

République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
Université Akli Mohand Oulhadj de Bouira



Faculté des sciences et sciences appliquées
Département Génie Electrique

Pour l'obtention du diplôme de master en Electronique des systèmes embarqués

Thème :

**Etude , réalisation et commande d'un robot auto
balancé à deux roues**

Sujet Proposé par :

***M^r*. FEKIR Mohamed**

Présenté par :

***M^r*. BOUSSAAD Smail
M^r. BOUACHRINE Salah**

***M^r*. HAMMOUCHE Sofiane de l'Université de Bouira**

***M^r*. FEKIR Mohamed de l'Université de Bouira**

***M^r*. BENSALIA Yassine de l'Université de Bouira**

***M^r*. ASSOUNI Salime de l'Université de Bouira**

Président

Directeur du Mémoire

Examineur

Examineur

PROMOTION 2017/2018

Remerciements

Avant toute chose, nous tenons à remercier Allah le tout puissant qui nous a donné la force et la patience d'accomplir ce travail. Ce mémoire n'aurait pas été possible sans l'intervention consciente d'un grands nombre de personne, nous souhaitons ici les en remercier. En premier lieu, nous tenons à remercier très chaleureusement notre prof **Fekir Mohamed** qui nous a permis de bénéficier de son encadrement. La patience et la confiance qu'il nous a témoigné ont été déterminant pour la réalisation de ce mémoire. Nos plus sincères remerciements s'étendent également au corps enseignants de la faculté des science et science appliquées pour la richesse et la qualité de leur enseignement et qui déploie de grands efforts pour assurer à leurs étudiants une formation actualisée.

Nous remercions aussi nos parents pour le soutien inconditionnel pour leur contribution et leur patience.

En fin nous tenons à remercier tous ceux qui d'une manière ou d'une autre ont participé dans la réussite de ce travail.

Dédicaces

Nous dédions ce travail....

A nos très chères mères

Affables, honorables, aimables : vous représentez pour nous le symbole de la bonté par excellence, la source de tendresse, et l'exemple de dévouement. Vous n'avez pas cessé de nous encourager et de prier pour nous.

Nous vous dédions ce travail en témoignage de notre profond amour. Puisse Dieu, le tout puissant, vous préserver et vous accorder santé, longue vie et bonheur.

A nos très chers pères

Aucune dédicace ne saurait exprimer l'amour, l'estime, le dévouement et le respect que nous avons toujours eu pour vous.

Rien au monde ne vaut les efforts fournis jour et nuit pour nous éduquer et pour notre bien être. Ce travail est le fruit des sacrifices que vous avez consentis pour notre éducation et notre formation.

A nos chers sœurs et frères

En témoignage de l'attachement, de l'amour et de l'affection que nous portons pour vous.

Nous vous dédions ce travail avec tous nos vœux de bonheur, de santé et de réussite.

A nos chères ami (e)s

En témoignage de l'amitié qui nous uni et des souvenirs de tous les moments que nous avons passé ensemble, nous vous dédions ce travail et nous vous souhaitons une vie pleine de santé et de bonheur.

Liste de figure

Chapitre 1 :

Figure 1.1 Schéma de robot Cartésien.....	3
Figure 1.2 Robot Cartésien Toshiba.....	4
Figure 1.3 Schéma de robot Cylindrique	4
Figure 1.4 Robot Seiko.....	4
Figure 1.5 Robot FANUC	5
Figure 1.6 Schéma de robot SCARA	5
Figure 1.7 Robot Sankyo.....	6
Figure 1.8 Robot COMAU	6
Figure 1.9 Robot Anthropomorphe	6
Figure 1.10 Robot Kawasaki	7
Figure 1.11 Robot ABB	7
Figure 1.12 Robot de type unicycle	8
Figure 1.13 Robot de type tricycle	8
Figure 1.14 Robot de type voiture.....	9
Figure 1.15 Robot de type omnidirectionnel	9
Figure 1.16 Photo d'un pendule inversé classiqueavecson schéma de principe	11
Figure 1.17 Photo d'un pendule inversé parallèle avec son schéma de principe	11
Figure 1.18 Photo d'un pendule inversé furuta avec son schéma de principe	12
Figure 1.19 Photo d'un pendule inversé stabilisé par volant d'inertie avec son schéma de principe	12
Figure 1.20 Photo d'un pendule inversé sur deux roues avec son schéma de principe	13
Figure 1.21 Photo de Joe-le-pendule et la photo du segway	14
Figure 1.22 Photo de robot Equibot	14
Figure 1.23 Photo de Segbot	15
Figure 1.24 IBOT en position deux roues et en position descente escalies	16

Chapitre 2 :

Figure 2.1 La carte Arduino Nano	19
Figure 2.2 La carte Arduino Uno	19
Figure 2.3 La carte Arduino Mega2560.	20
Figure 2.4 Moteur cc 3-6 v.....	21
Figure 2.5 Courant dans un champ magnétique.....	22
Figure 2.6 Le système balais-collecteurs.	22
Figure 2.7 Schéma équivalent du moteur à courant continu.....	23

Liste de figure

Figure 2.8 Montage de circuit L298N	26
Figure 2.9 Une carte GY-521 dotée de la puce MPU6050 (Accéléromètre et gyroscope).....	27
Chapitre 3:	
Figure 3.1 Corps et roues de robot	29
Figure 3.2 Schéma de carrosserie libre du sous-système mécanique du robot.	30
Figure 3.3 Schéma de régulation d'une boucle fermée.....	33
Figure 3.4 L'organigramme de fonctionnement de robot auto balance.....	34
Figure 3.5 Schéma de fonctionnement	35
Figure 3.6 Courbe pour régler le PID.....	38
Chapitre 4 :	
Figure 4.1 Les lignes SDA et SCL.....	41
Figure 4.2 Mécanisme des adresses	43
Figure 4.3 Affichage des valeurs de gyroscope	44
Figure 4.4 Orientation de gyroscope	44
Figure 4.5 Pilot moteur L298n	45
Figure 4.6 Les composants utilisé	48
Figure 4.7 Montage réalisé	48
Figure 4.8 Raccordement du module MPU6050 avec l'arduino	49
Figure 4.9 Placement du gyroscope sur le châssis	49
Figure 4.10 Placement de la batterie dans un deuxième étage.....	50
Figure 4.11 Corps du robot	50
Figure 4.12 Partie électronique	51
Figure 4.13 Version finale du robot	51
Figure 4.14 Constitution d'une batterie LiPo	52
Figure 4.15 Constitution interne de la batterie	52
Figure 4.16 Schéma de l'équilibrage	53
Figure 4.17 Mesure de la tension de batterie	54
Figure 4.18 LiPo/ LiFe balance charger	55
Figure 4.19 IDE arduino.....	55
Figure 4.20 Structure d'un programme Arduino	56
Figure 4.21 Télécharger le programme vers l'Arduino.....	57

Listes des Acronymes et Symboles

Acronymes et symboles :

A	Ampère.
C_f	Couple de frottement
C_m	Couple moteur
CMS	Composant monté en surface.
DDL	Degré de liberté
DMP	Digital Motion Processor.
E	Force électromotrice (Volt).
EPFL	Ecole Polytechnique fédérale de lausanne.
EPROM	Erasable Programmable Read Only Memory.
f	Constant de frottement
FEM	Force électromotrice.
I	Courant consommé par le moteur (Ampère).
I2C	Inter-integrated Circuit.
INT	INTeger, entier. Groupe de 16 bits, 2 octets groupés, considérés comme représentant un nombre entier négatif ou positif.
J	Moment d'inertie
k	Une constante caractéristique du moteur.
k_c	Constant de couple
LONG	"entier long". Groupe de 32 bits, 4 octets groupés, considérés comme représentant un nombre entier négatif ou positif.
LOOP	Boucle.
LOW	Bas.
MEMS	Micro-electroMechanical systems

Listes des Acronymes et Symboles

OUT	Souvent l'abréviation pour OUTput, Sortie. Est toujours en rapport avec le sens carte Arduino vers extérieur.
OUTPUT	Sortie.
PID	Proportionnel intégral dérivé
PIN	Broche.
POWER	Puissance, alimentation.
PROM	Programmable Read Only Memory.
PWM	Pulse Width Modulation.
PWR	Power
R	Résistance interne du bobinage (Ohm).
READ	Lire.
RELAY	Relais.
RESISTOR	Résistance.
RX	Receive, réception.
SCL	Serial Clock Line
SDA	Serial Data Line
SENSOR	Capteur
SERIAL	Série.
SETUP	Initialisation.
SWITCH	Basculer, interrupteur.
TX	Abréviation pour Transmit, transmission.
U	Tension d'alimentation du moteur (Volt).
USB	Universal Serial Bus.

