseau est large. Celui-ci comprend principalement la commande distribuée et la commande par supervision ou télé opération. La commande distribuée adresse les systèmes dans lesquels des nœuds de capteurs, d'actionneurs et de contrôleurs sont géographiquement distribués et en liaison à travers un réseau de communication.

Le projet de recherche présenté dans ce document se situe dans le cadre des systèmes de commande par WI-FI. Ce type de système est généralement utilisé dans des applications où l'opération d'un équipement ou la réalisation d'une tâche demande une expertise spécialisée. On pense par exemple à des applications médicales comme des systèmes de chirurgie à distance, ou encore des applications où la tâche à réaliser doit être effectuée dans un environnement dangereux pour un ouvrier comme l'exploration minière. On retrouve également des applications industrielles dans lesquelles des machines de production ne peuvent être complètement automatisées puisqu'elles doivent remplir des fonctionnalités diverses et un opérateur spécialisé doit intervenir régulièrement pour assurer un déroulement efficace de l'opération. La commande par supervision via WI-FI est également utilisée dans le cadre de l'enseignement de la mécatronique et des systèmes de commande. C'est dans ce contexte que le projet présenté ici a vu le jour. L'idée initiale du projet était de concevoir un système permettant la commande de plusieurs montages mécatroniques à distance à travers le réseau sans fil et lorsque nous parlons de cela, nous devrions prendre l'Internet en considération comme une vaste gamme de contrôle à partir d'une distance vraiment longue grâce à le réseau de communication.

Comme exemple de système de commander et contrôler par Wifi « Home Automation System Using Android and Wifi », c'est un système, qui implémente sur Android System, qui est populaire et profondément disponible La domotique peut être définie comme un système mis en œuvre dans lieu d'habitation où l'intention est de rendre le lieu intelligente de sorte que l'énergie est conservée et la sécurité est maintenu.

À titre d'exemple de laboratoire virtuel, (Lassó&Urbancsek, 2001) présentent un système robotique composé d'un micro robot muni d'un microscope capable de se déplacer dans son environnement alors qu'une caméra surveille l'environnement global et qu'une autre permet d'obtenir l'image du microscope. Ce travail utilise une approche basée sur les appliquettes Java (*Java Applets*) et propose une combinaison des protocoles *Transfert Control Protocol* (TCP) et *User Datagram Protocol* (UDP) pour les différents aspects de son application. On y propose également une technique d'encodage d'images à multiples résolutions afin de réduire le débit requis pour les transmettre.

Comme exemple de système de commande par supervision via Internet utilisé dans le cadre de l'enseignement de la mécatronique, on peut citer (Lindsay, Good &Halgamuge, 2000) où on présente un système permettant la commande d'un robot industriel CRS F3 à partir d'une appliquette Java intégrée dans une page web. Ce système a vu le jour à l'Université de Melbourne dans le cadre de la mise à jour ses pratiques d'enseignement de la mécatronique suite aux avancements des technologies de l'information et de l'Internet.

Un autre exemple de système développé dans le cadre de l'enseignement de la mécatronique est présenté par (Roesh, Roth &Nicolescu, 2005). Les auteurs présentent un système de commande d'un pendule démontrant le principe de la commande de structures mécaniques flexibles. Le contrôleur temps réel du système est implémenté localement sur un PC à l'aide des logiciels Matlabe, Simulink et WinCon alors qu'une interface de programmation composée d'un ensemble de *Dynamic Link Library* (DLL) permet l'intégration du logiciel Matlabe et d'une application serveur en Java. L'interface utilisateur est implémentée en tant qu'appliquette Java exécutée dans une page web et communique avec l'application serveur via Internet.

Le but principal de ce projet est de proposer et étudier un system mécatronique commandé par distance pour éviter le câblage et pour avoir une large surface de control sur le robot et à partir de ça on est soulignée certain points pour attendre dans ce projet de recherche

- Proposer et concevoir une conception pour le system mécanique
- Etudier et modéliser la dynamique d'inclinaison et le déplacement
- développer un modèled'état linéaire simple mais efficace pour la dynamique de l'inclinaison et du déplacementlinéaire
- mettre en place un system asservie pour assurer le mouvement
- proposer une architecture de communication

Le document est organisé de la manière suivante. Dans le chapitre 1, le problème est défini et les différentes parties du projet sont introduites.

Le chapitre 2 présente la conception mécanique et électronique du robot équilibriste utilisé la commande par supervision via wifi.

Le chapitre 3 traite de la modélisation mathématique de ce système mécatronique.

Le chapitre 4 porte sur le système en temps réel du robot incluant la conception d'algorithmes de commande et leur implémentation sur un système à base d'un Arduino.

Le chapitre 5 traite de l'architecture de communication mise en place afin de permettre la commande à distance et de l'application Android permettant à des utilisateurs de commander et l'interface Matlabe de surveiller le système mécatronique en temps réel

CHAPITRE 1

DÉFINITION DU PROBLÈME

Introduction

Le problème considéré dans ce projet de recherche consiste en la commande d'un robot équilibriste à distance. Ceci présente plusieurs défis considérables. Ceux-ci entrent dans deux grandes catégories : la commande des systèmes dynamiques en temps réel et la commande par WI-FI.

Dans le cadre de la commande par WI-FI, afin de pouvoir commander un système à distance, il est d'abord nécessaire de concevoir un système de commande local. Puisque ce système est en interaction constante avec son environnement, celui-ci doit être en mesure de répondre à des événements à l'intérieur de délais raisonnables afin de pouvoir accomplir la tâche avec succès. On parle alors d'un système temps réel, et il est donc nécessaire d'avoir recours aux techniques de programmation propres à ce type de système.

De plus, puisque le montage mécatronique consistant en un pendule inversé mobile est un système instable. il est nécessaire de concevoir un contrôleur capable de le stabiliser.

Dans le but de réduire les coûts du montage conçu et le temps, le système de commande et le system de contrôle est basé sur des composantes électroniques accessibles tel qu'un Arduino et son compliment Wifi pour communiquer avec l'ordinateur ou le phone mobile pour une utilisation flexible, commandé facilement à distance. Afin de prendre en considération le fait que les capteurs utilisés offrent des performances limitées, que leurs signaux contiennent du bruit et que le système de commande requiert une bonne précision sur la mesure de l'inclinaison du robot afin de pouvoir le stabiliser, une technique de fusion des signaux de plusieurs capteurs redondant est requise. Un algorithme particulier de traitement de signaux est donc nécessaire.

Puisque le but du projet est de commander un système à distance via Wifi, les problèmes reliés aux communications par Wifi doivent être pris en considération.

Et comme une vue extrême la possibilité d'utilisation d'internet est demandé au but de développement les privilèges de commande à distance, tel que représenté par le robot de l'UWA, les systèmes de commande par supervision via Internet étaient conçus en tant que pages web émettant des requêtes à un serveur qui devaient alors être traitées par un CGI consistant en un programme exécuté par ce serveur. Cette approche demande un haut taux d'échange d'information entre le client et le serveur puisqu'à chaque requête le serveur doit émettre une réponse sous la forme d'une nouvelle page web et ceci entraîne donc des délais

considérables dû aux traitements du serveur et aux échanges de données. Pour cette raison et pour simplifier la façon de communication on utilise le mode réseaux AD-HOC wifi,pour éviter accès internet

1.1Robot équilibriste

Ce system mécatronique été proposé comme un projet d'étude en plusieurs terrains d'expérience de diffèrent domaine dans le monde. Ce system présente plusieurs défis dans le fait que celui-ci est instable dans tous le temps de travail et requiert des améliorations sur la partie électronique sous bas des correcteurs mettre le dans une stable situation sans oublier la partie de conception et montage mécanique et la mise en place des composent et les organes pour bien divisé de poids sur tout la surface du robot.

La conception mécanique de ce robot est constituée d'un corps de trois étages et de deux roues, chacune étant entraînée par un moteur à courant continu muni d'une boîte de réduction. Le corps et les roues ont été conçus en utilisant un logiciel de CAO (Solide Works).

Le système électronique du robot tout dépend d'Arduino, de deux Encodeurs optiques utilisés pour la mesure des positions des roues, d'un accéléromètre à deux axes et d'un gyroscope utilisés pour mesurer l'angle d'inclinaison du robot, de deux puces de ponts en H utilisées pour la commande des moteurs à courant continu, d'un module d'une puce de communication sans fil pour la communication avec un ordinateur ou un téléphone mobile, d'un ensemble piles et de deux régulateurs de tension pour l'alimentation en puissance.

1.1.1 Revue de la littérature

Le sujet des robots équilibristes a attiré de nombreux chercheurs et amateurs en raison des défis qu'il représente. Dans la littérature, on peut trouver un certain nombre de projets intéressants qui méritent d'être mentionnés. Dans (Anderson D., 2007), l'auteur présente un robot nommé *nbot*, donne quelques informations générales sur la fusion des signaux des capteurs pour évaluer l'angle d'inclinaison et donne la structure de base d'un correcteur à retour d'état permettant de garder le robot en équilibre. Mais il ne donne pas de modèle mathématique de la dynamique et ne propose pas de technique de calcul des gains du correcteur. Dans (Cardi, Ener & Wagner, 2005), une équipe d'étudiants diplômés de *Woodruff School of Mechanical Engineering* de *Geogria Institute of Technology*présente un robot équilibriste nommé *The Teetering Buffalo*. Ils développent un modèled'état linéaire simple mais efficace pour la dynamique de l'inclinaison et du déplacement linéaire et expérimentent différentes structures de correcteurs et observateurs par simulations. Toutefois, bien qu'ils

offrent des vidéos de leur robot en action, leur contribution reste limités car ils n'ont pas discuté des aspects pratiques de leur projet et ne présentent pas de résultats expérimentaux détaillés. Dans (Grasser, D'Arrigo, Colombi & Rufer, 2002), une équipe de chercheurs de l'École Polytechnique Fédérale de Lausane présente leurs travaux sur la conception d'un robot nommé *Joe*figure 1.1. Ils développent un modèle mathématique de la dynamique et introduisent l'idée de dissocier la dynamique de l'inclinaison et de la position linéaire de la dynamique de direction. Ils évoquent les problèmes liés à l'utilisation de boîtes de réduction et proposent l'utilisation d'un filtre de premier ordre pour y remédier et une technique de placement de pôles pour calculer les gains d'un correcteur par retour d'état. Toutefois, ils omettent de montrer les détails de l'élaboration de leur modèle, ils ignorent la dynamique des moteurs et utilisent un gyroscope directement pour mesurer l'angle d'inclinaison, ignorant son effet de dérive. Dans (Nawawi & Ahmad, 2008), les auteurs utilisent le modèle développé dans (Grasser D'Arrigo, Colombi & Rufer, 2002) afin de contrôler leur propre robot pendule inversé à deux roues *Two-Wheeled Inverted Pendulum* (T-WIP) en utilisant un correcteur à retour d'état pour lequel les gains ont été calculés avec la technique de placement de pôles. Leur contribution vient du fait qu'ils ont mis en place l'algorithme dans le logiciel Matlab et sont en mesure de contrôler leur robot en temps réel. Cela nécessite une configuration relativement coûteuse et offre des avantages seulement pendant la phase de développement du produit. Dans (Ooi R., 2003) figure 1.2, l'auteur développe un modèle d'état pour la dynamique d'inclinaison et de déplacement linéaire, propose un filtre de Kalman pour estimer l'angle d'inclinaison et sa dérivée par rapport au temps, basé sur un sous-système composé exclusivement de capteurs, et propose un correcteur par retour d'état pour l'inclinaison et le déplacement linéaire et un correcteur PID pour la direction. Il propose également deux méthodes pour le calcul des gains du correcteur de l'inclinaison et du déplacement linéaire : Le *Linear Quadratic Regulator* (LQR) et le placement de pôles et montre des simulations et des résultats expérimentaux qu'il a obtenu. Malheureusement, la technique utilisée pour obtenir le modèle mathématique de la dynamique de l'inclinaison et du déplacement linéaire donne de nombreux termes non-linéaires et certains détails sont perdus après la linéarisation. En outre, l'approche du filtre de Kalman utilisé ne parvient pas à prendre avantage de la dynamique connue du système et exige le réglage manuel par essais et erreurs pour obtenir des résultats acceptables. Dans (Kadir H., 2005), il discrétise le modèle développé dans (Ooi R., 2003) dans le but de concevoir un correcteur discret. Il propose les approches du LQR et du placement de pôles et étudie la performance du système par simulation. Toutefois, sa contribution demeure restreinte car il ne montre pas de résultats expérimentaux. Dans

(Abdollah M., 2006), le chercheur utilise le modèle développé dans (Ooi R., 2003), ajoute la dynamique de direction et propose un algorithme de commande non-linéaire par mode de glissement qu'il étudie par le biais de simulations. Toutefois, il ne donne pas de résultats expérimentaux et son algorithme est très intensif en calculs et serait difficile à implémenter numériquement.



Figure 1.1 : JOE

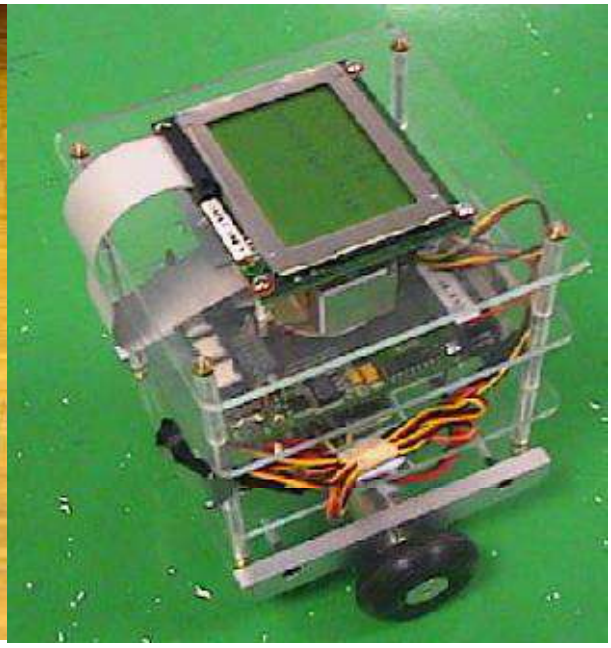


Figure 1.2 : robot de Rich Chi Ooi

1.1.2 Conception

Le robot équilibriste conçu est composé d'une plaque Arduino, de deux moteurs à courant continu, de deux puces de pont en H, de batteries, de deux régulateurs de tension d'un gyroscope, d'un accéléromètre, d'encodeurs optiques et d'une puce de communicationsans fil Wifi.

1.1.3 Modélisation Mathématique

Dans le cadre de ce projet, un modèle complet de la dynamique d'un robot équilibriste a été développé. Ce modèle tente de capturer les relations exactes entre les signaux émis par les capteurs et les états du système, comprenant l'effet des boîtes de réduction, prend en compte la dynamique des moteurs, et évite d'avoir à prendre des hypothèses irréalistes en ce qui concerne le déplacement du robot lors de l'établissement des modèles dynamiques découplés de la position linéaire et de la direction. De plus, le modèle développé tente de représenter la dynamique du système dans un modèle d'état linéaire relativement simple mais efficace. Ceci

est accompli en prenant l'hypothèse que certains phénomènes tels que l'effet des forces de réaction entre les roues et le corps du robot et la force centrifuge sont négligeables.

1.2 Commande via Internet

Internet est le meilleur vecteur pour contrôler, commander et mesurer à distance, à partir d'une simple page HTML et depuis n'importe quel terminal (ordinateur, pocket PC, iPhone, etc.), en milieu industriel comme en domotique. Cet ouvrage aborde à la fois les aspects théoriques et pratiques permettant de concevoir des montages électroniques capables de communiquer via Internet : protocoles TCP-IP et http, langages HTML et JavaScript ; montages dépendants, très simples, utilisant un PC qui se charge de toute la partie communication (cartes entrées/sorties, thermomètre, etc.).

1.2.1 Application web

Une page web permettant la commande par supervision du robot équilibriste via Internet a été réalisée. Cette page contient plusieurs éléments permettant la commande à distance du robot équilibriste ainsi que sa supervision. Ces éléments incluent une interface graphique, une simulation visuelle et une image provenant d'une caméra web. L'interface graphique permet à un opérateur de gérer la connexion au système et de générer une trajectoire voulue pour le robot. La simulation permet de visualiser l'évolution de la trajectoire générée, de la trajectoire anticipée et de celle réellement empruntée par le robot tandis que l'image provenant d'une caméra web sert de confirmation visuelle de la position actuelle du robot. Cette page web est générée par une application web exécutée sur un PC agissant comme serveur web. Cette application est développée de toute pièce dans le cadre de ce projet et est basée sur la technologie J2EE. Les délais de communication entre la page web et le serveur sont minimisés en faisant appel à la technique AJAX permettant des communications HTTP directement entre ceux-ci. De plus, le temps de réponse du serveur a été minimisé en portant une attention particulière sur l'architecture de cette application.

1.3 Commande via WIFI.

Quand on parle sur la commande via Wifi c'est-à-dire la communication sans fil indépendant avec l'internet en ce mode en tout dépend AD HOC comme un mode de fonctionnement celle-ci permet de communiquer et commander en manière plus simple et plus courante par les réseaux sans fil. La cellule de base de ce mode est IBSS (Independent Basic Set service) qui permet nous de communiqué sans l'aide d'une quelconque infrastructure

“Le protocole Wi-Fi offre des bénéfices supérieurs, sa portée est dans le rayon de 150-200m. L'application mobile peut également étendre la sécurité du système via une implémentation de l'application protégée par mot de passe”

1.3.1 Application Android

Pour ce système, nous utilisons la plate-forme Android car elle possède un marché énorme et une source ouverte. Un certain nombre d'outils de programmation différents ont été utilisés pour le développement de l'application. (ISSN:2319-7242)

- Eclipse Indigo JDK

L'application principale pour Android Smart Phone est codée dans l'environnement d'Eclipse Indigo IDE et compilée comme un exécutable Android standard qui s'exécute sur des plates-formes au-dessus du niveau API 8 (Gingerbread ou supérieur).

- Android SDK

Le kit de développement utilisé pour Eclipse Indigo IDE est le SDK ANDROID développé par Google.

Une application Android comporte deux méthodes pour créer un lien. Le premier utilise l'adresse IP du module Wi-Fi directement codé dans l'application pour les tests initiaux. Le deuxième est qu'il permet aux utilisateurs de rechercher l'appareil, ce qui devient une décision finale. Alors l'utilisateur peut sélectionner l'appareil à partir d'une liste, pour établir une connexion. Les étapes de base pour la connexion au module wifi étaient les mêmes pour les deux versions de l'application. Une fois l'adresse IP obtenue de la destination, l'utilisateur peut créer un socket avec un module Wi-Fi.