Listes des Acronymes et Symboles

V	Volt.
WIRE	Câble, fil.
ω	Vitesse angulaire en rad/s.

Sommaire

Introduction Générale.....	1
Chapitre 01 : Généralité sur les Robots	
1.1. Introduction	2
1.2. Définition d'un robot.....	2
1.3. Domaines d'utilisation de robot	2
1.3.1. Les robots industriels	2
1.3.2. Robots domestique ou ménagers	2
1.3.3. Robots en médecine et chirurgie	3
1.4. Différent type de robot	3
1.4.1 Les robots fixés	3
1.4.1.1. Classification des robots fixés	3
a. Robot cartésien	3
b. Robot cylindrique	4
c. Robot sphérique.....	4
d. Robot SCARA.....	5
e. Robot parallèle.....	6
f. Robot anthropomorphe	6
1.4.2. Les robots mobiles	7
1.4.2.1. Classification des robots mobiles	7
a. Robot unicycle.....	7
b. Robot tricycle	8
c. Robot voiture	8
d. Robot omnidirectionnel	9
1.5. Les pendules inversés.....	9
1.5.1 Définition	9
1.5.2. Intérêt de l'étude d'un pendule inversé.....	10
1.5.2.1 Le pendule inversé classique	10
1.5.2.2 Le pendule inversé en parallèle	11
1.5.2.3 Le pendule inversé de furuta.....	12
1.5.2.4 Le pendule inversé stabilisé par volant d'inertie	12
1.5.2.5 Le pendule inversé sur deux roues.....	13
1.5.2.6 Le pendule inversé sur deux roues EQUIBOT	14
1.5.2.7 Le pendule inversé sur deux roues SEGBOT	14
1.5.2.8 Fauteuil roulant électrique L'IBOT	15
1.6. Conclusion.....	16

Chapitre : 02 Le materiel Utilisé

2.1. Introduction	17
2.2. Les microcontrôleurs	17
2.3. La carte arduino	17
2.3.1. Logiciel	18
2.3.2. Les types des cartes arduino	19
2.3.2.1. Arduino nano	19
2.3.2.2. Arduino uno	19
2.3.2.3. Arduino Mega2560	20
2.4. Matériels et logiciels dérivés	20
2.4.1. Matériels libre et logiciel libre	21
2.5. Les moteurs	21
2.5.1. Principe de fonctionnement	21
2.5.2. Contrôle de sens et vitesse de rotation des moteur CC	23
2.6. L298N vs L293D	24
2.5.1. Connexions	24
2.5.2. Circuit	25
2.7. Les capteurs	26
2.7.1. Présentation du MPU6050	26
2.8. Conclusion	27

Chapitre 03 : Modelisation et équations dynamiques

3.1. Introduction	28
3.2. Principe de fonctionnement	28
3.3. Modélisation et équations dynamiques	29
3.4. Définition des coordonnées	29
3.4.1. Dérivation des équations dynamiques du mouvement	31
3.4.2. Dérivation du Lagrangien	31
3.4.3. La commande PID	32
3.5. Organisation de système en boucle fermée	33
3.6. L'organigramme pour le robot auto balance	34
3.7. Schéma de fonctionnement	35
3.8. Implémentation de PID numérique sur l'arduino	36
3.8.1. Le régulateur proportionnel (P : première règle)	36
3.8.2. Le régulateur proportionnel intégral (PI : première et seconde règle)	36
3.8.3. Le régulateur proportionnel dérivé (PD : première et troisième règle)	36
3.8.4. Le régulateur proportionnel intégral et dérivé (PID : première, seconde et troisième règle)	37
3.8.5. Régler les coefficients d'un PID	37

3.9. Conclusion.....	39
----------------------	----

Chapitre 04 : Conception de chassis et assemblage

4.1. Introduction	40
4.2. Realisation d'un circuit et un programme pour circuit mpu6050	40
4.3. Bus I2C.....	40
4.4. Positionnement du gyroscope.....	44
4.5. Modulation de largeur d'impulsion (PMW)	44
4.5.1. Activation moteur	45
4.5.2. Rotation moteur	45
4.6. Hardware	46
4.6.1. Réalisation de chassis	47
4.6.2. Constitution d'une batterie LiPo	52
4.7. Software	55
4.8. Conclusion.....	57
Conclusion Générale	58
Bibliographie	59
Reference des figure	62

Introduction générale

Le robot se distingue des autres machines par la complexité des mouvements qu'il peut produire en composant des mouvements élémentaires et surtout par son « intelligence », ou ses capacités décisionnelles en autonomie. Les robots sont donc, initialement, des machines destinées à remplacer l'Homme, faites pour travailler à sa place.[1]

Le travail présente un robot auto balancé à deux roues, il est basé sur la configuration du pendule inversé qui repose sur les dynamiques d'un système d'équilibrage pour équilibrer, par exemple le corps humain est un pendule inversé qui équilibre le haut du corps autour des articulations de la cheville à chaque pas, Pour équilibrer un robot à pendule inversé à deux roues, il est nécessaire de disposer d'informations précises sur l'angle d'inclinaison actuel en utilisant une unité de mesure (accéléromètres et gyroscopes intégrés sur la puce MPU6050) et comparer cet angle avec l'angle cible qui entrée dans les paramètres du PID (K_p, K_i, K_d). Un contrôleur PID est capable de contrôler l'angle du pendule avec ses paramètres K_p , K_i et K_d , après le PID envoi l'information au driver moteur qui contrôle les deux roues.

La rédaction du document, est scindée en quatre principaux chapitres :

Dans le premier chapitre, nous allons faire une description des robots où nous donnerons des définitions sur les différents types des robots et les différents types d'architectures.

Dans le deuxième chapitre, on décrit le matériel électrique nécessaire pour la réalisation et la conception de notre projet.

Quant au troisième chapitre on abordera la modélisation et la description du model et équation de mouvement du robot, et la démonstration à partir de Lagrangien.

Le quatrième chapitre en discutera la réalisation et les étapes de montages et d'assemblage du projet, et comment ajuster le PID pour équilibrer le robot.

Le travail ce termine par une conclusion générale.

Chapitre 01:

GENERALITE SUR LES ROBOTS

1.1. Introduction :

Dans ce chapitre, on va fournir une des cryptions sur les robots, quelques définitions concernant les robot. Les types de tâches, les différents types d'architectures et le principe de fonctionnement, ainsi que les caractéristiques des robots. Nous allons parler sur les pendules inversés et différents types des robots utilisant le pendule inversé.

1.2. Définition d'un Robot [1] [2]:

Le mot "robot" est apparu pour la première fois sous la plume du dramaturge Karel Capek. Ce néologisme est inspiré du mot tchèque robota, "travail forcé". Publié en 1921 et représenté la même année au Théâtre national de Prague, ce drame d'anticipation, traduit en anglais en 1923, a rencontré un égal succès à New York et à Londres. L'action se situe sur une île, où des robots sont fabriqués à la chaîne

Le robot est un dispositif électromécanique avec des contrôleurs capables de traiter des informations de l'environnement de manière automatique avec des capteurs et d'exécuter des opérations selon un programme.

1.3. Domaines d'utilisation de robot [3] :**1.3.1. Les robots industriels :**

Les Robots industriels sont des robots utilisés dans un environnement de fabrication industrielle. Ils sont utilisés dans la fabrication des automobiles, des composants et des pièces électroniques, des médicaments et de nombreux produits.

1.3.2. Robots domestiques ou ménagers :

Les Robots utilisés dans les maisons. Nous utilisons plusieurs robots dans notre vie quotidienne, tels que les aspirateurs robotiques, robots nettoyeurs de piscines, balayeuses, nettoyeurs gouttières et autres robots qui peuvent faire différentes tâches. Et autre, des robots de surveillance et de télé présence pouvaient être considérés comme des robots ménagers s'ils sont utilisés dans cet environnement.

1.3.3. Robots en médecine et chirurgie :

Dans des hôpitaux aussi, l'être humain utilise des robots par exemple pour l'aide à faire des opérations de chirurgie. Aussi, le robot infirmier est encore en projet. Il existe un robot qui s'appelle cyber squelette HAL aide les personnes à se déplacer. Et le robot patient permet aux futurs chirurgiens-dentistes d'apprendre à soigner sans faire de dégâts.

1.4. Différent Type De Robot [4] :

Les robots peuvent être classés en fonction de l'environnement dans lequel ils opèrent :

1.4.1. Les Robots Fixés :

Les robots fixes sont principalement des manipulateurs robotiques industriels qui fonctionnent dans des environnements bien définis adaptés aux robots. Les robots industriels effectuent des tâches répétitives spécifiques telles que des pièces à souder ou à peindre dans des usines de fabrication automobile. Avec l'amélioration des capteurs et des dispositifs d'interaction homme-robot, des manipulateurs robotiques sont de plus en plus utilisés dans un environnement moins contrôlé comme la haute précision chirurgicale.

1.4.1.1. Classification Des Robots Fixés :

a. Robot Cartésien:

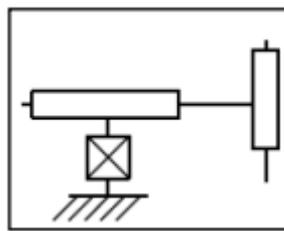


Figure 1.1: Schéma de robot Cartésien.

Caractéristiques :

3 axes \perp 2 à 2, série, 3 DDL.

Très bonne précision.

Lent.

Exemples :



Figure 1.2: Robot Cartésien Toshiba.

b. Robot Cylindrique :

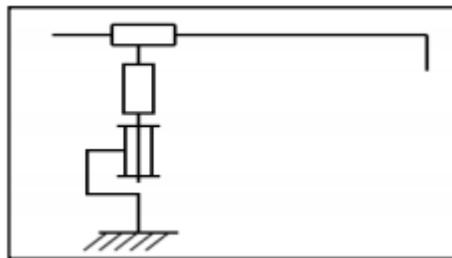


Figure 1.3: Schéma de robot Cylindrique.

Caractéristiques :

3 axes, série, 3 DDL.

Espace de travail cylindrique.

Très rapide.

Exemples :



Figure 1.4: Robot Seiko

c. Robot Sphérique:**Caractéristiques :**

3 axes, série, 3 DDL.

Espace de travail sphérique.

Grande charge utile.

Exemple :

Figure 1.5: Robot FANUC

d. Robot SCARA:

SCARA = Selective Compliance Articulated Robot for Assembly

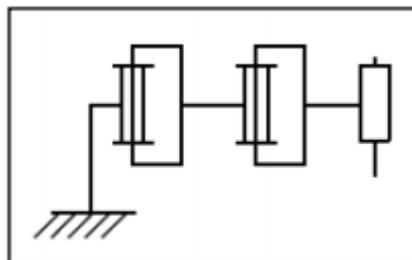


Figure 1.6: Schéma de robot SCARA.

Caractéristique :

3 axes, série 3 DDL.

Espace de travail cylindrique.

Précis.

Très rapide.

Exemples :

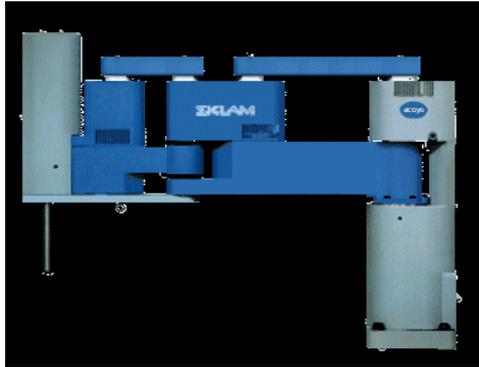


Figure 1.7: Robot Sankyo.

e. Robot Parallèle :

Caractéristiques :

Plusieurs chaînes cinématiques en parallèle.

Espace de travail réduit.

Précis (grande rigidité de la structure).

Rapide.

Exemples :



Figure 1.8: Robot COMAU

f. Robot Anthropomorphe:

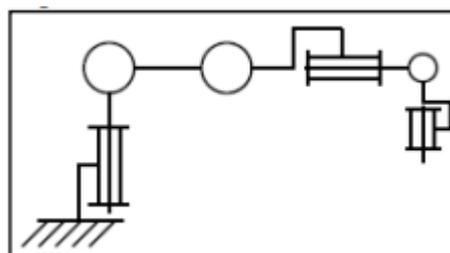


Figure 1.9: robot Anthropomorphe.

Caractéristiques :

Reproduisent la structure d'un bras humain.

6 axes, série, 6 DDL.

Exemples :

Architecture standard :



Figure 1.10: Robot Kawasaki.

Architecture à parallélogramme :



Figure 1.11: Robot ABB.

1.4.2. Les Robots mobiles [5]:

Un robot mobile est celui qui peut se déplacer dans son environnement de façon indépendante. Pour ce faire, le robot doit pouvoir naviguer, la portée et la précision de navigation requise varie en fonction de la taille du robot et du type de sa tâche.

1.4.2.1. Classification des robots mobiles[6] :**a. Robot unicycle :**

Un robot de type unicycle est actionné par deux roues indépendantes, il possède éventuellement des roues folles pour assurer sa stabilité. Son centre de rotation est situé sur l'axe reliant les deux roues motrices.

C'est un robot non-holonome, en effet il est impossible de le déplacer dans une direction perpendiculaire aux roues de locomotion. Sa commande peut être très simple, il est en effet assez facile de le déplacer d'un point à un autre par une suite de rotations simples et de lignes droites.

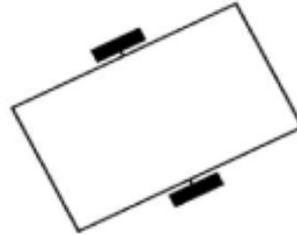


Figure 1.12: Robot de type unicycle.

b. Robot tricycle :

Un robot de type tricycle est constitué de deux roues fixes placées sur un même axe et d'une roue centrée orientable placée sur l'axe longitudinal. Le mouvement du robot est donné par la vitesse des deux roues fixes et par l'orientation de la roue orientable. Son centre de rotation est situé à l'intersection de l'axe contenant les roues fixes et de l'axe de la roue orientable.

C'est un robot non-holonome. En effet, il est impossible de le déplacer dans une direction perpendiculaire aux roues fixes. Sa commande est plus compliquée. Il est en général impossible d'effectuer des rotations simples à cause d'un rayon de braquage limité de la roue orientable.

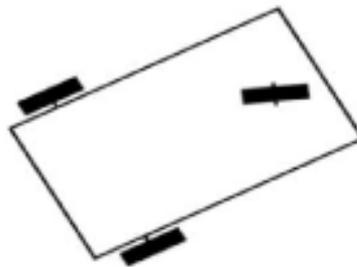


Figure 1.13: Robot de type tricycle.

c. Robot voiture :

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

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

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

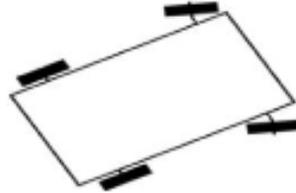


Figure 1.14: Robot de type voiture.

d. Robot omnidirectionnel :

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

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

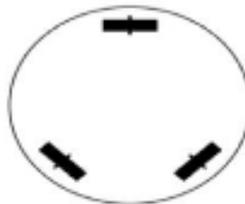


Figure 1.15: Robot de type omnidirectionnel.

1.5. Les pendules inversés :

1.5.1. Définition:

Un pendule inversé est un système mécanique, automatique composé d'une tige posée en position d'équilibre instable verticale sur une base qui peut être fixe ou mobile. La dynamique non linéaire et instable de ce système physique a fait de lui un système très intéressant et largement étudié dans la communauté des automaticiens, ce qui explique l'existence de plusieurs aspects et formes de pendules inversés.[07]

On peut trouver divers types de pendules inversés tels que : le pendule inversé simple, le pendule inversé double, le pendule inversé de Furuta, etc. Dans la suite ces différents types seront présentés.

1.5.2. Intérêt de l'étude d'un pendule inversé :

L'étude du pendule inversé a une importance dans la robotique. Il y a plusieurs raisons à cela :

➤ L'homme est un pendule double inversé dont les deux axes de rotation principaux sont les chevilles et les hanches. Quand nous sommes en position debout, nos articulations travaillent sans arrêt pour nous y maintenir. Les spécialistes qui travaillent à la réalisation de prothèses pour les hanches sont amenés à utiliser le modèle du pendule double inversé pour calculer l'ensemble des contraintes qui sont soumises à la prothèse.