1.4 Interface graphique sous Matlabe

Une interface graphique permet de contrôler une application interactivement avec la souris, plutôt que par lancement des commandes au clavier. Une interface graphique comprend des menus, des boutons, des "ascenseurs", des cases à cocher, des listes de choix, des zones de texte. Elle permet de "cliquer" directement sur des images, des graphiques ou des objets pour modifier la valeur d'une variable, déclencher des fonctions ou simplement faire apparaître des informations lors d'un survol à la souris.

Matlabe permet d'écrire assez simplement une interface graphique pour faire une application interactive utilisable par des utilisateurs non formés à Matlabe.Le GUIDE est un constructeur d'interface graphique qui regroupe tous les outils dont le programmeur a besoin pour créer une

interface graphique de façon intuitive.Il s'ouvre, en tapant guide dans le Command Windows de MATLAB.

Conclusion

Dans ce chapitre on est présenté notre projet comme il faut faire pour un meilleur résultat que nous espérons,bien sûr on est défini le problème qui nous supposons résoudre à la fin, des problèmes nous rencontrerons c'est sûr, au niveau de programmation comme exemple et çaen raison de notre connaissance limitée, donc on est eu recours à l'Arduino à cause de son large bibliothèque de programmation au n'importe qu'elle domaine spécialement robotique, et en tant que on ne va pas réalisé notre projet y'a un problème de manque des donnés donc on est obligé d'utiliser des exemples extérieures pour l'étude comme les données qui fournies par le gyroscope et l'accéléromètre.

CHAPITRE 2

CONCEPTION DU ROBOT ÉQUILIBRISTE

Introduction

Dans cette optique, plusieurs des composantes tels que les moteurs, les boîtes à réduction, les encodeurs optiques, les batteries, les puces de pont en H ont été récupérés parmi des composantes disponibles au laboratoire.

Dans un premier temps, la conception mécanique a été concevoir à l'aide de logiciel SolidWorks de façon que le robot peut marcher sans problème d'emplacement ou fabrication et un montage solide permet d'obtenir un bon résultat du command et de control, et au même temps la conception électronique faut mise en considération tellement que ce très-important Pour s'assurer la continuité entre l'ensemble mécanique et l'électronique

2.1 Mécanique

Le design proposé pour le robot équilibriste à deux roues est simple, léger, et peu coûteux. La conception mécanique est constituée d'un corps avec 3 étages et de deux roues, chacune entraîné par un moteur à courant continu de 24 Vmuni d'une boîte de réduction ayant un tel rapport de choix. Le corps et les roues ont été conçus en utilisant un logiciel de CAO Soli Works. La conception des roues comprend une série des pièces enlevée de afin de minimiser leur poids. Les paramètres physiques du corps et des roues ont été estimés en utilisant la géométrie des pièces avec le logiciel de SolidWorks et les paramètres des moteurs, boîtes de réduction et des encodeurs optiques ont été obtenus à partir de leurs fiches techniques respectives. La structure mécanique du robot est montrée à la FIG. 2.1. Pour une vue mieux figure (6) présent un dessin plus définie

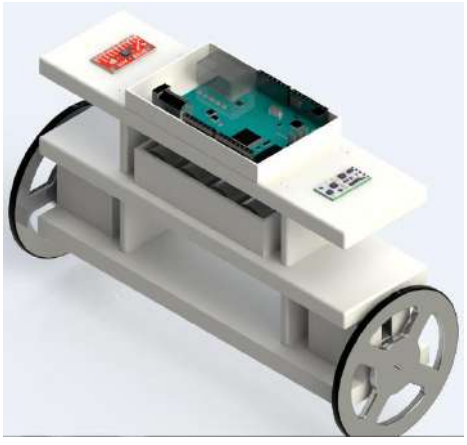


Figure 2.1 : robot équilibriste

Ce type de robot représente un système instable e présente certains défis qui seront abordés dans le reste du document. Dans un premier temps, afin de concevoir un correcteur capable de stabiliser le système, un modèle mathématique est nécessaire. Cet aspect sera examiné au chapitre 3.

2.1.1 Actionneurs

Les actionneurs utilisés pour le robot sont des moteurs à courant continu avec des boîtes de réduction. Et pour bien définie nous avons pris un exemple de ces moteurs du fabriquant Maxon Motors, donc ont un diamètre extérieur de 40 mm, une longueur de 86.2 mm, une puissance de 6 Wet une tension d'alimentation nominale de 24 V. Ils peuvent générer un couple constant allant jusqu'à 13.5 mNm, ont besoin d'un courant de 578 mApour démarrer et d'un courant maximal de 244 mA de façon continue lors de leur opération. Les moteurs sont commandés par des signaux de modulation en largeur d'impulsions générés par l'Arduino à travers un circuit de pont en H. Ces aspects seront abordés aux sections 2.2.4
La boîte de réduction utilisée est du même fabriquant. Il s'agit de têtes motrices à engrenages cylindriques ayant un facteur de réduction de 1 : 6, un rendement de 81 %et pouvant supporter un couple maximal de 100 mNm. Pour plus de détails concernant les moteurs et boîtes de réduction, le lecteur est référé aux fiches techniques (Maxon DC Motors, 2005), (Maxon Gear, 2005).

2.2 Électronique

Le système électronique conçu pour la commande du robot équilibriste dépend une plaque Arduino basé sur un microcontrôleur programmable traite l'algorithme de fonctionnement qui est déjà programmé. Deux encodeurs optiques avec une résolution de 100 impulsions par révolution sont utilisés pour mesurer les positions des roues. Un accéléromètre à deux axes et un gyroscope sont utilisés pour mesurer l'angle d'inclinaison du corps du robot et son taux de variation par rapport au temps. Deux puces de ponts en H sont utilisées pour la commande des moteurs à courant continu. Une puce de communication sans fil a été utilisée pour permettre la communication avec un ordinateur. Afin de garantir l'autonomie du robot, un ensemble de 20 piles de 1, 2 V montées en série est utilisé, ce qui donne un total de 24 V permettant d'alimenter les moteurs. La partie numérique du système est alimentée par l'intermédiaire de deux régulateurs de tension. L'architecture du système numérique est montrée à la Figure 2.2 et les principales composantes de ce système sont énumérées dans le tableau 2.1.

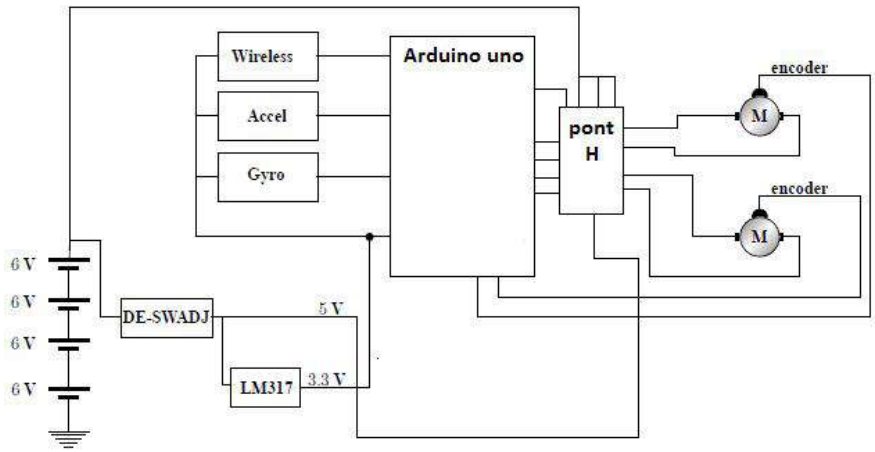


Figure 2.2 : Architecture du système numérique

2.2.1 Alimentation

L'alimentation des différentes composantes du système est réalisée par un circuit contenant un ensemble de batteries, des capacités, des résistances et des régulateurs de tension. Le schéma de ce circuit est présenté à la FIG. 2.3. Le niveau de tension de 24 V requis par les moteurs est obtenu à partir de 4 blocs de batterie de 6 V. Chacun de ces blocs de batteries est lui-même composé d'un ensemble de 5 batteries de 1.2 V. La capacité de 1000 µF est utilisée afin de stabiliser le niveau de tension lorsque les moteurs requièrent un courant considérable. Le régulateur de tension de modèle **DE-SWADJ** sert

	Description	**Fonction**
	Arduino+Wifi	Unité de calcul et de Communication
	Accéléromètre	Estimation de l'inclinaison
	Gyroscope	Mesure de vitesse d'inclinaison
	Régulateur de tension par découpage	Alimentation
	Régulateur de tension variable	Alimentation
	Puce de ponts en H	Commande de Moteur
	Moteur à courant continu	Actionneurs
	Réducteur	Transmission
	Encodeur optique	Mesure rotation moteur

Tableau 2.1 : Composantes du système

à abaisser le niveau de tension à 5 V afin d'alimenter les puces de ponts en H et les encodeurs optiques. Il s'agit d'un régulateur de tension par découpage permettant d'obtenir une sortie de tension variable. Cette composante a été sélectionnée puisque son principe de fonctionnement par découpage permet d'obtenir une meilleure efficacité énergétique en comparaison avec un régulateur de tension linéaire traditionnel.

Ce régulateur de tension est alimenté par l'ensemble des piles afin que celles-ci se déchargent de façon uniforme. Un deuxième régulateur de tension de modèle **LM317** a pour but d'abaisser le niveau de tension à 3.4 V afin d'alimenter le microcontrôleur, le gyroscope, l'accéléromètre et la puce de communication sans fil. Cette composante est un régulateur de tension linéaire variable. Celle-ci est utilisée puisqu'elle offre une bonne isolation contre les bruits afin d'éviter les interférences qui pourraient affecter le bon fonctionnement de la composante de communication sans fil. La résistance de 220 ainsi que la résistance variable de 2 Kservent à ajuster le niveau de tension obtenu à la sortie de ce régulateur de tension. La capacité de 1 µFsert à améliorer le fonctionnement de ce régulateur de tension en régime transitoire.

En générale la plaque Arduino peut générer tout ça, nous besoin juste alimenter la plaque par une alimentation de 24V et prend le 5V et 3.3V de la plaque Arduino

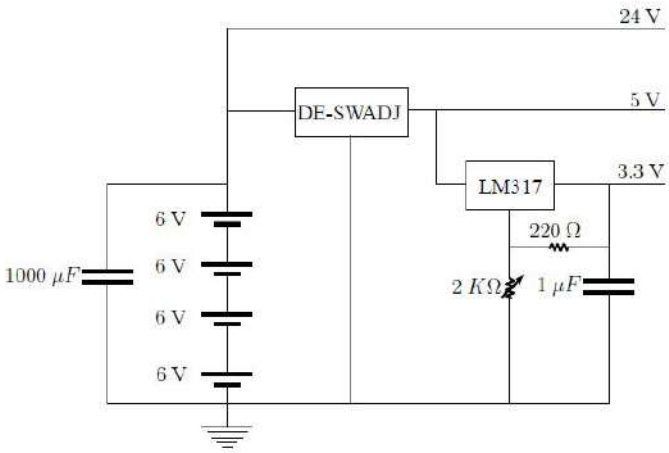


Figure 2.3 : Circuit d'alimentation

2.2.2 Capteurs

Afin de pouvoir stabiliser et commander le système en boucle fermée, on doit pouvoir mesurer certaines informations sur l'état actuel du système. Plus spécifiquement, on doit connaître l'angle d'inclinaison du robot, sa position linéaire, son angle de direction ainsi que les dérivées par rapport aux temps de ceux-ci. Afin de déterminer l'angle d'inclinaison du robot, un gyroscope ainsi qu'un accéléromètre ont été utilisés. Les signaux de ces capteurs comportent différents types de bruit et il est donc difficile d'obtenir une mesure fiable de

l'angle d'inclinaison à partir d'un seul de ceux-ci. Par contre, il est possible d'obtenir un bon estimé en les combinant. Quant à la position linéaire et l'angle de direction du robot, ceux-ci sont déterminés à partir des positions angulaires des deux roues obtenues à partir des encodeurs optiques installés sur les arbres des moteurs. Ces différents capteurs sont présentés dans les sections qui suivent.

2.2.2.1 Gyroscope

Un gyroscope est un capteur qui produit un signal proportionnel à sa vitesse de rotation. Le gyroscope utilisé sur le robot équilibriste consiste en un **IDG-300** du fabriquant *InvenSense*. Cette composante est sous la forme d'un circuit imprimé QFN ayant des dimensions de 6 x 6 x 1.5 mm et elle est représentée à la FIG. (2.4) Il s'agit d'un gyroscope analogique permettant de mesurer des vitesses de rotation autour de deux axes perpendiculaires. Dans le cas de cette application, un seul de ces canaux est utilisé, celui correspondant à l'axe parallèle à celui passant par les arbres des moteurs autour duquel le robot s'incline. Le signal analogique de ce canal est récupéré à l'aide du module de conversion analogique numérique (A/D) présenté à la section 2.2.6.3. Le gain reliant la sortie analogique à la vitesse de rotation est de $2\ mV\ s/_o$. Cette composante a également une troisième sortie analogique consistant en un signal de référence ayant une tension constante de $1.23\ V$. Ce signal de référence sera utilisé afin de calibrer les valeurs de conversion analogique numérique du signal indiquant la vitesse de rotation.

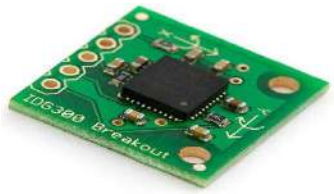


Figure 2.4 : Gyroscope **IDG-300**

L'une des faiblesses de ce type de capteur est qu'il existe une erreur ayant une moyenne non-nulle dans le signal qu'il produit dû à une sensibilité à la température ambiante. Puisque ce signal est proportionnel à la vitesse de rotation et qu'on cherche à estimer la position angulaire, ce signal doit être intégré et, à cause de cette erreur non-nulle, l'estimé dévie de la valeur réelle de position angulaire. Ce phénomène est connu sous le nom de dérive (*drift*). À cause de ceci, il est difficile d'estimer l'inclinaison du robot à partir du gyroscope uniquement. C'est pourquoi on aura aussi recours à un accéléromètre afin d'obtenir un autre estimé de l'inclinaison du robot. L'accéléromètre utilisé est présenté à la section 2.2.2.2 Il sera éventuellement possible d'obtenir un bon estimé de l'angle d'inclinaison en combinant

les estimés provenant du gyroscope et de l'accéléromètre en utilisant une technique connue sous le nom de filtres complémentaires. Cette technique est présentée en détail à la section 4.1.1.3.

2.2.2.2 Accéléromètre

Comme mentionné à la section précédente, il est nécessaire de combiner l'estimé de l'angle d'inclinaison du robot obtenu à partir du signal du gyroscope à un estimé obtenu par un autre moyen pour avoir un estimé fiable. Un type de capteur couramment utilisé dans ce contexte est un inclinomètre. Par contre, à cause de son principe de fonctionnement basé sur un pendule, ce type de capteur a une dynamique relativement lente et donc une bande passante étroite limitant la performance du système d'estimation. De façon alternative, un accéléromètre détenant une bande passante largement supérieure à celle d'un inclinomètre peut être utilisé afin d'obtenir une mesure de l'angle d'inclinaison. Cette dernière alternative a été choisie dans la conception du système de mesure du robot équilibriste en raison de la meilleure performance offerte par un tel capteur et étant donné que ce type de capteur est également moins coûteux.



Figure 2.5 : accéléromètre **LIS3LV02DQ**

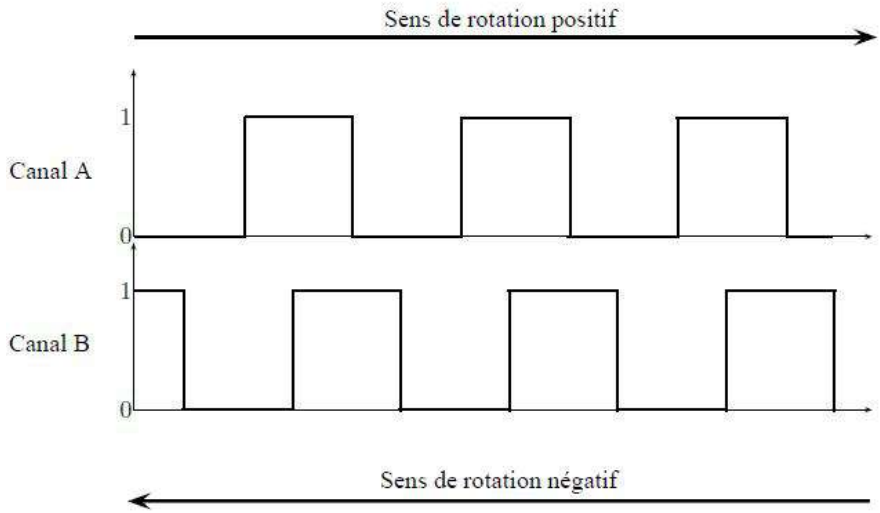
À la base, un accéléromètre est un capteur produisant un signal proportionnel à son accélération. Par contre, celui-ci ne détecte pas uniquement les accélérations mais aussi la force de la gravité. Par conséquent, il est possible d'obtenir un estimé de l'angle d'inclinaison du robot en utilisant un calcul de trigonométrie inverse sur les forces détectées dans les directions des axes perpendiculaires à l'axe autour duquel le robot s'incline.

L'accéléromètre utilisé sur le robot équilibriste consiste en un **LIS3LV02DQ**par exempledu fabriquant *STMicroelectronics*. Cette composante se présente sous la forme d'un circuit imprimé QFPN-28 ayant des dimensions de 7 x 7 x 1.8 mm et est démontrée à la FIG. (2.5). Dans le cas de cette application, on n'utilise que deux des canaux, étant ceux correspondant aux axes xet ztel que représentés à la FIG. (3.1) Ce sont dans ces directions que le vecteur de force de la gravité agit et il est donc possible d'obtenir un estimé de l'angle d'inclinaison à l'aide d'un calcul de trigonométrie inverse tel que décrit à la section (4.1.1.2)

L'une des faiblesses de ce type de capteur dans le contexte d'utilisation actuel est le fait que celui-ci ne détecte pas uniquement la force de la gravité mais aussi celles dues aux vibrations et aux mouvements. Donc, cet estimé n'est exact que lorsque le robot ne bouge pas. De façon générale, celui-ci contient également du bruit dû aux vibrations. Pour cette raison, il est difficile d'obtenir un bon estimé de l'angle d'inclinaison à partir des signaux de l'accéléromètre uniquement. C'est pourquoi il est nécessaire de combiner l'estimé de l'accéléromètre à celui du gyroscope à l'aide de filtres complémentaires, tel que décrit à la section (4.1.1.3)

2.2.2.3 Encodeurs Optiques

Un encodeur optique est un capteur permettant de déduire la position actuelle de l'arbre d'un moteur. Celui-ci est composé de deux sources lumineuses, de deux capteurs optiques correspondants et d'une roue ayant des fentes laissant passer la lumière. À sa sortie, l'encodeur produit des signaux sur deux canaux, le canal A et le canal B, correspondant aux deux paires de source lumineuse/capteur. Ces paires de source lumineuse/capteur sont espacées de façon à ce que les signaux produits aient une différence de phase de 90o, ce qui permet de déduire le sens de rotation du moteur. Une impulsion est produite dans l'un des canaux à chaque fois qu'une fente laisse passer de la lumière d'une source lumineuse à son capteur correspondant. La précision de ce type de capteur est en termes d'impulsions par révolution, ou, autrement dit, par le nombre de fentes de sa roue. Un exemple de signaux produits par un encodeur optique est présenté à la FIG. 2.6. On peut voir que lorsque le moteur tourne dans le sens positif, lors d'un front montant du signal du canal A, le signal du canal B est à l'état bas. Tandis que lorsque le moteur tourne dans le sens négatif c'est l'inverse qui se produit - lors d'un front montant du signal du canal A, le signal du canal B est à l'état haut. C'est cette propriété qui est utilisée afin de déterminer le sens de rotation d'un moteur.