➤ Dans le même ordre d'idée, la robotique utilise ce genre de concept.

➤ On voit apparaître des moyens de locomotion dotés de deux roues montées sur un même axe sur lequel on est en position debout. On accélère en se penchant en avant et en ralenti en se penchant en arrière. Le système est le même que le pendule inversé. La stabilité est aussi assurée par des gyroscopes mais nous n'entrerons pas dans ce genre de détails.

➤ La complexité de ce type de robot fait que les étudiants en automatique, en robotique en font un énorme travail de recherche puisqu'il englobe beaucoup de notions intéressantes pour eux : programmation, automatisation, mécanique...[08]

1.5.2.1. Le pendule inversé classique :

Ce système mécanique consiste en un chariot de masse M qui peut se déplacer horizontalement et librement sur un rail de guidage en supportant une tige de masse m libre en rotation autour d'un pivot.

L'application d'une force $f(t)$ va provoquer le déplacement du chariot et la déviation du pendule d'un angle $\theta(t)$ par rapport à la verticale. Ceci constitue un système mécanique sous actionné à deux degrés de liberté et un seul actionneur.

Dans le cas de stabilisation d'un tel système, le but de la commande serait de ramener le pendule à partir d'une position initiale à sa position d'équilibre instable et le maintenir autour de cette position.[09]

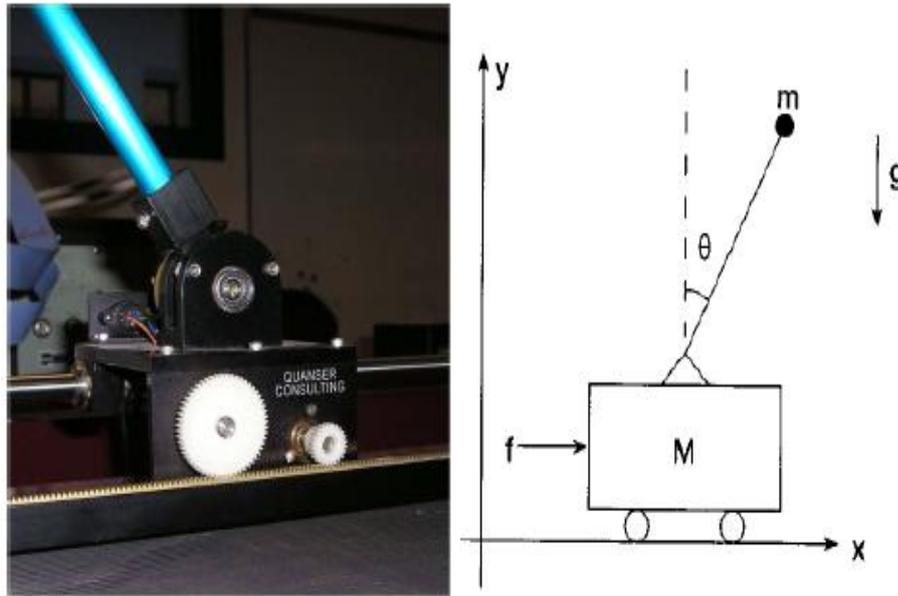


Figure 1.16: Photo d'un pendule inversé classique avec son schéma de principe.

1.5.2.2. Pendule inversé en parallèle :

Dans ce cas, le chariot supporte deux tiges indépendantes : une tige (L) de longueur l_L et une autre (B) de longueur l_B , les deux en rotation libre, Le déplacement $x(t)$ de la base va engendrer une déviation d'un angle θ_L par rapport à la verticale sur la première tige et d'un angle θ_B par rapport à la verticale sur la deuxième tige.[10]

Le but de la commande consiste à stabiliser les deux tiges autour de la verticale au point d'équilibre instable et les maintenir même dans le cas de présence de perturbations externes.

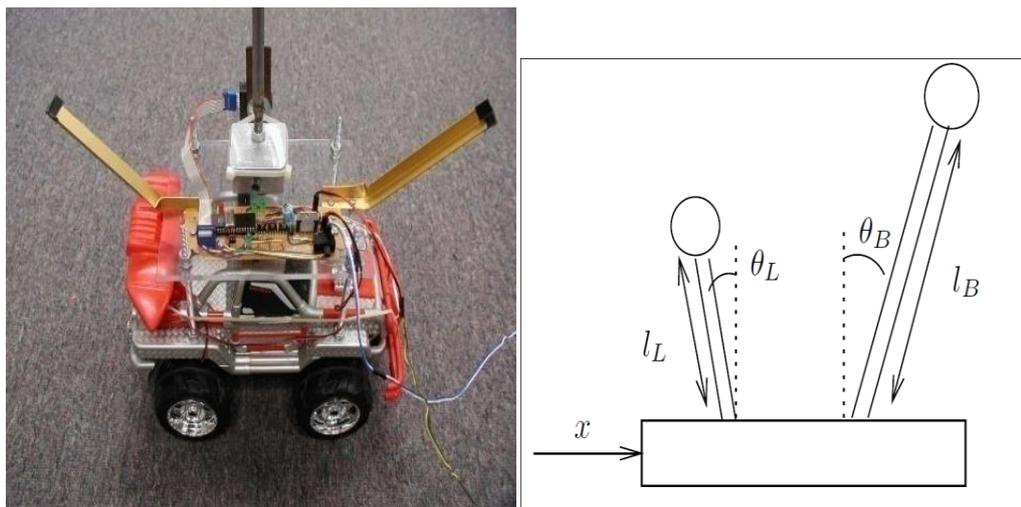


Figure 1.17: Photo d'un double pendule inversé parallèle avec son schéma de principe.

1.5.2.3. Le pendule inversé de Furuta:

Ce pendule a été conçu par K. Furuta Il est composé d'un bras actionné en rotation dans le plan horizontale, à son extrémité vient s'ajouter un pendule inversé monté en équilibre instable. La rotation infinie du bras assure la stabilisation et le maintien du pendule autour de la verticale au point d'équilibre instable.[11]

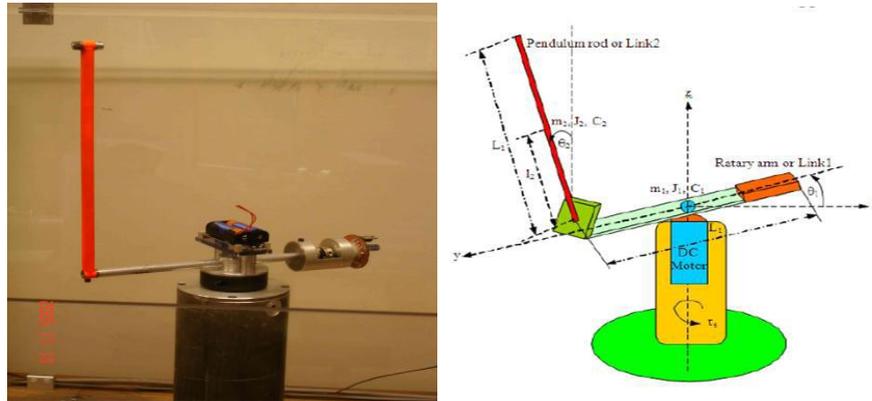


Figure 1.18: Photo de pendule de Furuta avec son schéma de principe

1.5.2.4. Le pendule inversé stabilisé par volant d'inertie:

C'est un système composé de deux corps mécaniques : un pendule inversé en rotation libre autour d'un pivot lié au bâti et un volant actionné (roue d'inertie) dont le centre de masse coïncide avec l'extrémité du pendule.

Le principe de fonctionnement de ce système est basé sur le mouvement de rotation du volant d'inertie qui provoque, par les effets dynamiques qu'il induit, la rotation de pendule.

Le but de la commande consiste à stabiliser le pendule autour de la verticale au point d'équilibre instable et le maintenir dans cet état.[12]

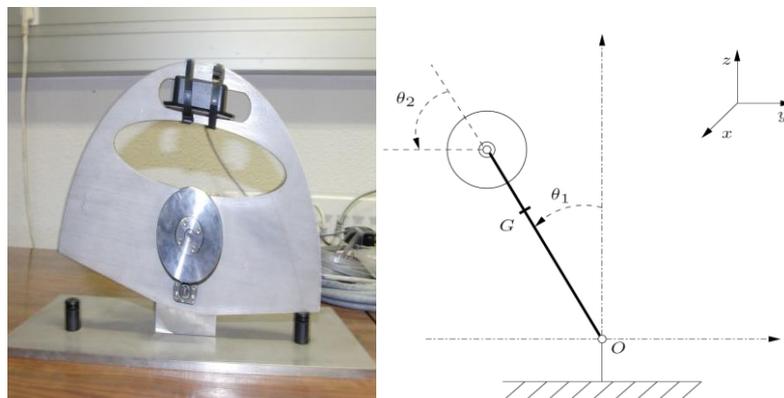


Figure1.19: Photo de pendule inversé stabilisé par volant d'inertie avec son schéma de principe

1.5.2.5. Le pendule inversé sur deux roues :

Le pendule inversé sur deux roues est considéré comme un prototype académique largement étudié dans le domaine de l'automatique. Comme son nom l'indique, il est constitué d'une base mobile (l'essieu et les deux roues) surmontée d'un pendule inversé en rotation libre autour d'un pivot (articulation passive) entre l'essieu et la tige du pendule.

Le principe de fonctionnement du système est très simple en théorie : quand le pendule penche vers l'avant, la partie mobile doit le rattraper en effectuant un mouvement vers l'avant et vice versa. La difficulté réside dans le réglage de l'intensité de réaction des roues afin d'agir face à l'angle que fait le pendule avec la verticale.[13]

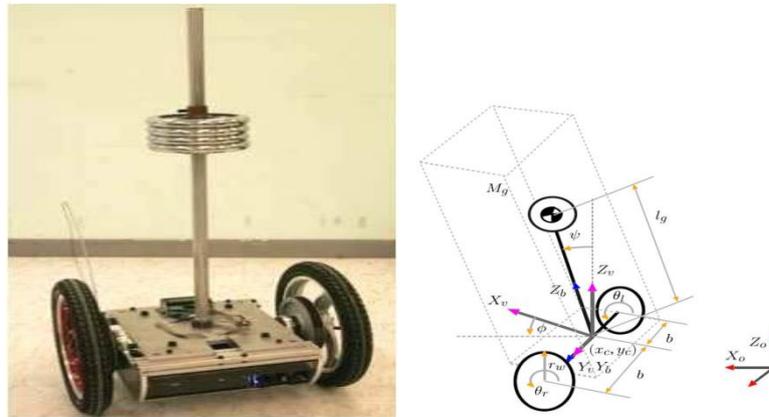


Figure 1.20: Photo de pendule inversé sur deux roues avec son schéma de principe

Comme une application réussite du principe de ce pendule, on peut citer le Segway

C'est un moyen de transport électrique et individuel inventé par Dean Kamen. Le dernier s'est inspiré d'un pendule inversé miniature connu sous le nom de Joe-le-Pendule qui a été développé en 1996 au laboratoire d'Electronique Industrielle de l'EPFL

Le pilotage du Segway est basé sur l'inclinaison du corps de l'utilisateur dans la direction de déplacement voulue. En effet, il faut se pencher en avant ou en arrière pour avancer ou reculer et à droite ou à gauche pour tourner.[14]

D'ailleurs, c'est à ce type de pendule inversé que nous nous sommes intéressés dans ce travail.



Figure 1.21: Photo du Joe-le-Pendule et Photo du Segway.

1.5.2.6. Le pendule inversé sur deux roues Equibot :

Est un robot d'équilibrage développé par DaniPiponi. Il utilise un capteur infrarouge pointu pour mesurer la distance au sol et utilise des informations de distance pour en déduire son angle d'inclinaison.[15]

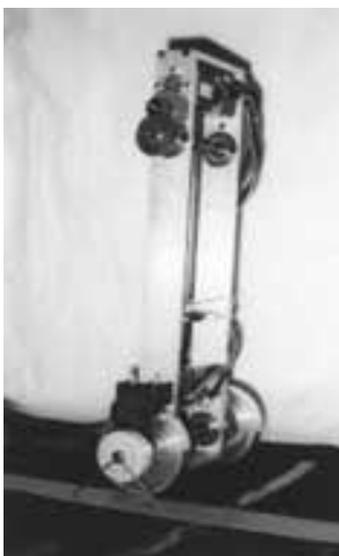


Figure 1.22: Photo de robot Equibot.

1.5.2.7. Le pendule inversé sur deux roues SEGBOT :

Le Segbot a été développé au printemps 2004. L'objectif était de concevoir et de construire un robot d'équilibrage à deux roues basé sur les mêmes principes du Segway.[16]

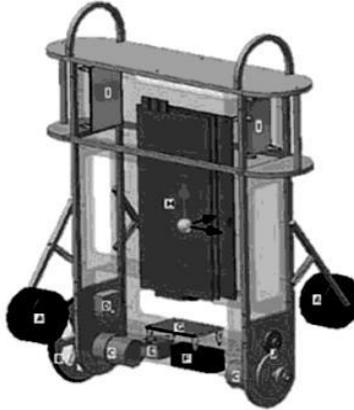


Figure 1.23: Photo de robot Segbot.

1.5.2.8. Fauteuil roulant électrique L'IBOT [17] :

Est un fauteuil roulant électrique automatisé développé par l'inventeur américain Dean Kamen et commercialisé par " Johnson & Johnson ". C'est un appareil équipé de quatre roues motrices qui est capable grâce à sa technologie médicale impressionnante, d'aider des personnes avec des problèmes graves de mobilité. Il est équipé d'un certain nombre de dispositifs qui le distingue de la plupart des fauteuils roulants actionnés, à savoir :

- Il permet à l'utilisateur d'être à la même hauteur que son interlocuteur (équilibre du type pendule inversé deux roues).
- Il assure un déplacement sur différents types de terrains (trottoirs, escaliers, gravier, etc).
- Il fonctionne à piles.

Cependant il présente quelques difficultés telles que :

- Il n'est pas adapté aux enfants ni aux personnes trop fortes.
- Il ne permet pas la verticalisation.
- Il est très encombrant : trop grand pour entrer dans une voiture.



Figure 1.24: IBOT en position deux roues et en position descente escaliers.

1.6. Conclusion :

Aujourd'hui, les robots sont des prototypes ils sont devenu important dans notre vie et auront une place dans l'avenir proche, ces machines facilitent de plus en plus la vie de l'être humain, ce qui consolide cette relation humain-robot. Les robots ont des avantages qui les rendent et améliorent vraiment notre vie. Dans le prochain nous aborderons le choix et les caractéristiques des différents composants utilisés dans ce projet.

Chapitre 02:

CHOIX DU MATÉRIEL

2.1. Introduction :

Dans ce chapitre, nous nous intéresserons aux composants électronique et dispositifs nécessaires pour la réalisation et la conception de notre projet (robot auto balancé). La réalisation de ce robot nécessite divers notions dans plusieurs domaines électronique, informatique et automatique.

Dans notre projet, nous avons utilisé des composants électroniques et autre électromécaniques. A cet effet, le choix ainsi que les caractéristiques des composants utilisés dans ce projet sont discutés.

2.2. Les Microcontrôleurs :

Un système embarqué, pour fonctionner, a besoin d'une unité de traitement, d'une mémoire morte pour le programme (PROM, EPROM, ...), d'une mémoire vive pour stocker les données, de circuits d'interfaçage pour connecter les périphériques qui vont permettre la communication avec l'extérieur.[18]

Les microcontrôleurs sont des circuits intégrés logiques qui permettent l'exécution d'un programme dont les actions dépendent de l'état des variables d'environnement du système. Ces circuits intégrés les moins cher sont de véritables ordinateurs miniatures autonomes utilisés dans les voitures (ordinateurs de bords, airbags, système de freinage freins ABS...), les lecteurs DVD, les fours à micro-ondes, etc... On peut définir un microcontrôleur comme une unité de traitement de l'information de type microprocesseur à laquelle on a ajouté des périphériques internes permettant de réaliser des montages sans nécessiter l'ajout de composants externes.[19]

2.3. La carte Arduino :

Le projet « Arduino » a été initié par un groupe d'enseignants et d'étudiants d'une école italienne en 2004 – 2005.