Firgure 2.6 : signaux produits par un encodeur optique

Deux méthodes peuvent être utilisées afin de décoder les signaux d'un encodeur optique. La première considère seulement les fronts montants du signal de l'un des deux canaux. Ceci nous procure la résolution nominale de l'encodeur. La deuxième méthode considère les fronts montants et descendants des signaux des deux canaux ce qui a pour effet de quadrupler la résolution. Cette deuxième méthode a été utilisée dans le programme exécuté par le microcontrôleur du robot en utilisant les interruptions sur changements qui seront expliquées en détail à la section 2.2.6.2. Les encodeurs optiques utilisés sur le robot équilibriste son des **Enc 22**de la compagnie *Maxon*. Ils ont une résolution de 100 impulsions par révolution du moteur. Ceci donne une résolution de 600 impulsions par révolution d'une roue avec la première méthode de décodage, et 2400 impulsions par révolution d'une roue avec la deuxième méthode.

2.2.3Pont H

Puisque le microcontrôleur ne peut générer des signaux de puissance capables d'alimenter les moteurs directement, un circuit d'interface de puissance est requis. Le type d'interface choisi consiste en une composante de circuit intégré de format PDIP de 16 pattes comportant deux ponts en H portant le numéro de pièce **L293D**figure (2.7). Chaque pont en H consiste en deux entrées et deux sorties et est capable de commander un moteur dans les deux sens de rotation. Les différents modes de fonctionnement d'un pont en H sont présentés dans le tableau 2.2 suivant :

IN 1	**IN 2**	**Mode**

HIGH	LOW	Direction 1
LOW	HIGH	Direction 2
LOW	LOW	Descente
HIGH	HIGH	Arrête

Tableau 2.2 : fonctionnement d'un moteur à partir de L293D

Puisque les moteurs ne fonctionnent pas à plein régime en tout temps, ceux-ci seront commandés par des signaux de modulation en largeur d'impulsion envoyés aux entrées des ponts en H de la composante d'interface de puissance. Le module du microcontrôleur permettant de générer de tels signaux sera présenté en détail à la section (2.2.6.5).

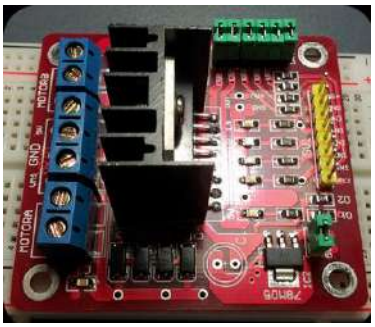


Figure 2.7 : pont H **L293D**

2.2.4 Communication sans fil

La norme la plus utilisée actuellement pour les réseaux sans fil est la norme IEEE 802.11, mieux connue sous le nom de Wi-Fi.

Le rayonnement géographique des ondes est relativement limité étant donné la faible puissance d'émission des solutions matérielles actuelles. Pour cette raison, les réseaux sans fil se sont avant tout développés comme réseaux internes, propres à un bâtiment, soit comme réseau d'entreprise, soit comme réseau domestique, souvent sous forme de réseau peer-to-peer (« pair à pair »), c'est-à-dire un réseau non centralisé, où chaque poste est autonome.

De cette façon notre projet été conçue pour communiquer, commandé ou recevait utilisant wifi model **ESP8266** figure 2.8 qui dépend dur un architecture de UE,TX/RXÉmetteur-récepteur Wi-Fiprésenté à en section 5.1.1

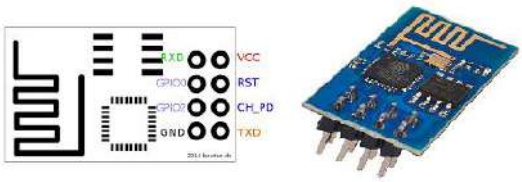


Figure 2.8 : WIFI module **ESP8266**

2.2.5 Arduino

Un module Arduino est généralement construit autour d'un microcontrôleur Atmel AVR (ATmega328, ATmega32u4) par exemple et de composants complémentaires qui facilitent la programmation et l'interfaçage avec d'autres circuits. Chaque module possède au moins un régulateur linéaire 5 V et un oscillateur à quartz 16 MHz (ou un résonateur céramique dans certains modèles).Le microcontrôleur est préprogrammé avec un bootloader de façon à ce qu'un programmateur dédié ne soit pas nécessaire.L'Arduino utilise la plupart des entrées/sorties du microcontrôleur pour l'interfaçage avec les autres circuits. Le modèle Décimal par exemple, possède 14 entrées/sorties numériques, dont 6 peuvent produire des signaux PWM, et 6 entrées analogiques. Les connexions sont établies au travers de connecteurs femelles HE14 situés sur le dessus de la carte, les modules d'extension venant s'empiler sur l'Arduino.

Les différentes versions des Arduino fonctionnent sous le même principe général figure (2.9):
A : ce sont les broches dites numériques (0 ou 1) ou « tout ou rien » ; elles offrent en sortie du 5 V et acceptent en entrée du 5 V sur le même principe.
FonctionsdigitalWrite () et digitalRead()et pour les ports PWM analogWrite().
B : ce sont les broches dites analogiques, valeur entre 0 V et 5 V
FonctionanalogRead()
C : les différentes broches d'alimentation :
Rouge : sortie 5 V (+)
Orange : sortie 3,3 V (+)
Noire : les masses (−)
Jaune : entrée reliée à l'alimentation (7 V-12 V)
Il y a des variations entre les différentes cartes (par exemple : UNO, la patte 13 est équipée d'une résistance).

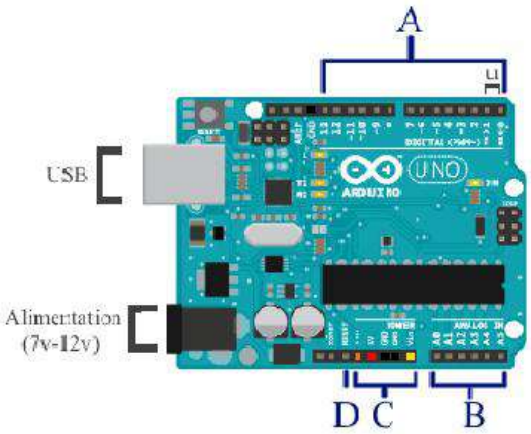


Figure 2.9 : plaque Adruino UNO

2.2.5.1 Principe de fonctionnement des interruptions

On a vu dans « La programmation, qu'est-ce que c'est » que l'Arduino exécutait son programme en séquence, instruction par instruction, dans l'ordre. Certaines instructions permettent de rompre cet ordre : les instructions conditionnelles comme le « if ... else » ou les instructions de boucle comme le « for(...) ». Mais l'Arduino reste sur ses rails et respecte les instructions qui lui sont données.

Une interruption, comme son nom l'indique, consiste à interrompre momentanément le programme que l'Arduino exécute pour qu'il effectue un autre travail. Quand cet autre travail est terminé, l'Arduino retourne à l'exécution du programme et reprend à l'endroit exact où il l'avait laissé.

Cet autre travail s'appelle le programme d'interruption ou la routine d'interruption ou encore une ISR pour « Interrupt Service Routine » en anglais.

Il est très important que les ISR aient un temps d'exécution le plus court possible. On ne fera donc aucun calcul compliqué et aucun appel à des fonctions longues comme un affichage sur un écran LCD. Il existe plusieurs sources d'interruption, on dit également vecteur d'interruption, sur l'AVR 328 qui équipe l'Arduino Uno, 26 au total. Nous n'allons pas toutes les passer en revue.

Nous allons seulement nous concentrer sur celles que nous pouvons utiliser dans nos sketchs. Il s'agit des 5 interruptions liées au changement de la tension présente sur une broche numérique qui sont donc ce que l'on appelle des interruptions externes. Les interruptions liées au timers, des compteurs internes à l'Arduino qui permettent de compter le temps de manière précise sont également intéressants et feront l'objet d'un autre article.

Une ISR n'est pas interrompue par une nouvelle interruption. La nouvelle interruption ne sera prise en compte que lorsque l'ISR en cours se terminera.

Le corollaire est qu'il ne faut pas appeler de fonctions qui se mettent en attente d'une interruption à partir d'une ISR. Comme l'interruption attendue ne peut pas déclencher une nouvelle ISR, la fonction attendra indéfiniment et tout le système se bloquera. C'est ce que l'on appelle un deadlock.

Les fonctions de Serial qui permettent d'afficher, via la ligne série et l'USB dans le moniteur série font exactement cela. Leur appel à partir d'une ISR est donc interdit.

Les Interruptions ont chacune une priorité. Par exemple, les interruptions externes sont plus prioritaires que les interruptions des Timers. L'Arduino exécutera les ISR dans leur ordre de priorité. L'ordre de priorité est donné dans la table ci-dessous. La source d'interruption située la plus en haut de la table est la plus prioritaire.

2.2.5.3 Temporisateurs

Un timer est un registre à l'intérieur du microcontrôleur qui s'incrémente (ou se décrémente) chaque fois qu'il reçoit une impulsion d'un signal d'horloge. Ce signal d'horloge peut être propre au microcontrôleur ou bien extérieur à celui-ci. Un timer est donc un compteur capable de compter le temps qui s'écoule, d'où son nom anglais de timer conter. Dans ce qui suit, le timer est toujours employé dans un mode où il s'incrémente, mais dans d'autres modes, il peut se décrémenter. Si le registre du timer comporte 8 bits, il est alors capable de compter de 0 à 255 (en hexadécimal, de 00 à FF). Lorsqu'il arrive à 255 (FF), un coup d'horloge supplémentaire devrait le faire passer à 256 (soit 100 en hexadécimal), ce qui n'est pas possible puisque le registre n'a que 8 bits.
Le module Arduino Uno est construit autour du microcontrôleur AVR ATmega328P d'Atmel qui possède 3 timers :

Le timer0 : sur 8 bits, utilisé par les fonctions delay(), millis() et micros(). Il commande également des PWM (Pulse Width Modulation ou Modulation par Largeur d'Impulsion) sur les broches 5 et 6.

Le timer1 : sur 16 bits, qui compte de 0 à 65535 (0 à FFFF en hexadécimal) et qui est utilisé par la bibliothèque Servo ou bien pour de la PWM sur les broches 9 et 10.
Le timer2 : sur 8 bits, qui est utilisé par la fonction Tone() ou bien pour de la PWM sur les broches 3 et 11.

2.2.5.4 Convertisseur analogique numérique (ADC)

Un signal numérique est un signal pouvant prendre deux états binaires : haut (HIGH) ou bas (LOW).
Un signal analogique est un signal pouvant prendre une infinité de valeurs intermédiaires.
Dans les faits, un signal numérique est un signal analogique, ne pouvant prendre que deux valeurs. On notera aussi que toutes les grandeurs physiques présentent dans la nature sont des grandeurs analogiquesEn électronique numérique, on travaille avec des bits et des octets. En analogique, on travaille avec des grandeurs physiques : tension, courant, résistance, fréquence, etc.
Pour pouvoir exploiter des mesures analogiques avec un microcontrôleur, il faut convertir la mesure analogique en une grandeur numérique. C'est justement le but des convertisseurs analogique / numérique.
Sans entrer dans les détails techniques, certes très intéressants, mais aussi très compliqués, un convertisseur analogique / numérique permet de mesurer une tension (valeur analogique) et de représenter cette tension au moyen d'une valeur numérique. Figure 2.10

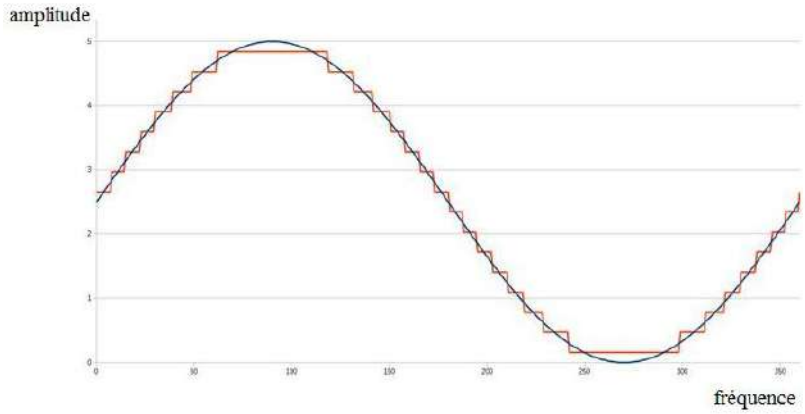


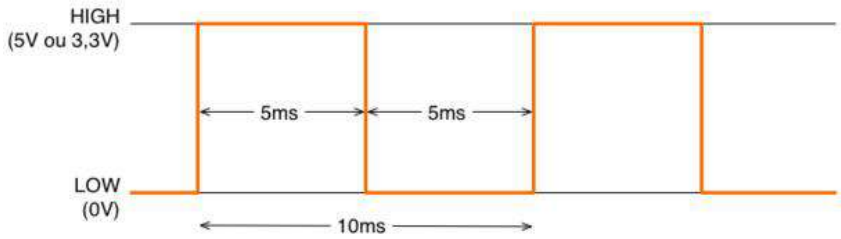
Figure 2.10 : analogique / numérique

L'idée est simple : associer une valeur numérique (un nombre entier) pour chaque valeur analogique d'une plage de tension bien précise.Un convertisseur analogique-numérique

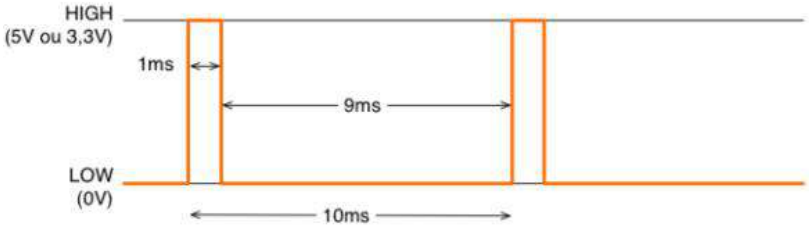
(CAN, parfois convertisseur A/N, ou en anglais ADC pour Analog to Digital Converter ou plus simplement A/D) est un montage électronique dont la fonction est de traduire une grandeur analogique en une valeur numérique (codée sur plusieurs bits), proportionnelle au rapport entre la grandeur analogique d'entrée et la valeur maximale du signal.Mesurer une tension est la base pour lire la sortie d'un capteur analogique, quel qu'il soit. C'est donc sans grande surprise que les microcontrôleurs modernes, comme ceux utilisés dans les cartes Arduino, disposent d'un convertisseur analogique / numérique intégré.

2.2.5.5 Modulation de largeur d'impulsion (PWM)

On reste en numérique, les signaux ont toujours une valeur LOW ou HIGH et le principe est de construire un signal qui est alternativement LOW et HIGH et de répéter très vite cette alternance. La DEL est donc alternativement allumée et éteinte mais le cycle est tellement rapide que la persistance rétinienne nous donne l'illusion d'une DEL allumée en permanence. Prenons par exemple une période de 10ms, soit une fréquence de 100Hz. Si la DEL est allumée pendant 5ms et éteinte pendant 5ms, comme sur la figure ci-dessous, l'impression sera une luminosité de 50% de la luminosité maximum.



Si la DEL est allumée pendant 1ms et éteinte pendant 9ms, l'impression sera une luminosité de 10% comme sur la figure ci-dessous.



Le pourcentage de temps passé à l'état HIGH sur la période du signal est appelé le rapport cyclique. Il varie donc de 0%, le signal est tout le temps LOW, à 100%, le signal est tout le temps HIGH.

Conclusion

La conception du rebot équilibriste et probablement la plus simple quand on parle sur les robots des roues en coté mécanique comme les petits (voitures, trains …. etc.).Mais elle est compliqué au niveau de l'électronique cela demande un bagage spécial pour mettre le en équilibre, malgré ça le mouvement qui nous prenons et bien en plusieurs coté et domaine comme un exemple le changement de direction il est très rapide, le poids est léger ce qui signifie minimum des frictionset un mouvement flexible permet nous de prendre n'importe chemin pour suivre, alors en ce projet on esttout dépendant en la plaque Arduino pour faire traiter et programmer le système mécatronique,à partir de prendre les données nécessaire de les capteurs à traiter et puis commander les actionneurs, qu'il va jouer sur l'équilibre du robot par avancement ou bien reculement dépend sur l'inclinaison du robot,tout ça on va présenterdans la prochaine chapitre.

CHAPITRE 3

MODÉLISATION MATHÉMATIQUE DU ROBOT

Introduction

Afin de pouvoir concevoir un correcteur capable de stabiliser le robot et lui faire poursuivre une trajectoire, il est nécessaire dans un premier temps de mettre au point un modèle mathématique qui représente fidèlement son comportement. La dynamique du robot sera représentée par deux modèles découplés, le premier décrivant la dynamique d'inclinaison et de déplacement linéaire, et le deuxième décrivant la dynamique d'angle de direction. La dynamique globale du robot a été découplée afin de pouvoir concevoir deux contrôleurs séparément ; le premier pour l'angle d'inclinaison et le déplacement linéaire et le deuxième pour l'angle de direction. De cette façon, il est possible d'assigner différentes performances aux deux sous-systèmes et éviter un couplage des dynamiques par le contrôleur.