L'Arduino emprunte son nom au « Bar di ReArduino », un pub de la ville d'Ivrée, située en Italie du Nord². Le nom propre Arduino est un cognat d'Ardenne.[20]

Licence libre, cependant, certains composants de la carte, comme le microcontrôleur par exemple, ne sont pas en licence libre. Le microcontrôleur peut être programmé pour analyser et produire des signaux électriques, de manière à effectuer des tâches très diverses

comme la domotique (le contrôle des appareils domestiques - éclairage, chauffage...), le pilotage d'un robot, etc. C'est une plateforme basée sur une interface entrée/sortie simple. Il était destiné à l'origine principalement mais pas exclusivement à la programmation multimédia interactive en vue de spectacle ou d'animations artistiques. C'est une partie de l'explication de la descendance de son interface de programmation de processing.[21]

Un module Arduino est généralement construit autour d'un microcontrôleur ATML AVR (ATmega328 ou ATmega2560 pour les versions récentes, ATmega168 ou ATmega8 pour les plus anciennes), et de composants complémentaires qui facilitent la programmation et l'interfaçage avec d'autres circuits. Chaque module possède au moins un régulateur linéaire 5V et un oscillateur à quartz 16 MHz (ou un résonateur céramique dans certains modèles).[22]

Le microcontrôleur est préprogrammé avec un boot loader de façon à ce qu'un programmeur dédié ne soit pas nécessaire.

Les modules sont programmés au travers d'une connexion série RS-232, mais les connexions permettant cette programmation diffèrent selon les modèles. Les premiers Arduino possédaient un port série, puis l'USB est apparu sur les modèles Diecimila, tandis que certains modules destinés à une utilisation portable se sont affranchis de l'interface de programmation, relocalisée sur un module USB-série dédié (sous forme de carte ou de câble).[23]

L'Arduino utilise la plupart des entrées/sorties du microcontrôleur pour l'interfaçage avec les autres circuits. Le modèle Diecimila 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 femelle HE14 situés sur le dessus de la carte, les modules d'extension venant s'empiler sur l'Arduino. Plusieurs sortes d'extensions sont disponibles dans le commerce.[24]

Certains modules non officiels, compatibles avec la technologie Arduino, utilisent des connecteurs mâles pour une utilisation aisée avec des plaques de test.

2.3.1. Logiciel :

Le logiciel de programmation des modules Arduino est une application Java, libre et multiplateforme, servant d'éditeur de code et de compilateur, et qui peut transférer le firmware et le programme au travers de la liaison série (RS-232, Bluetooth ou USB selon le module). Il

est également possible de se passer de l'interface Arduino, et de compiler et uploader les programmes via l'interface en ligne de commande.[25]

Le langage de programmation utilisé est le C++, compilé avec avr-g++ 4, et lié à la bibliothèque de développement Arduino, permettant l'utilisation de la carte et de ses entrées/sorties. La mise en place de ce langage standard rend aisé le développement de programmes sur les plates-formes Arduino, à toute personne maîtrisant le C ou le C++.[22]

2.3.2. Les types des cartes arduino :[26]

Les modules d'origine des différentes versions de l'Arduino sont fabriqués par la société italienne Smart Project. Quelques-unes des cartes de marque Arduino ont été conçues par la société américaine SparkFun Electronics.

Dix-sept versions des cartes de type Arduino ont été produites et vendues dans le commerce à ce jour dans l'ordre chronologique ci-dessous et voilà les cartes les plus utilisés :

2.3.2.1. Arduino Nano : une version encore plus petite de l'Arduino alimenté par USB et utilisant un ATmega168 (avant la version 3) ou ATmega328 (à partir de la version 3.0) de type CMS.

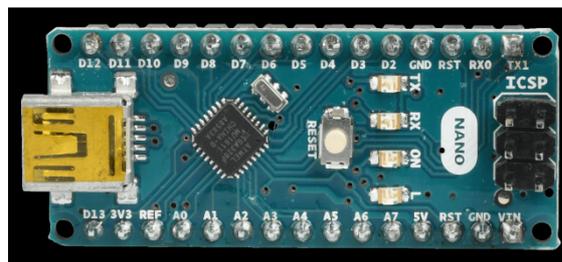


Figure-2.1 : La carte ArduinoNano.[14]

2.3.2.2. Arduino Uno :

Utilise un ATmega328 comme les derniers modèles de Duemilanove, mais alors que le Duemilanove utilisait une puce pour la programmation via un connecteur USB, le Uno utilise une puce ATmega8U2 programmé comme un convertisseur série9.



Figure-2.2 : La carte Arduino Uno. [14]

2.3.2.3 Arduino Mega2560 : est équipé d'un ATmega2560 de type CMS, augmentant la mémoire totale disponible à 256 ko. Il est équipé aussi de la nouvelle puce USB ATmega8U210.

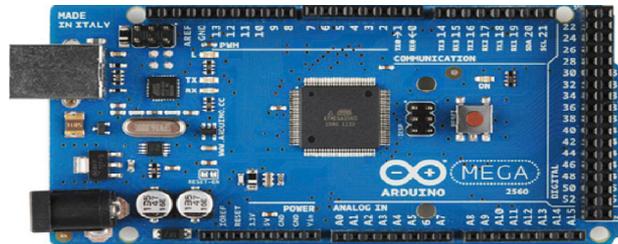


Figure-2.3 : La carte Arduino Mega2560. [14]

2.4. Matériels et logiciels dérivés :

Plusieurs logiciels et matériels compatibles Arduino (bien que non officiels) ont été produits commercialement par d'autres fabricants :

- **Les kits Freeduino:** compatibles Arduino, fabriqués et vendus par Modern DeviceCompany.
- **Le FreeduinoThrough-Hole compatible Arduino :** module évitant l'utilisation de composants CMS et destiné à une fabrication personnelle.
- **Le Boqrduino :** un clone peu cher du Decimila, avec des connecteurs pour une utilisation sur plaque de test.
- **Le Romeo :** variation de l'Arduino Uno dédiée à la robotique, fournit des interfaces de puissance pour contrôler des moteurs.

Plusieurs logiciels compatibles Arduino (bien que non officiels) ont été développés pour faciliter la programmation des cartes Arduino. Bien que très différents les uns des autres, ils permettent une programmation sans utilisation du langage Arduino. On retrouve :

- La solution graphique Ardublock qui permet une programmation par assemblage de blocs fonctionnels. C'est un plug-in qui ajoute une interface graphique à celle d'Arduino.
- La solution graphique MiniBlock qui permet elle aussi une programmation par assemblage de blocs fonctionnels. C'est un plug-in qui ajoute une interface graphique à celle d'Arduino.[27]

Carte de contrôle pour imprimantes 3D RepRap comme les Generation7 Electronics 15 ou les Sanguinololu16 base sur une carte Arduino complétée par des drivers de moteur pas à pas des acquisitions de température et des sorties de puissance.

Certain logiciels musicaux proposent grâce à leur module série une intégration de l'Arduino dans le processus de création musical

2.4.1. Matériel libre et logiciel libre :

Le design matériel de l'Arduino est distribué sous licence Créative Commons Attribution Share-Alike 2.5 et est disponible sur le site d'Arduino. Les schémas ainsi que les typons de circuits sont également disponibles. Le code source de l'environnement de programmation et les bibliothèques embarquées sont disponibles sous licence.[28]

2.5. Les moteurs :[29]

Dans notre projet chaque roue est motorisée indépendamment de l'autre, offrant le contrôle de lacet par différentiation, voir inversion du régime de rotation. Le moteur utilisé est de type moteur à courant continu de 3 - 6 v.



Figure-2.4. Moteur cc 3-6 v. [15]

2.5.1 Principe de fonctionnement : [30]

Le moteur à courant continu se compose :

- De l'inducteur ou du stator,
- De l'induit ou du rotor,
- Du collecteur et des balais.

Lorsque le bobinage de l'inducteur du moteur est alimenté par un courant continu, sur le même principe qu'un moteur à aimant permanent, il se crée un champ magnétique (flux d'excitation) de direction Nord-Sud.

Une spire capable de tourner sur un axe de rotation est placée dans le champ magnétique. De plus, les deux conducteurs formant la spire sont chacun raccordés électriquement à un demi collecteur et alimentés en courant continu via deux balais frotteur.

D'après la loi de Laplace (tout conducteur parcouru par un courant et placé dans un champ

magnétique est soumis à une force), les conducteurs de l'induit placés de part et d'autre de l'axe des balais (ligne neutre) sont soumis à des forces F égales mais de sens opposé en créant un couple moteur : l'induit se met à tourner.