Les valeurs des paramètres du moteur et de la boîte de réduction ont été obtenues à partir de leurs fiches techniques tandis que les valeurs des moments d'inertie du robot faut estimées à partir du modèle de conception mécanique assisté par ordinateur dans le logiciel SolidWorks et les poids des différentes composantes été obtenus en les pesants avec une balance électronique. Toutes les valeurs des paramètres sont présentées. Pour le développement du modèle mathématique représentant la dynamique du robot, certaines hypothèses doivent être faites. De façon générale, on considère que le robot se maintient autour de la position verticale, que ses roues restent en contact avec le sol en tout temps et qu'il se déplace à basse vitesse. Les forces de réactions entre le corps du robot et les roues ainsi que la force centrifuge due au mouvement d'inclinaison du robot sont également négligées. On considère que les moteurs appliquent des couples aux roues et sur le corps du robot simultanément et que les roues appliquent à leur tour des forces sur le sol provoquant une accélération linéaire du robot ainsi qu'une accélération angulaire autour de l'axe vertical faisant varier l'angle d'orientation du robot. Ces hypothèses sont prises afin d'obtenir un modèle linéaire, bien qu'approximatif, relativement simple représentant bien la dynamique du système autour de la position d'équilibre. Le contrôleur conçu à la section (4.2) devra être suffisamment robuste afin d'assurer la stabilité du système malgré le fait que le modèle n'est qu'une approximation de la dynamique véritable du système.

Cette approche a été utilisée par (Ooi R., 2003), (Kadir H., 2005) et (Abdollah M., 2006)et (LOZEAU Mathieu)

3.1 Inclinaison et déplacement linéaire

Tout d'abord, un modèle pour la dynamique d'inclinaison et de déplacement linéaire est mis au point en faisant l'hypothèse que le robot se déplace en ligne droite. En tenant compte de la symétrie du robot par rapport à l'axe vertical, il est possible de ne considérer que la moitié du robot avec un seul moteur, en notant que des couples de valeurs égales doivent être appliqués par chacun des deux moteurs afin de provoquer un déplacement purement linéaire du robot. Par conséquent, la même tension doit être fournie aux deux moteurs et cette tension $u_x(t)$ sera considérée comme l'entrée de ce sous-système. En se référant aux variables et paramètres définis dans la liste des notations et des symboles, on peut faire les observations suivantes :

– la masse de la moitié du robot M consiste en la somme de la moitié de la masse du corps m_b et de la masse d'une roue m_w:

$$M = m_b + m_w$$

– la constante de couple $K_t$et la constante de la force électromotrice $K_e$sont équivalentes dans le système d'unités international

– la position linéaire du robot $x(t)$peut être obtenue à partir du déplacement angulaire d'une roue $\theta(t)$et le rayon d'une roue r_w de cette façon :

$$x(t) = r_w * \theta(t) \tag{3.1}$$

– le déplacement angulaire de l'arbre du moteur$\theta_i(t)$ est relié au déplacement angulaire de l'arbre de la boîte de réduction $\theta_o(t)$ par le facteur de réduction r_g de la boîte de réduction de la façon suivante :

$$\theta_i(t) = r_g * \theta_o(t) \tag{3.2}$$

– le déplacement angulaire de l'arbre de la boîte de réduction $\theta_o(t)$ est constitué du déplacement angulaire de la roue $\theta(t)$ et de l'inclinaison du robot $\psi(t)$:

$$\theta_o(t) = \theta(t) + \psi(t) \tag{3.3}$$

– la force de friction $F(t)$ entre la roue et le sol est responsable d'engendrer une accélération linéaire du robot :

$$F(t) = M * \ddot{x}(t) \tag{3.4}$$

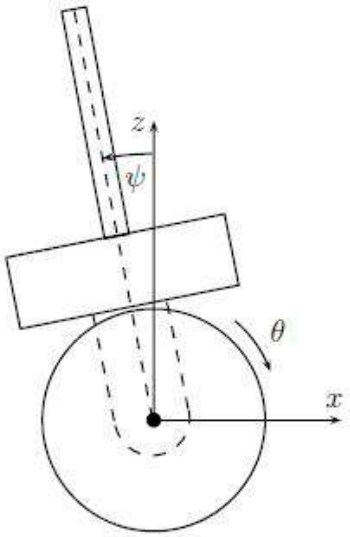


Figure 3.1 Inclinaison et déplacement linéaire du robot

3.1.1 Dynamique d'un moteur

Le couple généré par un moteur est proportionnel à son courant :

$$T_i(t) = K_t * i(t) \tag{3.5}$$

En faisant l'hypothèse que l'inductance du moteur est négligeable, il est possible d'exprimer le courant en fonction de la tension d'entrée et la force électromotrice en utilisant la loi d'Ohm :

$$i(t) = \frac{u_x(t)}{r_a} - \frac{K_e \dot{\theta}_i(t)}{r_a} \tag{3.6}$$

On peut maintenant établir la relation entre le couple généré par le moteur et la tension appliquée à son entrée en combinant les équations (3.5) et (3.6) de cette façon :

$$T_i(t) = K_t \left[\frac{u_x(t)}{r_a} - \frac{K_e \dot{\theta}_i(t)}{r_a} \right] \tag{3.7}$$

3.1.2 Couple appliqué à une roue

La relation exprimant le couple appliqué à une roue par un moteur à travers la boîte de réduction est la suivante :

$$T(t) = \eta * r_g * T_i(t)$$

En remplaçant $T_i(t)$ par la relation trouvée en (3.7), on obtient la relation suivante :

$$T(t) = \frac{\eta r_g K}{r_a} u_x(t) - \frac{\eta r_g K^2}{r_a} \dot{\theta}_i(t)$$

En utilisant la relation entre $\theta_i(t)$et $\theta_o(t)$donnée par l'équation (3.2), on peut exprimer l'équation précédente en fonction de $\theta_o(t)$de cette façon :

$$T(t) = \frac{\eta r_g K u_x(t)}{r_a} - \frac{\eta r_g^2 K^2}{r_a}\dot{\theta}_o(t)$$

Puis, en considérant la dérivée de l'équation (3.3) par rapport au temps, on obtient :

$$T(t) = \frac{\eta r_g K}{r_a}u_x(t) - \frac{\eta r_g K^2}{r_a}\dot{\theta}_i(t) - \frac{\eta r_g K^2}{r_a}\dot{\psi}(t)$$

Finalement, à partir de l'équation (3.1), on obtient la relation donnant le couple appliqué à une roue $T(t)$en fonction de la tension à l'entrée d'un moteur $u_x(t)$, de la vitesse de déplacement linéaire du robot$\dot{x}(t)$ et de la dérivée par rapport au temps de son angle d'inclinaison $\dot{\psi}(t)$:

$$T(t) = \frac{\eta r_g K}{r_a}u_x(t) - \frac{\eta r_g^2 K^2}{r_a r_w}\dot{x}(t) - \frac{\eta r_g^2 K^2}{r_a}\dot{\psi}(t) \quad (3.8)$$

3.1.3 Dynamique d'inclinaison

En se référant à la figure 3.2, et en notant que le couple appliqué à une roue est également appliqué à la moitié du corps du robot, on peut faire le bilan des couples agissant sur la moitié du corps du robot comme suit :

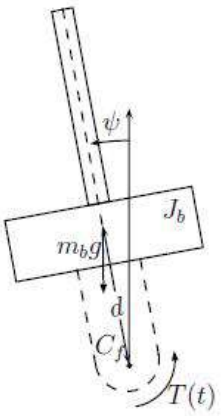


Figure 3.2 Diagramme de corps libre de la dynamique d'inclinaison

$$J_b\ddot{\psi}(t) = m_b * g_d * \sin\psi(t) + T(t) - C_f * \dot{\psi}(t) - C_f * \dot{\theta}(t)$$

En remplaçant $T(t)$par la relation trouvée en (3.8), on obtient la relation suivante :

$$J_b\ddot{\psi}(t) = m_b g_d \sin(\psi(t)) + \frac{\eta r_g K}{r_a}u_x(t) - \frac{\eta r_g^2 K^2}{r_a r_w}\dot{x}(t) - \frac{\eta r_g^2 K^2}{r_a}\dot{\psi}(t) - C_f\dot{\psi}(t) - C_f\frac{\dot{x}(t)}{r_w}$$

Ce qui donne à son tour :

$$\ddot{\psi}(t) = \frac{m_b g\, d \sin(\psi(t))}{J_b} - \left[\frac{\eta r_g^2 K^2 + C_f r_a}{r_a J_b}\right]\dot{\psi}(t) - \left[\frac{\eta r_g^2 K^2 + C_f r_a}{r_a r_w J_b}\right]\dot{x}(t) + \frac{\eta r_g K}{r_a J_b} u_x(t)$$

Par contre, cette équation contient le terme non linéaire$\sin(\psi(t))$ et nous devons la linéariser afin de pouvoir utiliser la théorie de la commande des systèmes linéaires. En considérant l'hypothèse que le robot reste autour de la verticale, $\psi(t)$peut être considéré comme étant petit, et on obtient $\sin(\psi(t)) \approx \psi(t)$, ce qui implique à son tour que :

$$\ddot{\psi}(t) = \frac{\eta r_g K}{r_a J_b} u_x(t) + \frac{m_b g d}{J_b}\psi(t) - \left[\frac{\eta r_g^2 K^2 + C_f r_a}{r_a J_b}\right]\dot{\psi}(t) - \left[\frac{\eta r_g^2 K^2 + C_f r_a}{r_a r_w J_b}\right]\dot{x}(t) \quad (3.9)$$

3.1.4 Dynamique des roues et du déplacement linéaire

En se référant à la figure 3.3, on effectue le bilan des couples appliqués à l'une des roues comme suit :

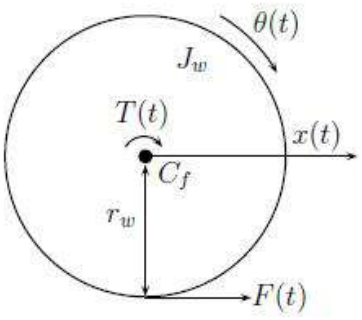


Figure 3.3 Diagramme de corps libre d'une roue

$$J_w \ddot{\theta}(t) = T(t) - F(t) r_w - C_f \dot{\theta}(t) - C_f \dot{\psi}(t) \quad (3.10)$$

La force appliquée au sol par une roue $F(t)$, peut être remplacée par une relation équivalente selon l'équation (3.4) :

$$J_w\ddot{\theta}(t) = T(t) - r_w M\ddot{x}(t) - \frac{C_f}{r_w}\dot{x}(t) \quad - C_f\dot{\theta}(t) - C_f\dot{\psi}(t)$$

En remplaçant $T(t)$ par la relation trouvée en (3.8), on obtient la relation suivante :

$$J_w\ddot{\theta}(t) = \frac{\eta r_g K}{r_a}u_x(t) - \frac{\eta r_g^2 K^2}{r_a r_w}\dot{x}(t) - \frac{\eta r_g^2 K^2}{r_a}\dot{\psi}(t) - r_w M\ddot{x}(t) - \frac{C_f}{r_w}\dot{x}(t) - C_f\dot{\psi}(t)$$

Cette équation peut s'écrire sous la forme suivante :

$$\left[\frac{J_w}{r_w} + M r_w\right]\ddot{x}(t) = \frac{\eta r_g K}{r_a}u_x(t) - \left[\frac{\eta r_g^2 K^2 + C_f r_a}{r_a}\right]\dot{\psi}(t) - \left[\frac{\eta r_g^2 K^2 + C_f r_a}{r_a r_w}\right]\dot{x}(t)$$

Ce qui donne à son tour :

$$\ddot{x}(t) = \left[\frac{\eta r_w r_g K}{r_a(J_w + M r_w^2)}\right]u_x(t) - \left[\frac{\eta r_w r_g^2 K^2 + C_f r_w r_a}{r_a(J_w + M r_w^2)}\right]\dot{\psi}(t) - \left[\frac{\eta r_g^2 K^2 + C_f r_a}{r_a(J_w + M r_w^2)}\right]\dot{x}(t) \tag{3.11}$$

3.1.5 Modèle d'état de la dynamique linéaire

En définissant le vecteur d'état $x(t)=[\psi(t) \quad \dot{\psi}(t) \quad x(t) \quad \dot{x}(t)]^T$et le vecteur de sortie $y(t) = [\psi(t) \quad x(t)]^T$ à partir des équations (3.9) et (3.11), on obtient le modèle d'état suivant :

$$\begin{cases} \dot{x}(t) = Ax(t) + Bu_x(t) \\ y(t) = Cx(t) \end{cases} \tag{3.12}$$

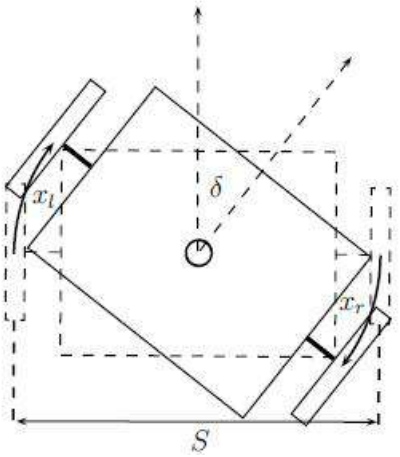
$$A = \begin{bmatrix} 0 & 1 & 0 & 0 \\ \frac{m_b g_b}{J_b} & -\frac{\eta r_g^2 K^2 + C_f r_a}{r_a J_b} & 0 & -\frac{\eta r_g^2 K^2 + C_f r_a}{r_a r_w J_b} \\ 0 & 0 & 0 & 1 \\ 0 & -\frac{\eta r_w r_g^2 K^2 + C_f r_w r_a}{r_a(J_w + M r_w^2)_)} & 0 & \frac{\eta r_g^2 K^2 + C_f r_a}{r_a(J_w + M r_w^2)} \end{bmatrix}$$

$$B = \begin{bmatrix} 0 \\ \frac{\eta r_g K}{r_a J_b} \\ 0 \\ \frac{\eta r_w r_g K}{r_a (J_w + M r_w^2)} \end{bmatrix}, \qquad C = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

En réalité, les valeurs des paramètres ne peuvent être connues avec exactitude. Aussi, celles-ci peuvent varier dans le temps à cause de certains phénomènes tels que l'usure des composantes et la décharge des batteries. De plus, le modèle a été établi sous certaines hypothèses, en négligeant certains phénomènes et a été approximé linéairement à un point de fonctionnement. Pour toutes ces raisons, le modèle contient des incertitudes et le contrôleur conçu à la section 4.2 devra être en mesure de stabiliser le système malgré leur présence.

3.2 Angle de direction

Le modèle représentant la dynamique de l'angle de direction du robot est établi en prenant en considération le fait que des couples égaux mais opposés doivent être appliqués par les deux moteurs de manière à engendrer un mouvement de rotation pure au robot sans affecter son inclinaison et sa position linéaire. Par conséquent, des tensions égales mais opposées doivent être appliquées aux deux moteurs et l'amplitude de ces tensions



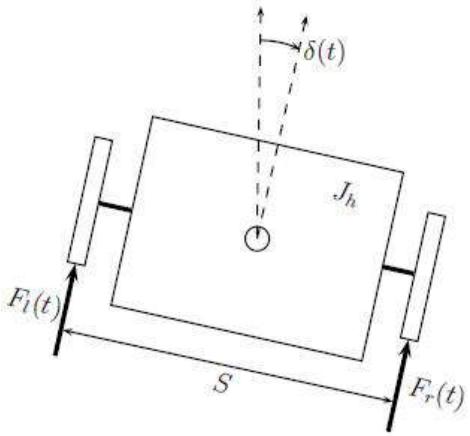


Figure 3.4 Déplacements des roues lors d'unFigure 3.5 : Diagramme de corps libre de la changement de directiondynamique d'angle de direction

u_h(t) est considérée comme étant l'entrée de ce sous-système. Ici, l'hypothèse que le robot se trouve dans le voisinage de la position verticale et que par conséquent son moment d'inertie autour de l'axe vertical peut être considéré comme une constante, est prise en considération. En se référant aux variables et aux paramètres définis dans la liste des notations et des symboles, on débute par observer le déplacement des roues produit par un changement d'angle de direction. En se référant à la figure 3.4, on peut voir que les distances parcourues par les deux roues x_r(t) et x_l(t) lorsque le robot tourne sur lui-même sont reliées à l'angle de direction δ (t) de la façon suivante :

$$x_l(t) = \delta\,(t)\frac{S}{2}$$

$$x_r(t) = -\delta\,(t)\frac{S}{2}$$

À partir de ces deux équations, on peut exprimer l'angle de direction en fonction des déplacements des roues de la façon suivante :

$$\delta\,(t) = \left[\frac{x_l(t) - x_r(t)}{S}\right] \tag{3.13}$$

Les déplacements des roues sont reliés à leurs déplacements angulaires par les équations suivantes :

$$x_r(t) = r_w\,\theta_r(t) \tag{3.14}$$

$$x_l(t) = r_w\,\theta_l(t) \tag{3.15}$$

À partir de la définition des tensions appliquées aux moteurs produisant un mouvement de changement de direction stipulant que celles-ci doivent être égales mais opposées, on peut établir la relation suivante entre la tension appliquée au moteur gauche $u_l(t)$, la tension appliquée au moteur droit $-u_r(t)$et l'entrée $u_h(t)$du sous-système qui régit la dynamique d'angle de direction :

$$u_l(t) = -u_r(t) = u_h(t)$$

La différence entre les tensions appliquées aux moteurs peut être exprimée de la façon suivante :

$$u_l(t) - u_r(t) = 2u_h(t) \tag{3.16}$$

À partir de l'équation (3.10), on peut exprimer la force F(t) appliquée par une roue sur le sol par la relation suivante :

$$F(t) = \frac{T(t) - J_W\ \ddot{\theta}(t) - C_f\dot{\theta}(t) - C_f\dot{\psi}(t)}{r_w}$$

Ensuite, à l'aide de l'équation (3.8), on peut exprimer cette dernière équation comme suit :

$$F(t) = \frac{\eta r_g K}{r_a r_w}u(t) - \left[\frac{\eta r_g^2 K^2 + C_f r_a}{r_a r_w}\right]\dot{\psi}(t)$$
$$- \left[\frac{\eta r_g^2 K^2 + C_f r_a}{r_a r_w^2}\right]\dot{x}(t) - \frac{J_w}{r_w}\ddot{\theta}(t)$$

On peut maintenant exprimer les forces appliquées au sol par chacune des roues par les équations suivantes :

$$F_l(t) = \frac{\eta r_g K}{r_a r_w}u_l(t) - \left[\frac{\eta r_g^2 K^2 + C_f r_a}{r_a r_w}\right]\dot{\psi}(t) \quad (3.17)$$
$$- \left[\frac{\eta r_g^2 K^2 + C_f r_a}{r_a r_w^2}\right]\dot{x}_l(t) - \frac{J_w}{r_w}\ddot{\theta}_l(t)$$

$$F_r(t) = \frac{\eta r_g K}{r_a r_w}u_r(t) - \left[\frac{\eta r_g^2 K^2 + C_f r_a}{r_a r_w}\right]\dot{\psi}(t) \quad (3.18)$$
$$- \left[\frac{\eta r_g^2 K^2 + C_f r_a}{r_a r_w^2}\right]\dot{x}_r(t) - \frac{J_w}{r_w}\ddot{\theta}_r(t)$$