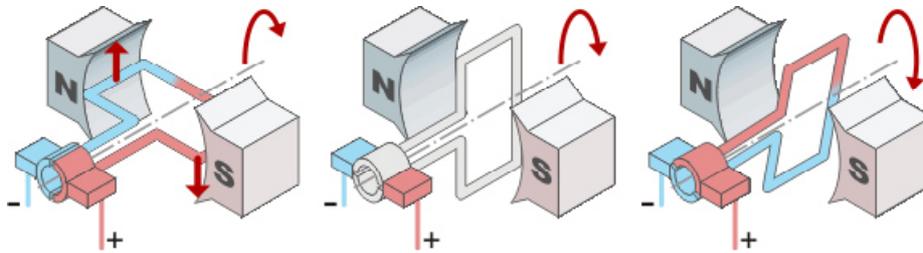


Figure 2.5: Courant dans un champ magnétique. [15]

Si le système balais-collecteurs n'était pas présent (simple spire alimentée en courant continu), la spire s'arrêterait de tourner en position verticale sur un axe appelé communément "ligne neutre".

Le système balais-collecteurs **Figure (2.6)** a pour rôle de faire commuter le sens du courant dans les deux conducteurs au passage de la ligne neutre. Le courant étant inversé, les forces motrices sur les conducteurs le sont aussi permettant ainsi de poursuivre la rotation de la spire.

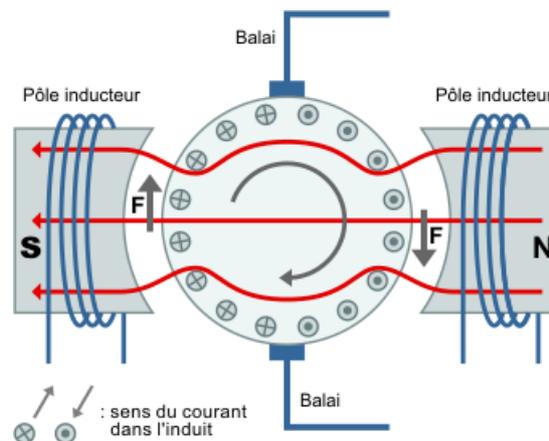


Figure 2.6: Le système balais-collecteurs. [16]

Dans la pratique, la spire est remplacée par un induit (rotor) de conception très complexe sur lequel sont montés des enroulements composés d'un grand nombre de spires raccordés à un collecteur "calé" en bout d'arbre. Dans cette configuration, l'induit peut être considéré comme un seul et même enroulement semblable à une spire unique

La Figure (2.7) montre le schéma équivalent du moteur à courant continu. Si le courant est en régime continu on peut négliger l'inductance dans ce circuit, le moteur se comporte comme une résistance en série avec un générateur de tension.

I : courant consommé par le moteur (Ampère)

U : Tension d'alimentation du moteur (Volt)

E : force électromotrice (Volt)

R : résistance interne du bobinage (Ohm)

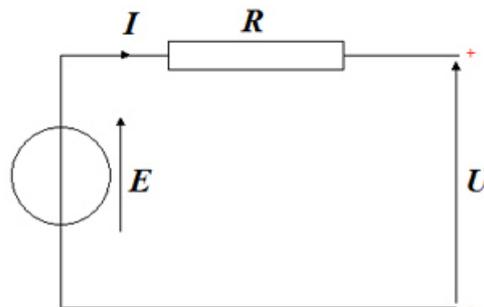


Figure 2.7: Schéma équivalent du moteur à courant continu. [16]

Du schéma équivalent de la figure (2.7), on déduit l'équation

$$U = E + RI \quad (1)$$

La tension FEM est proportionnelle à la vitesse angulaire

$$E = K\omega \quad (2)$$

Avec :

k : une constante caractéristique du moteur.

ω : vitesse angulaire en rad/s.

le principe fondamental de la dynamique (seconde loi de newton) appliqué à un solide en rotation permet d'écrire:

$$C_m - C_f = J \frac{dw(t)}{dt} \Leftrightarrow k_c I(t) - f.w(t) = j . \frac{dw(t)}{dt} \quad (3)$$

où C_m : couple moteur

C_f : couple de frottement

w : la vitesse angulaire

J : moment d'inertie

k_c : constant de couple

f : constant de frottement

2.5.2. Contrôle de sens et vitesse de rotation des moteur à courant continue:

Notre problème est comment on peut faire tourner un moteur dans les deux sens en continuant bien sûr de gérer sa vitesse .la solution bien sûr c'est grâce au **pont en H**.

C'est un circuit intégré développé par des électroniciens et qui est maintenant disponibles en commerce.

C'est une puce électronique est un circuit intégré, c'est-à-dire une sorte de petite boîte qui contient des composants miniaturisés et connectés entre eux afin de répondre à une fonction particulière

En branchant cette petite carte à un microcontrôleur ou carte Arduino il est possible de contrôler simultanément deux moteurs en courant continu, ou un moteur pas à pas.

On trouve sur le marché deux types avec des références différents et bien sur avec des schémas interne différents.

2.6. L298N vs L293D :[31]

Pour des projets impliquant de petits moteurs de faible puissance, le L293D est souvent privilégié parce qu'il coûte moins cher, qu'il s'insère facilement dans une breadboard et qu'il n'est pas nécessaire de lui ajouter des diodes puisque ces dernières sont déjà intégrées à l'intérieur du circuit intégré, mais ne supporte pas des courants supérieurs à 1 A, et il n'est pas vraiment conçu pour qu'on puisse facilement le munir d'un radiateur.

Le L298N peut quant à lui supporter un courant de 2 A, ce qui permet de contrôler des moteurs plus puissants. Si nous achetons un L298N (circuit intégré seulement), ça nous coûte plus cher qu'un L293D, ses broches sont disposées d'une façon qui s'insèrent difficilement dans une breadboard, et nous devenons lui ajouter des diodes, des condensateurs, un radiateur, etc.

Tous ces inconvénients disparaissent si nous nous procurons un module déjà monté...même le prix, soit significativement moins que ce que coûte habituellement un circuit intégré L298N seul, sans les autres composants soudés sur la carte.

2.6.1. Connexions :

Les 4 sorties "OUT1", "OUT2", "OUT3" et "OUT4" servent à brancher les moteurs (un moteur branché à OUT1 et OUT2, et un deuxième moteur, s'il y a lieu, branché à OUT3 et OUT4).

Les moteurs et la cartes sont alimentés grâce aux connecteurs GND et +12 V.

Il n'est pas nécessaire que la tension de l'entrée "+12 V" soit de 12 V. Selon les moteurs que nous utilisons, il est possible qu'une tension aussi faible que 6 V soit suffisante, et qu'une tension de 12 V soit trop élevée.

En fait, nous pouvons soumettre l'entrée +12 V à une tension allant jusqu'à 35 V si nous le désirons mais il faut faire attention: si nous soumettons l'entrée +12V à une tension qui dépasse 12 V, nous devons d'abord enlever le jumper 12 V afin d'éviter de griller le régulateur de tension qui alimente la sortie 5 V.

Contre toute attente, le connecteur "+5V" n'est pas une entrée, mais une sortie qui fournit 5 V et qui pourrait nous permettre, par exemple, d'alimenter un microcontrôleur. Cette sortie 5 V est disponible à la condition que le "jumper 12 V" soit en place (il faut donc que la tension au connecteur +12V n'excède pas 12 V).

Notre Arduino sera branché aux entrées ENA, IN1, IN2, IN3, IN4 et ENB.

Si IN1 est à 5 V pendant que IN2 est à 0 V, le moteur branché aux sorties OUT1 et OUT2 tourne dans un sens. Si IN1 est à 0 V pendant que IN2 est à 5 V, le moteur tourne dans l'autre sens.

ENA est l'entrée "enable": le moteur tourne à la condition que ce pin se trouve à un niveau logique haut. Si nous laissons en place le jumper qui se trouve sur ce pin, elle demeurera toujours active et le moteur tournera à sa vitesse maximale. Pour contrôler la vitesse de rotation du moteur, nous devons retirer le jumper et brancher le pin ENA à une sortie PWM de l'Arduino.

Nous l'aurons deviné, c'est le même principe pour les pins ENB, IN3 et IN4, sauf que ces entrées permettent de contrôler le moteur relié aux sorties OUT3 et OUT4.

2.6.2. Circuit : [32]

Voici donc un circuit qui répond à toutes ces conditions. Ce n'est pas indiqué sur le schéma, mais l'Arduino est déjà alimenté par le câble USB qui le relie à l'ordinateur, ce qui explique pourquoi nous n'avons pas relié la sortie +12V du contrôleur L298N au pin 5V de l'Arduino.