En se référant à la FIG. 3.5, on effectue le bilan des couples agissant sur le robot autour de l'axe vertical :

$$J_d\ddot{\delta}(t) = [F_l(t) - F_r(t)]\frac{S}{2}$$

En remplaçant $F_l(t)$et $F_r(t)$par les expressions trouvées en (3.17) et (3.18), on établit la relation suivante :

$$J_d\ddot{\delta}(t) = \frac{\eta r_g KS}{2r_a r_w}[u_l(t) - u_r(t)] + \left[\frac{S\eta r_g^2 K^2 + SC_f r_a}{2r_a r_w^2}\right][\dot{x}_r(t) - \dot{x}_l(t)] + \frac{S\,J_w}{2r_w}\left[\ddot{\theta}_r(t) - \ddot{\theta}_l(t)\right]$$

À partir de l'équation (3.16), on peut exprimer cette dernière équation en fonction de l'entrée $u_h(t)$de la manière suivante :

$$J_d\,\ddot{\delta}\,(t)=\frac{\eta r_g K S}{r_a r_w}u_h(t)+\left[\frac{S\eta r_g^2K^2+SC_f r_a}{2r_a r_w^2}\right][\dot{x}_r(t)-\dot{x}_l(t)]+\frac{S\,J_w}{2r_w}\left[\ddot{\theta}_r(t)-\ddot{\theta}_l(t)\right]$$

En considérant la dérivée par rapport au temps de l'équation (3.13), on peut exprimer l'équation précédente en termes de la vitesse de changement de direction $\dot{\delta}\,(t)$ comme suit :

$$J_d\,\ddot{\delta}\,(t)=\frac{\eta r_g K S}{r_a r_w}u_h(t)+\left[\frac{S^2\eta r_g^2K^2+S^2C_f r_a}{2r_a r_w^2}\right]\dot{\delta}\,(t)+\frac{S\,J_w}{2r_w}\left[\ddot{\theta}_r(t)-\ddot{\theta}_l(t)\right]$$

Les accélérations angulaires des roues $\ddot{\theta}_l(t)$ et $\ddot{\theta}_r(t)$ peuvent premièrement être exprimées en fonction des accélérations des roues en considérant la dérivée seconde par rapport au temps des équations (3.14) et (3.15) de cette façon :

$$J_d\,\ddot{\delta}\,(t)=\frac{\eta r_g K S}{r_a r_w}u_h(t)+\left[\frac{S^2\eta r_g^2K^2+S^2C_f r_a}{2r_a r_w^2}\right]\dot{\delta}\,(t)+\frac{S\,J_w}{2r_w^2}[\ddot{x}_r(t)-\ddot{x}_l(t)]$$

Puis, ces accélérations des roues $\ddot{x}_r(t)$et $\ddot{x}_l(t)$peuvent être exprimées à leur tour en termes de l'accélération du changement de direction $\ddot{\delta}\,(t)$en considérant la dérivée seconde par rapport au temps de l'équation (3.13) de la façon suivante :

$$J_d\,\ddot{\delta}\,(t)=\frac{\eta r_g K S}{r_a r_w}u_h(t)+\left[\frac{S^2\eta r_g^2K^2+S^2C_f r_a}{2r_a r_w^2}\right]\dot{\delta}\,(t)+\frac{S^2J_w}{2r_w^2}\ddot{\delta}\,(t)$$

Ce qui donne :

$$\left[J_d+\frac{S^2J_w}{2r_w^2}\right]\ddot{\delta}\,(t)=\frac{\eta r_g K S}{r_a r_w}u_h(t)+\left[\frac{S^2\eta r_g^2K^2+S^2C_f r_a}{2r_a r_w^2}\right]\dot{\delta}\,(t)$$

Finalement, on obtient :

$$\ddot{\delta}\,(t)=\left[\frac{2\eta r_g r_w KS}{r_a(2J_d r_w^2+J_w S^2)}\right]u_h(t)-\left[\frac{\eta S^2 r_g^2K^2+S^2r_aC_f}{r_a(2J_d r_w^2+J_w S^2)}\right]\dot{\delta}\,(t) \qquad (3.19)$$

3.2.1 Modèle d'état de la dynamique de direction

En définissant le vecteur d'état $x_h(t)=\left[\dot{\delta}\,(t)\;\delta\,(t)\right]^T$ et le vecteur de sortie $y_h(t)=\delta\,(t)$, à partir de l'équation (3.19), on obtient le modèle d'état suivant :

$$\begin{cases}\dot{X}_h(t)=A_hX_h(t)+B_hu_h(t)\\ Y_h(t)=C_hX_h(t)\end{cases} \qquad (3.20)$$

$$A_h = \begin{bmatrix} 0 & 1 \\ 0 & \dfrac{\eta S^2 r_g^2 K^2 + S^2 C_f r_a}{r_a(2J_d r_w^2 + J_W S^2)} \end{bmatrix}$$

$$B_h = \begin{bmatrix} 0 \\ \dfrac{2\eta r_w r_{gKS}}{r_a(2J_d r_w^2 + J_W)} \end{bmatrix}, C_h = [1 \quad 0]$$

La dynamique du robot équilibriste est alors décrite par (3.12) et (3.20). Ces modèles seront utilisés pour la conception d'algorithmes de commande par retour d'état en utilisant la méthode de placement de pôle.

3.3 Perturbations externes

Avant de passer à la conception des algorithmes de commande et de traitement de signaux, des perturbations externes doivent être considérées et être incluses dans les modèles mathématiques de la dynamique du robot afin d'avoir recours à la théorie placement des pôles ou H_∞ pour viser à minimiser leur impact sur le comportement dynamique du robot.

Le type de perturbation considérée est des tensions additionnelles appliquées aux moteurs. On définit $w_x(t)$ comme étant une tension additionnelle appliquée aux deux moteurs affectant la dynamique linéaire du robot et w_h (t) comme étant un écart de tension additionnel appliqué aux deux moteurs affectant la dynamique de direction. Ceux-ci produisent à leur tour des couples additionnels appliqués aux roues et au corps du robot par les moteurs venant perturber le système. Lorsque le robot est commandé à distance et que le contrôleur fait en sorte que le robot suit une trajectoire envoyée par un opérateur, des perturbations au niveau de la consigne en provenance de cet opérateur, tels que des délais causés par les transferts de données via Internet, sont éventuellement perçues comme des perturbations au niveau de la commande calculée par les contrôleurs et correspondent à des perturbations au niveau des tensions appliquées aux moteurs. De plus, cette façon de faire est appropriée puisqu'il s'agit d'un type de perturbation pouvant être produite de manière précise et repliable lors de tests expérimentaux visant à valider l'efficacité et la performance du contrôleur.

Une fois que les perturbations externes ont été définies, il est maintenant nécessaire d'évaluer leur impact sur la dynamique du système.

3.3.1 Dynamique d'inclinaison et de déplacement linéaire

À partir de l'équation (3.9) et en considérant que la tension appliquée à un moteur affectant la dynamique linéaire du robot consiste maintenant en la somme de la tension provenant du contrôleur $u_x(t)$ et la perturbation externe $w_x(t)$, il est possible d'établir la relation suivante :

$$\ddot{\psi}(t) = \frac{\eta r_g K}{r_a J_b}[u_x(t) + w_x(t)] + \frac{m_b g d}{J_b}\psi(t) - \left[\frac{\eta r_g^2 K^2 + C_f r_a}{r_a J_b}\right]\dot{\psi}(t) - \left[\frac{\eta r_g^2 K^2 + C_f r_a}{r_a r_w J_b}\right]\dot{x}(t) \quad (3.21)$$

De façon similaire, pour la dynamique de déplacement linéaire, à partir de l'équation (3.11), on obtient :

$$\ddot{x}(t) = \left[\frac{\eta r_w r_g K}{r_a(J_w + Mr_w^2)}\right][u_x(t) + w_x(t)] - \left[\frac{\eta r_w r_g^2 K^2 + C_f r_w r_a}{r_a(J_w + Mr_w^2)}\right]\dot{\psi}(t) - \left[\frac{\eta r_g^2 K^2 + C_f r_a}{r_a(J_w + Mr_w^2)}\right]\dot{x}(t) \quad (3.22)$$

À partir des définitions de la section 3.1.5 et des équations (3.21) et (3.22), on peut représenter la dynamique d'inclinaison et de déplacement du robot soumis à une perturbation externe par le modèle d'état suivant :

$$\begin{cases} \dot{x}(t) = Ax(t) + Bu_x(t) + B_w w_x(t) \\ y(t) = Cx(t) \end{cases} \quad (3.23)$$

Où les matrices A, Bet Crestent inchangées etB_w= B.

3.3.2 Dynamique de direction

Pour la dynamique de direction, à partir de l'équation (3.19) et en considérant que l'écart de la tension appliquée à un moteur par rapport à la moyenne consiste maintenant en le somme de la portion calculée par le contrôleur $u_h(t)$ et la perturbation externe $w_h(t)$, on obtient la relation suivante :

$$\ddot{\delta}(t) = \frac{2\eta r_g r_w K S}{r_a(2J_d r_w^2 + J_w S^2)}[u_h(t) + w_h(t)] - \left[\frac{\eta S^2 r_g^2 K^2 + S^2 r_a C_f}{r_a(2J_d r_w^2 + J_w S^2)}\right]\dot{\delta}(t) \quad (3.24)$$

À partir des définitions de la section 3.2.1 et de l'équation (3.24), on peut représenter la dynamique de direction du robot soumis à une perturbation externe par le modèle d'état suivant :

$$\begin{cases} \dot{x}_h(t) = A_h x_h(t) + B_h u_h(t) + B_{wh} w_h(t) \\ y_h(t) = C_h X_h(t) \end{cases} \quad (3.25)$$

Où les matrices A_h, B_h et C_h restent inchangées et $B_{wh}=B_h$. Les modèles d'états représentant la dynamique du robot lorsque soumis à des perturbations externes seront utilisés lors de la conception d'un contrôleur par retour d'état.

Conclusion

Dans ce chapitre on était tout adopté sur l'approche de (LOZEAU Mathieu) qui propose une méthode utile permet nous de modéliser mathématiquement notre système en manière bien généralisé pour prendre des modèles d'état pour la dynamique linéaire et la dynamique de direction.

CHAPITRE 4

LE SYSTÈME EN TEMPS RÉEL

Introduction

Puisque le système sous étude est un système instable en boucle ouverte, il est nécessaire de concevoir un contrôleur afin de le stabiliser et éventuellement pouvoir le commander à distance. Les algorithmes de commande développés reposent sur la technique de commande par retour d'état. Cette technique permet de stabiliser le système tout en lui assignant une certaine performance pouvant être caractérisée, entre autres, en termes de réponse temporelle et de sensibilité aux perturbations externes. Plusieurs techniques existent pour déterminer les gains appropriés d'un contrôleur par retour d'état. Dans le cadre de ce projet, une méthode ont été utilisées la méthode par placement des pôles reposant sur l'assignation d'une performance temporelle. Le type de contrôleur choisi requiert d'avoir accès aux valeurs des états du système tels que définis dans les modèles mathématiques développés au chapitre 3. Il est nécessaire de développer des algorithmes permettant d'obtenir des estimés fiables de ces états à partir des signaux, des capteurs, et ce malgré la présence de bruits dans ces signaux.

4.1 Filtrage

En raison de la nature du système et de la sélection des capteurs, on n'a pas directement accès aux variables d'états. Seul les signaux $\psi_a(t)$, $\dot{\psi}_g(t)$, $\theta_{ir}(t)$ et $\theta_{il}(t)$ sont disponibles directement à partir de l'accéléromètre, du gyroscope et des encodeurs optiques respectivement. Les variables d'états doivent donc être déterminées à partir de ces signaux de sortie. Il est également important de noter que les capteurs utilisés pour mesurer l'angle d'inclinaison ont quelques inconvénients et il faut donc avoir recourt à des techniques de traitement de signaux afin d'obtenir un estimé fiable. Aussi, en raison de l'usage des boîtes de réduction, du bruit est introduit dans les mesures de la position linéaire et de l'angle de direction obtenues à partir des signaux des encodeurs optiques. Il est donc souhaitable de traiter ces signaux également avant de les utiliser pour la commande du robot en boucle fermée.

Dans le cadre de ce projet, la technique des filtres complémentaires a été choisie afin d'obtenir un estimé de l'inclinaison du robot. Tandis que des filtres passe-bas ont été utilisés afin de traiter les signaux provenant des encodeurs optiques. Les différents algorithmes utilisés afin d'obtenir de bons estimés des états sont présentés à la figure 4.1 et les divers filtres conçus sont présentés en détail dans les sections qui suivent.

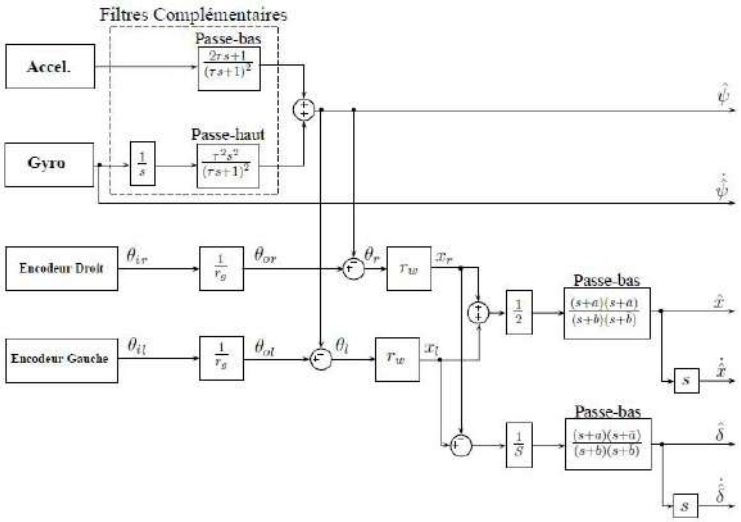


Figure 4.1 : Traitement de signaux et estimation des états

4.1.1 Angle d'inclinaison

Des estimés de l'angle d'inclinaison peuvent être obtenus de deux façons différentes à partir du gyroscope et de l'accéléromètre installés sur la plate-forme située au-dessus des moteurs du robot. Premièrement, puisque le gyroscope produit un signal proportionnel à la vitesse de rotation, comme mentionné à la section 2.2.2.1, il est possible d'intégrer ce signal afin d'obtenir un estimé de l'angle d'inclinaison. Deuxièmement, il est également possible d'obtenir un estimé à l'aide de la trigonométrie inverse à partir des signaux de l'accéléromètre afin de déterminer dans quelle direction la gravité agit sur celui-ci (voir figure 4.2).

4.1.1.1 Estimé du gyroscope

Pour le gyroscope, on a :

$$\psi_g(t) = \int \dot{\psi}_g dt \qquad (4.1)$$

Où $\dot{\psi}_g$ est le signal à la sortie du gyroscope. Malheureusement, il y a une erreur dans le signal $\dot{}_g$ du gyroscope. La moyenne de cette erreur est non-nulle en raison de la sensibilité à la température ambiante et, en conséquence, l'estimé $_g$ obtenu à l'aide du gyroscope (4.1) a une erreur qui grandit avec le temps. Cet effet est connu sous le nom de dérive. Cette dérive peut être considérée comme un bruit à basse fréquence relativement aux bruits présents dans le signal de l'accéléromètre. Un filtre passe-haut est donc nécessaire pour se débarrasser de celui-ci. La méthode utilisée afin de déterminer la fréquence de coupure appropriée pour ce filtre sera présentée à la fin de cette section.

4.1.1.2 Estimé de l'accéléromètre

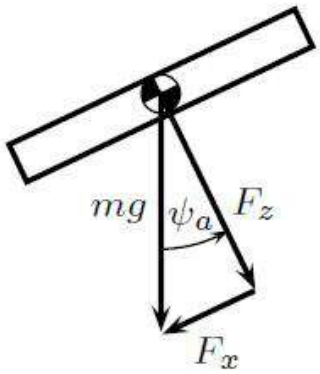


Figure 4.2 : Accéléromètre utilisé en tant qu'inclinomètre

À partir de la FIG. 4.2, on peut voir que :

$$\tan \psi_a = \frac{F_x}{F_z}, \text{ Ce qui donne } \psi_a = \tan^{-1} \frac{F_x}{F_z} \qquad (4.2)$$

Malheureusement, les signaux $F_x$et $F_z$de l'accéléromètre contiennent du bruit dû aux vibrations qui sont détectés par l'accéléromètre. De plus, l'accéléromètre ne détecte pas seulement l'effet de la gravité, mais aussi les forces d'inertie qui sont présentes lorsque le robot est en mouvement. Par conséquent, l'estimé $\psi_a$obtenu à partir de l'accéléromètre (4.2) n'est valable que lorsque le robot est statique. Tous les bruits mentionnés ci-dessus peuvent être considérés comme des bruits à haute fréquence relativement au bruit présent dans le signal du gyroscope. Un filtre passe-bas est donc nécessaire afin de les éliminer.

4.1.1.3 Filtres complémentaires

Le concept de filtres complémentaires, tel qu'expliqué par (Baerveldt&Klang, 1997), consiste à sélectionner un ensemble de filtres passe-haut et passe-bas appliqués aux signaux des estimés obtenus à partir du gyroscope et de l'accéléromètre, respectivement, nous permettant d'obtenir un bon estimé de l'angle d'inclinaison en additionnant leurs sorties. Afin d'obtenir cet effet désiré, ces filtres doivent avoir la même fréquence de coupure et la somme de leurs fonctions de transfert doit être égale à 1, ce qui signifie une bande passante théoriquement infinie :

$$G_a(s)+G_g(s) = 1 \tag{4.3}$$

Où $G_a(s)$est la fonction de transfert du filtre passe-bas utilisé sur l'estimé obtenu à partir de l'accéléromètre et $G_g(s)$est la fonction de transfert du filtre passe-haut utilisé sur l'estimé obtenu à partir du gyroscope.

En utilisant un ensemble de filtres ayant les formes suivantes :

$$G_a(s)=\frac{\overline{\psi_a}(s)}{\psi_a(s)} = \frac{2\tau s+1}{(\tau s+1)^2}$$

$$G_g(s)=\frac{\overline{\psi_g}(s)}{\psi_g(s)} = \frac{\tau^2 s^2}{(\tau s+1)^2}$$

Où τ est la constante de temps des deux filtres reliée à leur fréquence de coupure, $\overline{\psi_a}(t)$et $\overline{\psi_g}(t)$sont les estimés de l'angle d'inclinaison obtenus respectivement à partir de l'accéléromètreet du gyroscope une fois filtré. Il est clair que, peut importer la valeur de τ,les filtres sélectionnés satisfont le critère de largeur de bande passante infinie (4.3). Eneffet, on a :

$$G_a(s) + G_g(s)=\frac{2\tau s+1}{(\tau s+1)^2} + \frac{\tau^2 s^2}{(\tau s+1)^2}= \frac{2\tau s+1+\tau^2 s^2}{\tau^2 s^2+2\tau s+1}=1$$

Les diagrammes de Bode de ces filtres sont démontrés aux FIG. 4.3 et 4.4. Graphiquement et à l'aide de ISIS proteus, le diagramme de Bode de la somme de ces filtres montre que le gain est unitaire pour toutes fréquences et la phase est égale à zéro pour toutes fréquences. En conséquence, on obtient un très bon estimé de l'angle d'inclinaison$\hat{\psi}(t)$en additionnant les sorties des deux filtres $\overline{\psi_a}(t)$et $\overline{\psi_g}(t)$. Les diagrammes de Bode de chacun des filtres ainsi que celui de leur somme sont montrés sur le même graphique à la FIG. 4.5.

(En vert le gain et en rouge la phase)



Figure 4.3 : Diagramme de Bode du filtre passe-bas$G_a(s)$

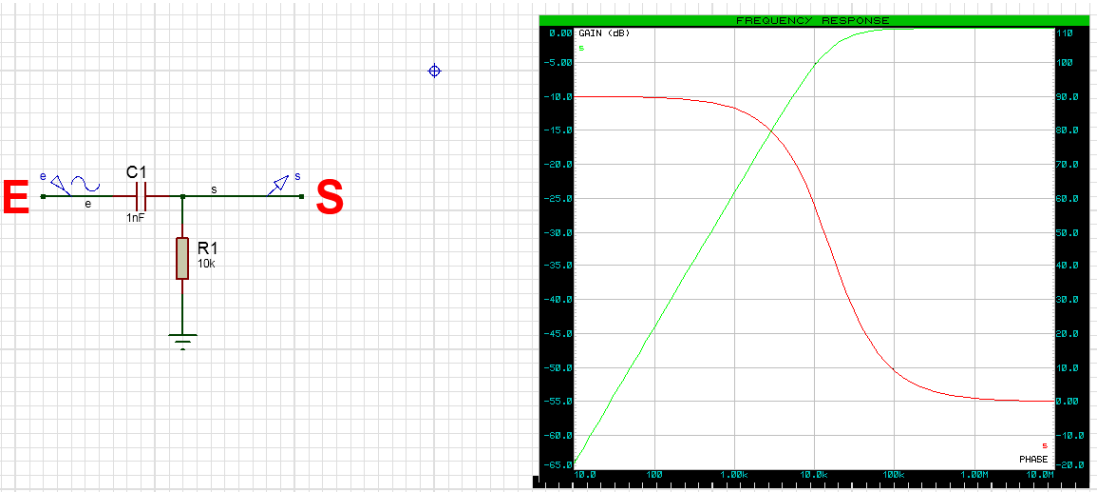


Figure 4.4 : Diagramme de Bode du filtre passe-haut $G_g(s)$

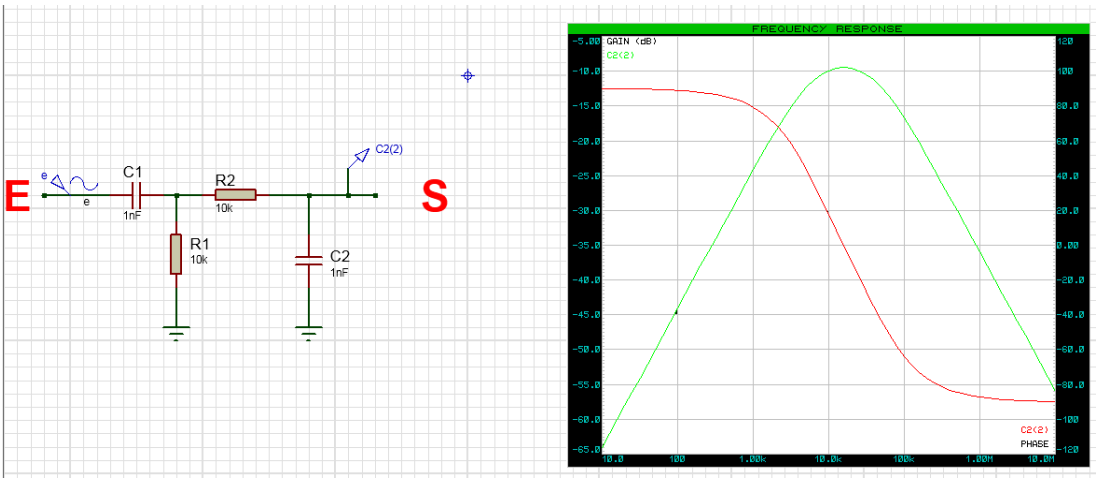


Figure 4.5 : Diagrammes de Bode des filtres complémentaires $G_a(s)G_g(s)$

Puisque l'estimé obtenu à partir de l'accéléromètre est valide uniquement lorsque le robot n'est pas en mouvement, il est préférable d'utiliser l'estimé du gyroscope autant que possible. Par conséquent, il est souhaitable d'utiliser une fréquence de coupure aussi basse que possible mais suffisamment élevée pour éliminer l'effet de dérive. En tant que telle, la constante de temps τa été déterminée expérimentalement, en commençant avec une valeur élevée et en l'abaissant jusqu'à ce que le filtre passe-haut du gyroscope élimine effectivement la dérive. La valeur de τdéterminée de cette façon et utilisée dans l'implémentation est de $0,5\ s$.
Donc, numériquement, les fonctions de transfert des filtres utilisés sont :

$$G_a(s) = \frac{s + 1}{0.25s^2 + s + 1}$$

$$G_g(s) = \frac{0.25s^2}{0.25s^2 + s + 1}$$

4.1.2 Filtres pour positions des roues

En raison de l'utilisation des boîtes de réduction, lorsqu'un moteur change de direction les dents d'engrenage perdent le contact et l'encodeur optique détecte des vitesses élevées qui ne reflètent pas la dynamique réelle du robot. Il est donc nécessaire de recourir des filtres passe-bas en vue d'éliminer des mesures de la position linéaire $x(t)$et de l'angle de direction $\delta(t)$ ce bruit de haute fréquence provenant de la position des roues $x_r(t)$et $x_l(t)$obtenus à partir des encodeurs optiques, en particulier avant de prendre des dérivées par rapport au temps.

De nombreux types de filtres passe-bas existent, mais des filtres Butterworth de second ordre ont été choisis, car ils offrent une réponse presque parfaitement plate sur la bande passante. La forme générale du filtre Butterworth de second ordre est composée de deux pôles complexes et deux zéros complexes :

$$G_x(s) = \frac{(s + a)(s + \bar{a})}{(s + b)(s + \bar{b})}$$

4.2 Correcteur

Afin de stabiliser le système et de lui faire suivre une trajectoire désirée, on doit concevoir des lois de commande appropriées. Dans cette section, un correcteur par retour d'état pour l'inclinaison et le déplacement linéaire, un correcteur par retour d'état pour l'angle de

direction et un correcteur proportionnel, plusieurs techniques permettant de calculer la valeur des gains appropriés de ces correcteurs sont proposées.

4.2.1 Références et intégrateurs

Afin d'assurer une erreur nulle en régime permanent pour la position linéaire et l'angle de direction lors d'une poursuite de trajectoire, des intégrateurs sont ajoutés. Pour ce faire, chacun des modèles d'états développés est augmenté en ajoutant un état supplémentaire consistant en l'intégrale de l'erreur en position en utilisant la procédure décrite par (Boukas, 1995).

– Pour la dynamique de la position linéaire, on définit une référence en vitesse $V_r(t)$ indiquant la vitesse voulue du robot. À partir de cette référence en vitesse, il est égalementpossible de définir la position de référence $x_r(t)$comme suit :

$$x_r(t) = \int V_r(t)\, dt$$

Ensuite, selon la référence en position $x_r(t)$, on définit l'erreur $e_x(t)$et un état additionnel $\tilde{x}_r(t)$comme suit :

$$e_x(t) = x_r(t) - x\ (t)$$

$$\tilde{x}_r(t) = \int e_x(t)\, dt$$

$$\dot{\tilde{x}}_r(t) = e_x(t)$$

La dynamique du système augmenté devient alors :

$$\begin{cases} \dot{\tilde{X}}\ (t) = \tilde{A}\tilde{X}\ (t) + \tilde{B}u_x(t) + \tilde{B}_w w_x(t) + \tilde{E}x_r(t) \\ \tilde{y}\ (t) = y(t) = \tilde{C}\tilde{X}(t) \end{cases} \tag{4.4}$$

$$\tilde{X}(t) = [X^T(t) \quad \tilde{x}\ (t)]^T,$$

$$\tilde{A} = \begin{bmatrix} A & 0 \\ -C_{xi} & 0 \end{bmatrix},$$

$$\tilde{B} = [B^T \quad 0]^T,$$

$$\tilde{B}_w = \tilde{B},$$

$$\tilde{C} = [C \quad 0],$$

$$\tilde{E} = [0 \quad 0 \quad 0 \quad 0 \quad 1]^T,$$

$$C_{xi} = [0 \quad 0 \quad 1 \quad 0],$$

On peut voir que le système est instable, mais contrôlable et observable. Par conséquent, le système peut être stabilisé par un correcteur à retour d'état. La loi de commande utilisée pour effectuer le suivi d'une vitesse et position linéaires utilise l'action intégrale et l'action directe de la référenceet prend donc la forme suivante :

$$u_x(t) = K\big(r(t) - X(t)\big) + K_{xi}\tilde{x}(t) \tag{4.5}$$

Où

$$r(t) = [0 \quad 0 \quad x_r(t) \quad V_r(t)]^T,$$
$$\text{K}=[-K_{\psi} \quad -K_{\dot{\psi}} \quad K_x \quad K_v]$$

On peut exprimer cette loi de commande de la façon suivante :

$$u_x(t) = Kr(t) - \tilde{K}\tilde{X} \quad (t)$$
$$\text{Où } \tilde{K} = [K \quad K_{xi}]$$

– Pour la dynamique de l'angle de direction, de manière semblable, on débute par introduire la référence en vitesse de rotation $w_r(t)$indiquant la vitesse de changement de direction voulue du robot. Puis, la référence en angle de direction $\delta_r(t)$peut être définie de cette façon :

$$\delta_r(t) = \int w_r(t)\, dt$$

Selon la référence en angle de direction$\delta_r(t)$, on définit l'erreur $e_h(t)$et le nouvel état $\tilde{\delta}_r(t)$ comme suit :

$$e_h(t) = \delta_r(t) - \delta \quad (t)$$
$$\tilde{\delta}_r(t) = \int \ e_h(t)\, dt$$
$$\dot{\tilde{\delta}}_r(t) = e_h(t)$$

La dynamique augmentée du système devient donc :

$$\begin{cases} \dot{\tilde{X}}_h(t) = \tilde{A}_h \tilde{X}_h(t) + \tilde{B}_h u_h(t) + \tilde{B}_{wh} w_h(t) + \tilde{E}_h \delta_r(t) \\ \tilde{y}_h(t) = y_h(t) = \tilde{C}_h \tilde{X}_h(t) \end{cases} \quad (4.6)$$

$$\tilde{X}_h(t) = [X_h^T(t) \quad \tilde{\delta}(t)]^T,$$

$$\tilde{A}_h = \begin{bmatrix} A_h & 0 \\ C_{\delta i} & 0 \end{bmatrix},$$

$$\tilde{B}_h = [B_h^T \quad 0]^T$$

$$\tilde{B}_{wh} = \tilde{B}_h,$$

$$\tilde{C}_h = [C_h \quad 0],$$

$$\tilde{E}_h = [0 \quad 0 \quad 1]^T,$$

$$C_{\delta i} = [1 \quad 0],$$

L'algorithme de commande utilisé pour le suivi de la vitesse et position angulaire de la direction utilise l'action intégrale et l'action directe de la référence et adopte alors la forme suivante :

$$u_h(t) = K_h\big(r_h(t) - X_h(t)\big) + K_{\delta i}\tilde{\delta}(t) \quad (4.7)$$

Où

$$r_h(t) = [\delta_r(t) \quad w_r(t)]^T$$

$$K_h = [K_\delta \quad K_w]$$

On peut exprimer cette loi de commande de la façon suivante :

$$u_h(t) = K_h r(t) - \widetilde{K}_h \tilde{X}_h(t)$$

Où

$$\widetilde{K}_h = [K_h \quad K_{\delta i}]$$

4.2.2 Structure du correcteur

Les tensions d'entrée $u_r(t)$et $u_l(t)$à envoyer aux deux moteurs à courant continu sont calculées en combinant les résultats obtenus à partir des correcteurs de la dynamique linéaire et de la dynamique de l'angle de direction de la façon suivante :

$$u_r(t) = u_x(t) - u_h(t)$$

$$u_l(t) = u_x(t) + u_h(t)$$

De cette façon, on a :

$$\frac{u_r(t) + u_l(t)}{2} = u_x(t)$$

$$u_l(t) - u_r(t) = 2u_h(t)$$

Tel que défini précédemment.

La structure complète du correcteur est présentée à la FIG. 4.6.

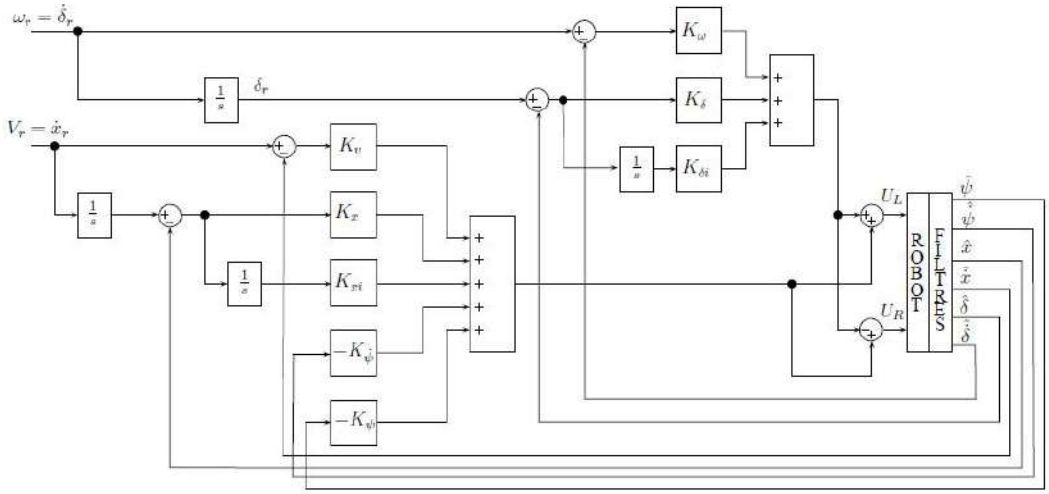


Figure 4.6 : Structure globale du correcteur par retour d'état

Et en manière simple on peut présenter comme en figure 4.7.

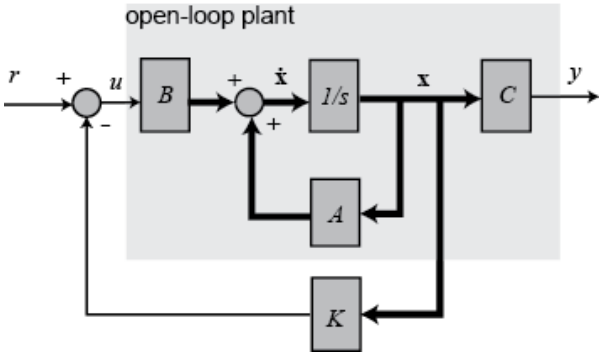


Figure 4.7 : Schéma de blocage pour le contrôleur Pole-placement

4.2.3 Placement de pôles

Maintenant qu'on a établi les lois de commandes permettant d'assurer la stabilité du système et de permettre la poursuite de référence, il est maintenant nécessaire de déterminer la valeur des différents gains du correcteur.

L'une des techniques élémentaires utilisée afin de calculer la valeur des gains $\widetilde{K}$ et $\widetilde{K}_h$est la technique par placement de pôles.

4.2.3.1 Dynamique linéaire

Le modèle d'état augmenté de la dynamique de l'angle d'inclinaison et du déplacement linéaire (4.4) est de l'ordre 5. Donc, afin de déterminer le gain $\widetilde{K}$approprié, il est nécessaire de choisir les emplacements désirés de 5 pôles. En choisissant les critères de performance suivants :

– un système stable

– un temps de réponse à 5% de 2.0s

– un dépassement de l'ordre de 5%.

Il est possible de choisir l'emplacement voulu de deux pôles dominants selon les formules suivantes :

$$D = e^{-\pi\zeta\sqrt{1-\zeta^2}} \approx 0.05 \rightarrow \zeta = \frac{\sqrt{2}}{2}$$

$$t_{s5\%} = \frac{3}{\zeta w_n} = 2s \rightarrow w_n = \frac{3}{t_{s5\%}\zeta} = 2.1213\frac{rad}{s}$$

$$p_{1,2} = -\zeta w_n \pm j w_n\sqrt{1-\zeta^2} = -1.5 \pm 1.5j$$

Les trois autres pôles doivent être dominés par ces deux pôles afin que la dynamique du système soit approximée par ces derniers. Deux de ceux-ci sont donc sélectionnés comme étant une paire de pôles complexes ayant un facteur d'amortissement ζpresque égal à 1 afin de réduire au minimum toute oscillation indésirable et étant 6 fois plus rapide que les pôles dominants. Tandis que le dernier pôle est choisi comme étant un pôle réel dont la valeur est égal à la partie réelle des deux pôles complexes dominés :

$$p_{3,4} = -9 \pm 0.0001j$$

$$p_5 = -9$$

En égalisant les pôles en boucle fermée voulus à l'équation caractéristique de la façon suivante :

$$\left|\mathrm{sI} - \tilde{A} - \tilde{B}\widetilde{K}\right| = (s - p_1)(s - p_2)(s - p_3)(s - p_4)(s - p_5)$$

$$\widetilde{K} = [K_\psi \quad K_{\dot{\psi}} \quad K_x \quad K_v \quad -K_{xi}]$$

4.2.3.2 Dynamique de direction

Le modèle d'état représentant la dynamique augmentée de l'angle de direction est d'ordre 3. On doit donc choisir l'emplacement de 3 pôles afin de déterminer le gain $\widetilde{K}_h$approprié.
En raison de l'hypothèse selon laquelle la dynamique d'angle de direction soit découplée de la dynamique de déplacement linéaire, il est souhaitable de la rendre suffisamment plus rapide que la dynamique de l'angle d'inclinaison et du déplacement linéaire.
Par conséquent, les pôles sont sélectionnés comme étant une paire de pôles dominants complexes avec un facteur d'amortissement presque égal à 1 et étant 6 fois plus rapide que les pôles dominants de la dynamique de déplacement linéaire et un pôle réel 4 fois plus rapide que ces pôles complexes dominants :

$$p_{h1,2} = -9 \pm 0.0001j$$

$$p_{h3} = -36$$

En posant l'égalité entre les pôles voulus en boucle fermée et l'équation caractéristique de la façon suivante :

$$\left|\mathrm{sI} - \tilde{A}_h - \tilde{B}_h\widetilde{K}_h\right| = (s - p_{h1})(s - p_{h2})(s - p_{h3})$$

Il est possible d'utiliser une technique de résolution appropriée pour obtenir les gains

$$\widetilde{K}_h = [K_w \quad K_\delta \quad K_{\delta i}]$$

4.3 Fréquence d'échantillonnage

Avant de discrétiser les fonctions de transfert des différents filtres, il faut d'abord choisir une fréquence d'échantillonnage appropriée. La fréquence d'échantillonnage doit être au moins deux fois supérieure à la bande passante du système selon le critère de Shannon.

Par contre, il ne faut pas qu'elle soit trop élevée afin de permettre aux calculs d'être effectués par le microcontrôleur à chaque période d'échantillonnage et de limiter la sensibilité aux bruits.

La bande passante du système est obtenue à partir des pôles du système en boucle fermée.

Lorsque le robot est stabilisé avec le contrôleur par retour d'état obtenu avec la technique de placement des pôlesnous pouvons voir que le pôle représentant la plus haute fréquence du système est situé à environ -30. Donc, afin de respecter le critère de Shannon, nous devons utiliser une fréquence d'échantillonnage d'au moins $60\ rad/s$soit environ $10\ Hz$, Selon (Franklin & Powell, 1997), il est souhaitable d'utiliser une fréquence d'échantillonnage étant au moins 30fois plus rapide que la bande passante du système en boucle fermée

La fréquence d'échantillonnage retenue est T_e = 250 Hz. Ceci représente un fréquence étant plus de 50 fois supérieure à la fréquence associée au pôle le plus rapide de la dynamique en boucle fermée du robot et il supportable par le microcontrôleur Arduino.

4.3.1 Filtre du gyroscope

Dans le cas du filtre passe-haut appliqué à l'estimé de l'angle d'inclinaison obtenu à partir du gyroscope, celui-ci est premièrement combiné à l'intégrale qui est utilisée pour obtenir cet estimé. Ainsi, on obtient :

$$G_{gi}(s) = \frac{\bar{\psi}_g(s)}{\dot{\psi}_g(s)} = \frac{1}{s}\frac{0.25\,s^2}{0.25\,s^2 + s + 1} = \frac{0.25\,s}{0.25\,s^2 + s + 1} \qquad (4.8)$$

La discrétisation de la relation (4.8) avec la transformé en Z à l'aide de Matlabe et par les commandes de Laplace et Z on obtient la relation suivante :

$$G_{gi}(z) = \frac{\bar{\psi}_g(z)}{\dot{\psi}_g(z)} = \frac{z^3 - 3.846z^2 + 2.771z}{z^3 - 2.846z^2 + 2.7351z - 0.8875}$$

Celle-ci est ensuite implémentée dans le microcontrôleur sous la forme d'équation aux différences suivante :

$$\bar{\psi}_g(k) = \dot{\psi}_g(k) - 3.846\dot{\psi}_g(k-1) + 2.771\dot{\psi}_g(k-2) + 2.846\bar{\psi}_g(k-1) - 2.7351\bar{\psi}_g(k-2) + 0.8875\bar{\psi}_g(k-3)$$

4.3.2 Filtre de l'accéléromètre

Dans le cas du filtre passe-bas appliqué à l'estimé de l'angle d'inclinaison obtenu à partir de l'accéléromètre, la fréquence d'échantillonnage utilisée est moins élevée que celle utilisée pour le reste du système puisque nous ne voulons conserver que les basses fréquences. Ceci a pour but de réduire le nombre de calculs à être effectués par le microcontrôleur.

La fréquence d'échantillonnage pour ce filtre est choisie comme étant une fraction de celle utilisée pour le reste du système et est fixée à$f_{ea} = 150\,HZ$. La fonction de transfert de ce filtre est ensuite discrétisée en utilisant la transformé en Z, ce qui donne la fonction de transfert discrète suivante :

$$G_a(z) = \frac{\bar{\psi}_a(z)}{\psi_a(z)} = \frac{z^3 - 2.01z^2 + 1.98z}{z^3 - 2.98z^2 + 2.95z - 0.97}$$

Celle-ci est ensuite implémentée dans le microcontrôleur sous la forme d'équation aux différences suivante :

$$\bar{\psi}_a(k) = \psi_a(k) - 2.01\psi_a(k-1) + 1.98\psi_a(k-2) + 2.98\bar{\psi}_a(k-1) - 2.95\bar{\psi}_a(k-2) + 0.97\bar{\psi}_a(k-3)$$

Un exemple pour la comparaison des résultats

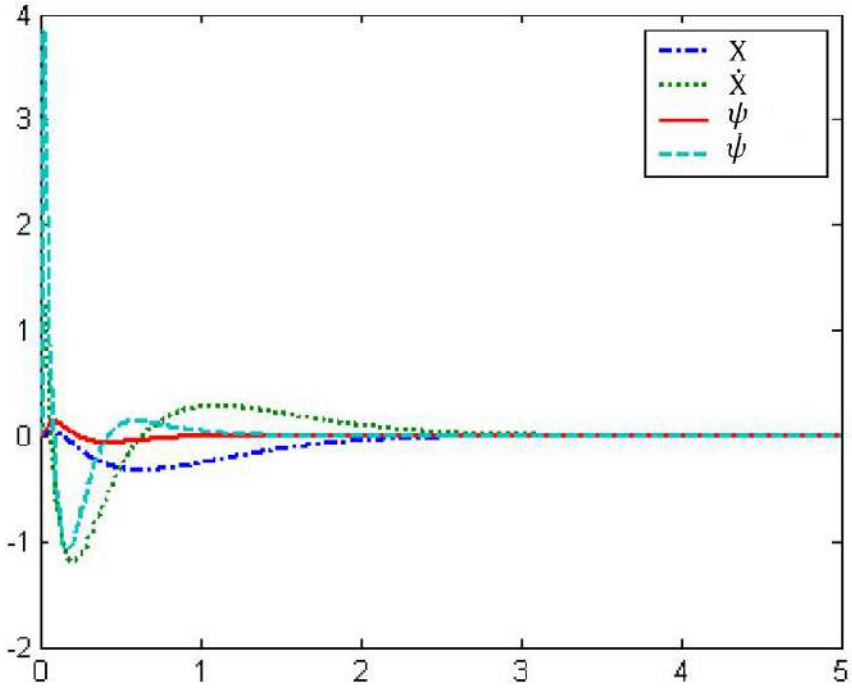


Figure 4.8 : Réponse du système avec le contrôle de placement de pôles (3)

Conclusion

Après la modélisation du chapitre 3, le chapitre 4 tenant en compte les matricesque nous avons obtenuespour appliquer les lois de commande et par l'action intégral et l'action directe le correcteura conçu, etle filtrage qui améliorer les signaux pour une meilleure estimation tout ça contribué de mettre le robot en équilibre, en fin et d'après l'expérience de (RichChi Ooi) la résulta en figure 4.8 présente la réponse indicielle de notre système avec placement de pole contrôleur

CHAPITRE 5

COMMANDE VIA WIFI

Introduction

Le lien de communication plus besoin de sécurité. WIFI protocoles fournies plus sécurité pour sécuriser la connexion.

Une application Android a deux méthodes pour crier un lien, le premier est utilisé IP adresse de module WIFI codé directement dans l'application pour le test initial, la deuxième permettant aux utilisateurs de chercher des appareils que devient la dernière décision, après les utilisateurs peut sélectionner l'appareil d'après une liste, pour faire une connexion.

Notre appareil que nous voulons connecter avec, c'est le robot équilibriste à travers le module WIFI que nous utilisons de référence ESP8266. À l'aide de l'application Android qui nous choisissons est Blynk app disponible au Play store où pouvons télécharger cette application est flexible pour n'importe qu'il veut utiliser les boutons de commande, la figure 5.1

Et pour commander les moteur à courant continue plutôt la direction du robot à l'aide de pont H de référence L293D qui nous aidons de programmer le en propose ce schéma en figure 5.5 et pour programmer ces moteur il faut suivre les conditions en section (2.2.4)

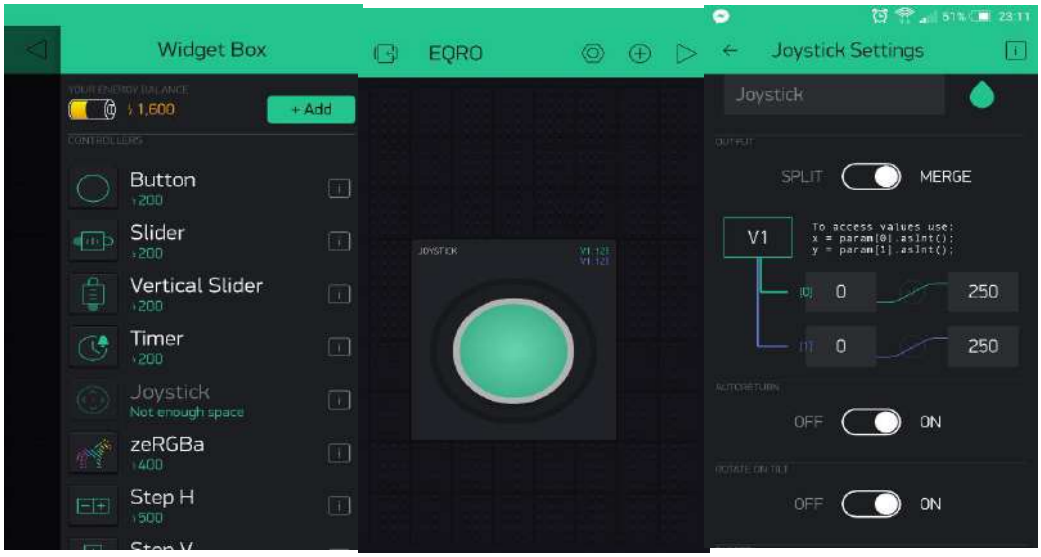


Figure 5.1 :l'interface de commande et configuration

5.1 Architecture de communication

Afin de pouvoir commander le système mécatronique présenté aux chapitres précédents à distance via wifi, une architecture de communication comportant plusieurs nœuds et différents protocoles a été mise en place. Une vue d'ensemble de cette architecture est présentée à la

Figure 5.2.Lors de la commande par supervision via wifi, l'information numérique circulant entre le robot et l'opérateur passe par plusieurs nœuds de communication. Ces différents nœuds sont présentés en détail dans les sections qui suivent.

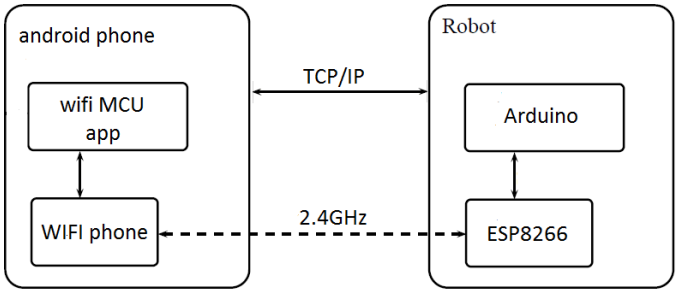


Figure 5.2 : architecture de communication

5.1.1 Émetteur-récepteur Wi-Fi

Émetteur-récepteur Wi-Fi. La plupart du Framework Wi-Fi est implémenté dans FPGA. Dans En outre, l'émetteur-récepteur est implémenté dans le même FPGA. Nous avons implémenté la D2D state machine et sa logique correspondante dans le contrôleur en temps réel. Le contrôleur est également en charge d'alimentation des données à la chaîne de traitement de transmission FPGA et la lecture des données décodées de chaîne de traitement de réception FPGA. La figure 5.4 montre la structure du cadre Wi-Fi. (ArashAsadi, mars 2016)

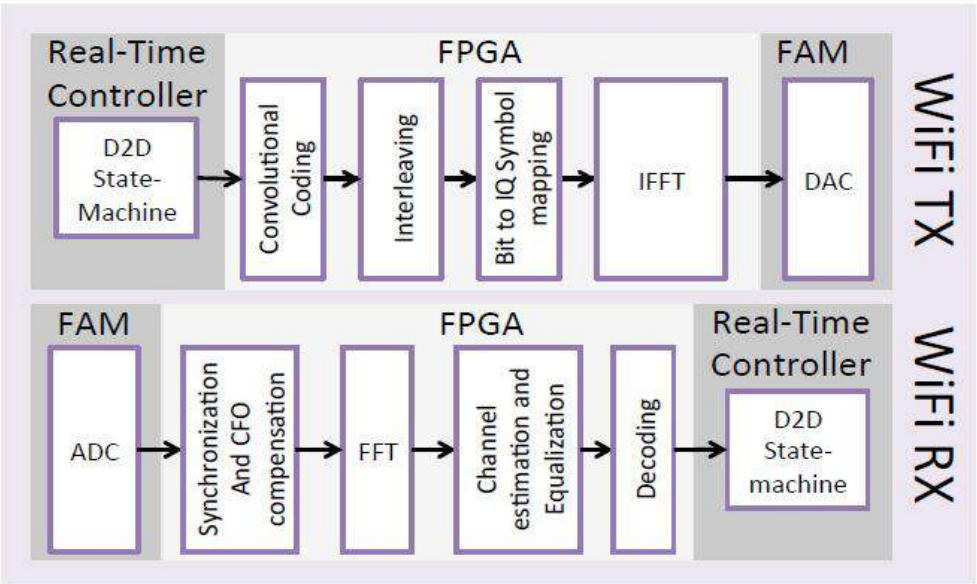


Figure 5.3: architecture d'UE WIFI interface, ré(1)

5.1.2 Robot

Bien entendu, le premier nœud est le robot équilibriste. Sa conception et son fonctionnement on était présentés dans les chapitres précédents. Du point de vue de l'architecture de communication, celui-ci est en lien avec l'appareil mobile par l'intermédiaire d'une liaison sans fil assurée par la composante de communication sans fil **SPE8266**.

5.1.2.1 Protocol TCP/IP

Les données échangées entre le robot et l'appareil téléphone sont acheminées à l'aide du protocole TCP/IP, Ce qui signifie que les données doivent savoir où elles vont nécessiter une adresse IP et un transport TCP

5.1.2.2 Protocol SPI

L'interface de périphériques séquentiel (*Serial Peripheral Interface* - SPI) est un protocole de communications numériques permettant à plusieurs composantes électroniques de communiquer entre elles sur un bus. Il s'agit d'un protocole permettant des communications synchrones (dans les deux sens en même temps) ayant une topologie maître/esclave dans laquelle il existe un seul maître par système, celui-ci étant responsable de sélectionner l'esclave avec lequel communiquer et de générer un signal d'horloge pour cadencer les échanges de données.

On est utilisé ce Protocol pour faire relié l'Arduino avec le SD mémoire pour prendre des donnés de performance du robot à partir les capteursde gyroscope et l'accéléromètreun programme et déjà pré sous Arduino studio comme un exemple figure 5.4

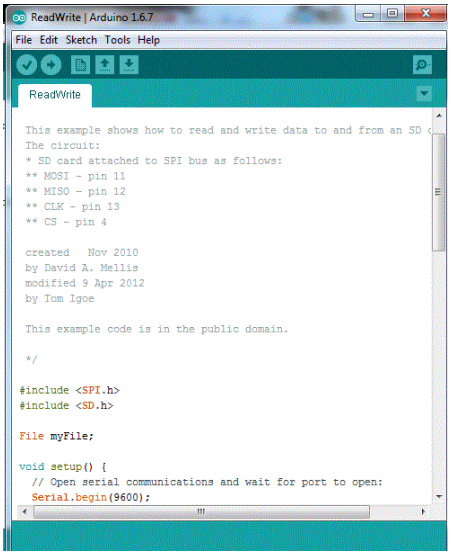


Figure 5.4 : lire /écrire depuis SD mémoire

5.2 Android App

Robot commandé par Wi-Fi à l'aide de l'application Arduino et Blynk. Ce Robot basé sur Arduino peut être contrôlé sans fil à l'aide de tout téléphone intelligent Android Wi-Fi.
Pour la démonstration de Wi-Fi Contrôlé le Robot, nous avons utilisé une application mobile Android appelée "Blynk". Blynk est une application très compatible avec Arduino, pour créer beaucoup projet de la base. Cette application peut être téléchargée depuis Google Play Store et peut être configurée facilement

5.2.1La configuration de l'applicationavecl'Arduino

Nous devrions obtenir Auth jeton dans l'application Blynk.

Char auth [] = " a9001eb535f648188d34a47033179319"

Le travail du robot commandé par Wi-Fi est très simple, il nous suffit d'avancer ou de glisser le joystick dans la direction, où nous voulons déplacer le robot. Comme si nous voulions déplacer le robot dans la direction avant, nous devons faire glisser le joystick 'cercle' dans vers l'avant direction. De même, nous pouvons déplacer le Robot dans le sens Gauche, Droite et en arrière en faisant glisser le joystick dans la direction respective. Maintenant, dès que nous libérons le Joystick, il reviendra au centre et Robot s'arrêtera.

L'application Blynk envoie des valeurs du joystick à deux axes à Arduino, via un support Wi-Fi. Arduino reçoit les valeurs, les compare avec des valeurs prédéfinies et déplace le robot en conséquence dans cette direction. Le schéma dans la figure 5.5.

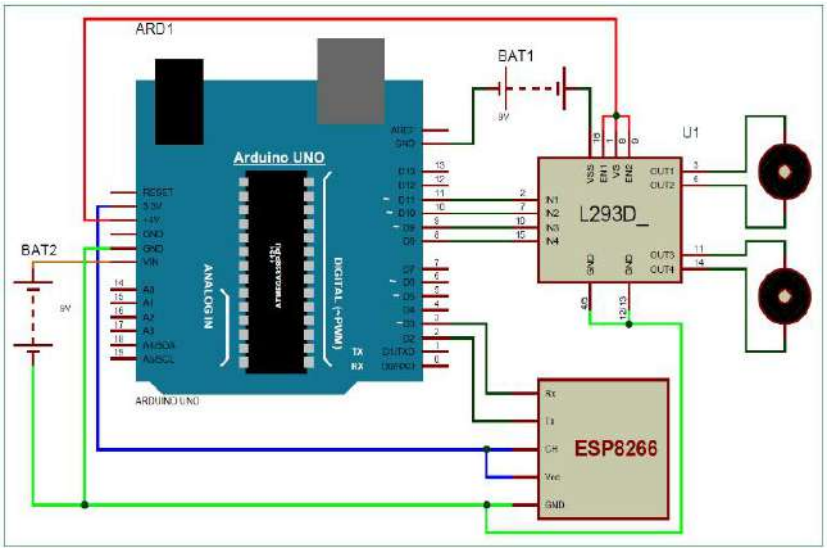


Figure 5.5 : schéma de contrôle

5.2.2Explication de programmation

Le programme est presque prêt dans Arduino IDE. Il suffit de télécharger Blynk Library pour Arduino. Et après avoir apporté quelques modifications, l'utilisateur peut créer son propre robot à commande Wi-Fi.(Saddam)

D'abord, nous avons inclus toutes les bibliothèques nécessaires pour exécuter ce code dans Arduino IDE, puis entré Auth jeton, de l'application Blynk, dans la chaîne d'auth. Nous connectons ici la broche série Wi-Fi avec Software Serial of Arduino. La broche choisie 2 comme RX et 3 comme TX.

```
#define BLYNK_PRINT Serial
#include <ESP8266_SoftSer.h>
#include <BlynkSimpleShieldEsp8266_SoftSer.h>
#include <SoftwareSerial.h>
SoftwareSerial EspSerial(2, 3); // RX, TX
ESP8266 wifi(EspSerial);
char auth[] = "a9001eb535f648188d34a47033179319";
```

Ensuite, nous avons défini les broches de sortie (8,9,10,11) pour les moteurs et avons écrit certaines fonctions de direction pour déplacer le Robot dans une direction particulière : void forward (), void vers l'arrière (), void vers la droite () et vide vers la gauche ()

Après cela, dans la fonction de configuration, nous initialisons tous les périphériques requis, comme donner direction aux broches du moteur, commencer la communication en série, fournir le nom d'utilisateur et le mot de passe Wi-Fi.

```
void setup()
{
Serial.begin(9600);
delay(10);
EspSerial.begin(9600);
delay(10);
Blynk.begin(auth, wifi, "username", "password");
pinMode(m11, OUTPUT);
pinMode(m12, OUTPUT);
pinMode(m21, OUTPUT);
pinMode(m22, OUTPUT);
```

```
}
```

Maintenant, nous avons vérifié certaines conditions de contrôle du robot. Ici, nous avons sélectionné la broche virtuelle 1 (V1) pour prendre l'entrée de Blynk App pour contrôler le robot. Comme nous avons utilisé l'option marge dans l'application, nous obtiendrons les valeurs des axes x et y à la même broche.

```
BLYNK_WRITE(V1)
{
int x = param[0].asInt();
int y = param[1].asInt();
if(y>220)
forward();
else if(y<35)
backward();
else if(x>220)
right();
else if(x<35)
left();
else
Stop();
}
```

Enfin, nous devons exécuter la fonction Blynk en boucle, pour exécuter le système.

```
void loop()
{
Blynk.run();
}
```

5.2.2.1 Le code

```
#define BLYNK_PRINT Serial
#include <ESP8266_SoftSer.h>
#include <BlynkSimpleShieldEsp8266_SoftSer.h>

 #include <SoftwareSerial.h>
 SoftwareSerial EspSerial(2, 3); // RX, TX

ESP8266 wifi(EspSerial);

char auth[] = " a9001eb535f648188d34a47033179319";

#define m11 8
#define m12 9
#define m21 10
#define m22 11

void forward()
{
  digitalWrite(m11, HIGH);
  digitalWrite(m12,LOW);
  digitalWrite(m21,HIGH);
  digitalWrite(m22,LOW);
}

void backward()
{
  digitalWrite(m11, LOW);
  digitalWrite(m12,HIGH);
  digitalWrite(m21,LOW);
  digitalWrite(m22,HIGH);
}

void right()
{
  digitalWrite(m11, HIGH);
  digitalWrite(m12,LOW);
  digitalWrite(m21,LOW);
  digitalWrite(m22,LOW);
}

void left()
{
  digitalWrite(m11, LOW);
  digitalWrite(m12,LOW);
  digitalWrite(m21,HIGH);
  digitalWrite(m22,LOW);
}

void Stop()
{
  digitalWrite(m11, LOW);
  digitalWrite(m12,LOW);
  digitalWrite(m21,LOW);
  digitalWrite(m22,LOW);
}

void setup()
{
  Serial.begin(9600);
  delay(10);
  EspSerial.begin(9600);
  delay(10);

  Blynk.begin(auth, wifi, "username", "password");
  pinMode(m11, OUTPUT);
```

```
  pinMode(m12, OUTPUT);
  pinMode(m21, OUTPUT);
  pinMode(m22, OUTPUT);
}
BLYNK_WRITE(V1)
{
  int x = param[0].asInt();
  int y = param[1].asInt();
 /*  Serial.print("X = ");
  Serial.print(x);
  Serial.print("; Y = ");
  Serial.println(y);*/
  if(y>220)
  forward();
  else if(y<35)
  backward();
  else if(x>220)
  right();
  else if(x<35)
  left();
  else
  Stop();
}
void loop()
{
  Blynk.run();
}
```

5.3 Interface graphique

L'interfacede ce robot présente la performance de mouvement par des données qui a été produit par les capteurs pour le suivi, donc dans cette section on est proposée aussi une idée qui dit que si on suive la vitesse angulaire qui provient de gyroscope par le temps on peut situer la situation du robot dans le chemin comme une dos d'âne ou un trou, donc en situation équilibre le gyroscope indique le zéro, si y'a une inclinaison il indique la vitesse angulaire négatif ou bien positif que signifier si le robot montée ou faire descendre figure .5.5, présente un exemple nous prenons d'après une application de téléphone qu'il fait la même mesure, par ce que le téléphone intelligent contient la pièce électronique gyroscope, l'application est sur le nom de (physics toolbox suite) qui nous fournissent par les données qui nous besoin de faire observer.



Figure 5.5 : diagramme de test robot montée un dos d'âne

Notre interface principale a été conçue d'après logiciel Matlabe utilisant le Guide et un petit programme de développement pour utiliser l'interface graphique qui permet nous de visualiser les données selon la carte SD et le fichier en forme Excel, figure 5.6

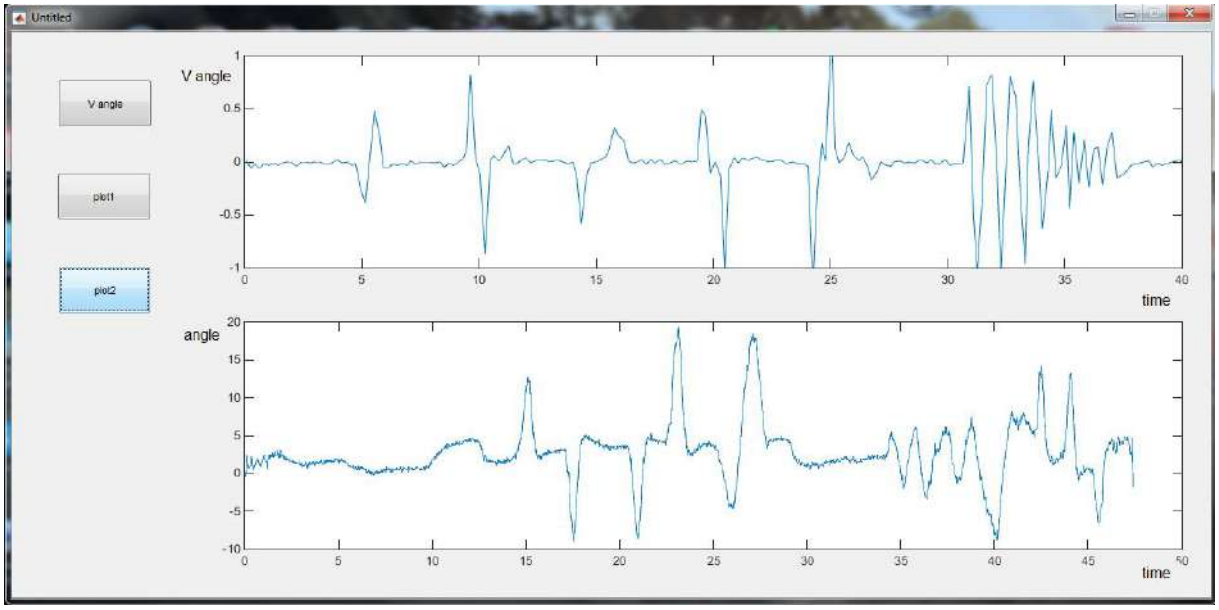


Figure 5.6 : interface graphique

5.3.1 Développement sous Matlabe (Guide)

La partie développée sous Matlabe est comme suite :

```
% --- Executes on button press in V angle.
function pushbutton1_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
global data
[file pat]=uigetfile('*.xlsx');
pf=[pat,file];
data=xlsread(pf);

% --- Executes on button press in plot1.
function pushbutton2_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
global data
time1=data(:,1);
vit=data(:,2);
plot(time1,vit);
xlim(handles.axes1,[0 40]);
ylim(handles.axes1,[-1 1]);
axes(handles.axes1);


% --- Executes on button press in plot2.
function pushbutton3_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton3 (see GCBO)

% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
global data
time2=data(:,3);
patch=data(:,4);
plot(time2,patch);
xlim(handles.axes3,[0 50]);
ylim(handles.axes3,[-20 20]);
axes(handles.axes3);
```

Conclusion

Dans ce chapitre la partie command par WI-FI été mise en place en manière simple et utile au bute deprendre tout ce qui est facile à utiliser comme les applications qui nous proposions mais on reste en besoin des expériences en temps réel pour bien définir la partie command, la programmation de la carte Arduino pour téléguider été proposer par (Saddam), utilisant un robot de 3 roues.

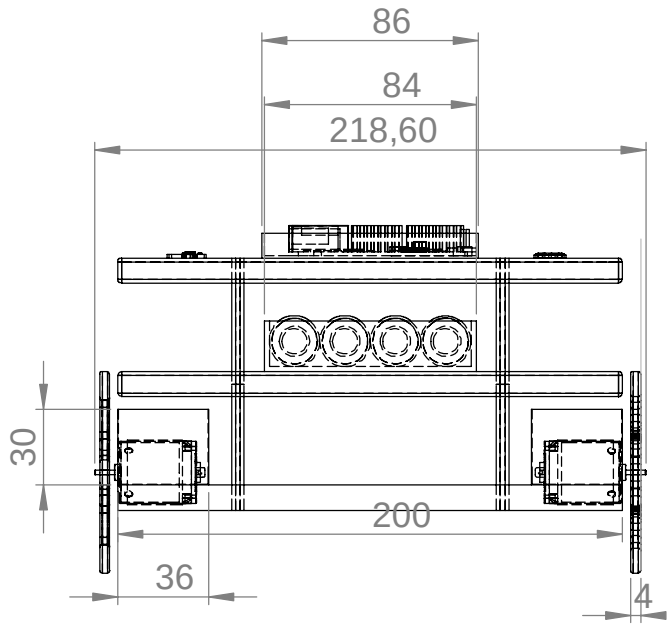
CONCLUSION GÉNÉRALE

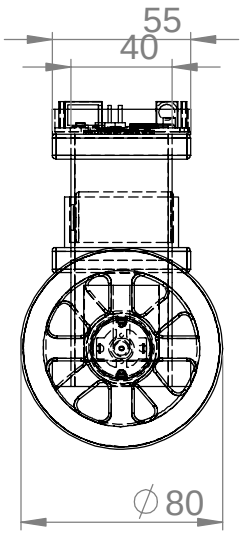
Ce projet de recherche portait sur la modélisation mathématique de notre système mécatronique et proposer un moyen de communication, commander le robot à distance,mais grâce à nos études, nous avons dépendu de quelques exemples externes pour essayerde se rapprocher les définitions et de la manière d'atteindre notre objectif c'est que le robot reste en équilibrealors qu'il se déplace,d'après le expérience que nous tenons en compte (LOZEAU Mathieu) en peut dire que l'expérience a été prouver.

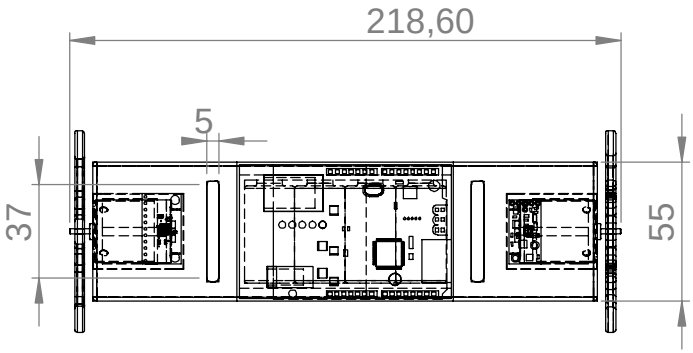
Du point de vue des algorithmes de traitement des signaux et de commande, il est possible d'envisager d'autres techniques de traitement des signaux des capteurs afin d'éliminer leurs bruits et d'obtenir un estimé fiable des états du robot, le filtre de Kalman est un ensemble d'équations mathématiques pour des données récursives optimales, algorithme de traitement qui fournit la solution de la méthode de (least squares). Il Intègre toutes les informations pouvant être fournies et traite toutes les informations disponibles mesures, indépendamment de leur précision pour estimer la valeur actuelle des variables S'intéresse à l'utilisation d'une connaissance du système et de la dynamique de mesure

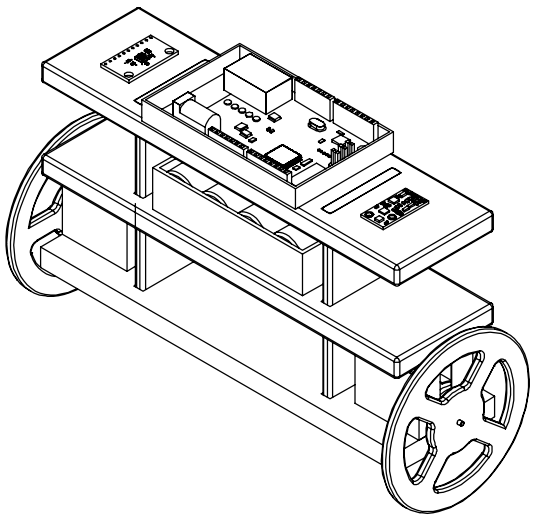
Nous avons parlé de la communication via Internet dans la section 1.2 et son site d'application dans l'intention de proposer une manière sous développée de commander à distance, et lorsque nous disons à distance cela signifie dans n'importe quel point sur terre qui al'accès au réseau, on propose comme un exemple de système de commande via Internet utilisé dans le cadre de l'enseignement de la mécatronique, on peut citer (Lindsay, Good &Halgamuge, 2000) où on présente un système permettant la commande d'un robot industriel CRS F3 à partir d'une appliquette Java intégrée dans une page web. Ce système a vu le jour à l'Université de Melbourne dans le cadre de la mise à jour ses pratiques d'enseignement de la mécatronique suite aux avancements des technologies de l'information et de l'Internet.

Et comme une idée supplémentaire on est déjà proposer pour voir et testé notre chemin d'après un capteur de vitesse angulaire la question est pourquoi, donc c'est par ce que le gyroscope peut ressentir les très petits vitessessi le robot en état équilibre et stable en mouvement sur un plan n'importe qu'elle obstacle le gyroscope nous fournit des données qui nous transformons à un graphe à l'aide de l'interface Matlabe pour observer tout ce qui se passesur le chemin, cette proposition peut utiliser au domaine génie civile pour suivre les routes en manière professionnel









Université Dr.Yahia Fares de Médéa

Dessin	Robot Auto Equilibré	**Mecatronique**
Master 2		**ECL 1:3**
11/05/2017	**Benrami youcef /Saadoudi Abdelmajid**	

RÉFÉRENCES

1. ArashAsadi.March 2016.Opportunistic Device-To-Device Communication In Cellular Networks: From Theory To Practice

2. LOZEAU Mathieu.SEPTEMBRE 2009.COMMANDE PAR SUPERVISION DE SYSTÈMES MÉCATRONIQUES VIA INTERNET

3. Rich Chi Ooi.2003.Balancing a Two-Wheeled Autonomous Robot

4. International Journal Of Engineering And Computer Science ISSN:2319-7242 Volume 3 Issue 10 October, 2014 Page No.8792-8794

5. Prof. R.S. Suryavanshi,Volume 3 Issue 10 October, 2014Page No.8792-8794,Home Automation System Using Android and WiFi

6. https://circuitdigest.com/microcontroller-projects/arduino-wifi-controlled-robot, saddam

7. https://www.arduino.cc/en/Tutorial/HomePage

8. http://www.locoduino.org/spip.php?article69

9. http://www.locoduino.org/spip.php?article64

10. http://docs.blynk.cc/

11. S. W. Nawawi, M. N. Ahmad, and J. H. S. Osman; 2008; Real-Time Control of a Two-Wheeled Inverted

12. Electronic and Communication Engineering Vol2, No:3, 2008Real-Time Control of a Two-Wheeled Inverted Pendulum Mobile Robot, S. W. Nawawi, M. N. Ahmad, and J. H. S. Osman

13. Boukas, E. K. (1995). Systèmes Asservis. Éditions de Polytechnique,Montréal, Canada.

15. Boukas, E. K. (2005). Stochastic Switching Systems : Analysis and Design. Birkhäuser

16. Maxon. (2005). Fiche technique des moteurs.

17. Maxon. (2005). Fiche technique des réducteurs.

18. http://www.gecif.net/articles/genie_electrique/filtre/

20. http://d1.amobbs.com/bbs_upload782111/files_10/ourdev_264005.pdf

21. Grasser, A. D. Arrigo, S. Columbia, A. C. Rufer, and S. Member, "JOE : A Mobile , Inverted Pendulum," no. February, pp. 107–114, 2002.

22. http://d1.amobbs.com/bbs_upload782111/files_10/ourdev_264005.pdf

23. Kadir, H.B. (2005).*Modelling and control of a balancing robot using digital state space approach.* Faculty of Electrical Engineering, UniversitiTeknologi Malaysia

24. Abdollah, M. (2006). *Proportional Integral Sliding Mode Control of a two-wheeled balancing robot.* Faculty of Electrical Engineering, UniversitiTeknologi Malaysia