Driver Moteur L293D shield Conçu pour fonctionner parfaitement avec arduino.

L'Arduino shield pour Moteur comprend 4 conducteurs H-Bridge, nous permettant de contrôler jusqu'à 4 moteurs à courant continu avec un contrôle indépendant de vitesse et la direction de notre Robot ou nos moteurs pour être précis.

Ce Shield fournit également deux pins de plus pour connecter 2 servo moteurs.

Il est un outil facile à utiliser le module qui se branche tout simplement sur le connecteur femelle de notre Arduino. Il est bien documenté avec des schémas et des bibliothèques disponibles partout.

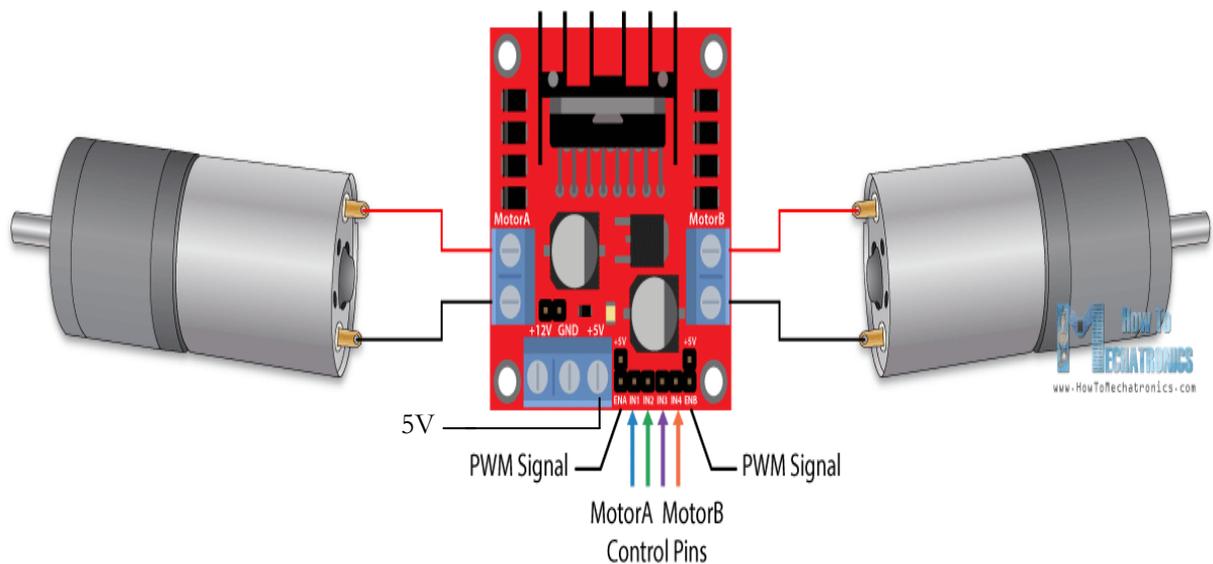


Figure-2.8 : Montage de circuit L298N. [17]

2.7. Les 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 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 une bonne estimation en les combinant. [33]

2.7.1. Présentation du MPU6050 :

La puce est équipée d'un accéléromètre 3 axes et d'un gyroscope 3 axes. Outre l'acquisition des mesures, la puce est aussi équipée d'un processeur appelé DMP (Digital Motion Processor) se chargeant de convertir les mesures brutes en données compréhensibles pour un humain. Le processeur DMP réalise les calculs directement sur la puce allégeant ainsi le travail de l'Arduino. On peut aussi lire les mesures brutes sans passer par le traitement du DMP mais c'est très compliqué. [34]



Figure-2.9 : Une carte GY-521 dotée de la puce MPU6050 (Accéléromètre et gyroscope). [18]

2.8. Conclusion :

Nous avons donné dans ce chapitre quelques définitions concernant les composants utilisés dans ce projet. On a étudié le fonctionnement des composants et on a compris comment relier les composants entre eux. Dans le chapitre suivant on va parler sur la modélisation du robot étudié dans notre projet.

Chapitre 03:

MODELISATION ET EQUATION DYNAMIQUE

3.1. Introduction :

Dans le chapitre précédent nous avons parlé sur le matériel dont nous avons besoin pour la réalisation. Pour ce chapitre en discute sur la régulation et les méthodes de réglage des paramètres de PID et son implémentation sans faire des calculs ainsi le principe de fonctionnement du robot auto balance.

3.2. Principe de fonctionnement :

Pour développer un système de contrôle fiable et performant pour un robot d'équilibrage à deux roues, il est essentiel de comprendre les paramètres du système. La représentation de ceux-ci peut être réalisée à travers un modèle mathématique. La théorie du pendule inversé est plus généralement connue sous le nom de théorie des pôles et des chariots, et bien que le robot d'équilibrage à deux roues ne se compare pas directement au pôle et au chariot, les mêmes principes sont en vigueur. Dans le modèle du système, le chariot correspond aux roues tandis que le poteau correspond au châssis du robot.[35]

Les coefficients de frottement ont été négligés dans ce projet car le robot devrait traverser de nombreux types de terrains et de surfaces. Si les coefficients devaient être pris en compte lors de la conception et de la mise en œuvre des systèmes de commande, des capteurs, des circuits et une consommation électrique supplémentaires seraient alors nécessaires pour obtenir ces nouvelles valeurs pendant le fonctionnement. Le temps, les efforts et les ressources nécessaires pour créer cette capacité dépassent de loin les avantages que l'on pourrait attendre de leur inclusion.[36]

Il est nécessaire de généraliser les effets des roues gauche et droite et de les incorporer sous le terme combiné «roues». Cela simplifie les calculs car les deux roues travailleront à l'unisson pour maintenir la stabilité. Pour déterminer les exigences spécifiques de couple (forces) pour chaque roue, la valeur des roues peut être divisée par deux pour une valeur de roue unique approximative. Cette approche est considérée comme acceptable car le terrain et la surface varient entre les roues sur certains terrains.

Le principe du pendule inversé a pour but de garder les roues sous le centre de la masse du châssis du robot. Si le robot commence à s'incliner vers l'avant, alors pour maintenir la stabilité, la roue devra avancer pour revenir sous la masse du châssis. Si cela n'est pas maintenu, le robot va simplement tomber. Les dynamiques de système suivantes sont associées au problème mathématique.[37]

3.3. Modélisation et équation dynamique :

Un sous-système mécanique se compose du corps du robot et de deux roues. Le corps peut être modélisé comme un pendule inversé avec la masse concentrée au centre de gravité et l'axe de rotation au-dessus de l'axe de roues. Pour la dérivation d'équations de mouvement, un modèle planaire est utilisée lorsque le robot se déplace. Supposons qu'il n'y ait pas de glissement entre les roues et le sol. [38]

3.4. Définition des coordonnées :

Ce système robotique est similaire au problème «le pendule inversé sur un chariot en mouvement». De même au problème du pendule, ce système de robot peut être séparé en deux parties: “partie roue” et “partie corps” (**Figure 3.1**). La principale différence entre ce système et le pendule inversé est l'inertie et la position du centre de masse de la partie du corps, car nous cherchons un modèle plus détaillé du système plutôt que de l'approcher comme une masse ponctuelle. Pour ce système, nous devons calculer l'inertie et la position du corps basée sur la distribution de masse de la partie du corps, qui est déterminée par la conception matérielle.[39]

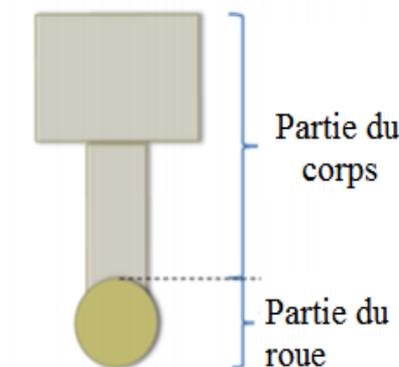


Figure 3.1: Corps et roues de robot.

Pour dériver les équations dynamiques de ce système, nous avons défini les coordonnées comme indiqué à la **Figure 3.2**. Dans ce système, nous supposons que le robot se déplace horizontalement sans glisser entre le sol et les roues.

CHAPITRE 03 MODELISATION ET EQUATION DYNAMIQUE

X	position horizontale du centre de la roue par rapport à une origine définie
X_c	position horizontale du point m de la partie du corps par rapport à une origine définie
φ	angle de rotation de la roue dans le sens des aiguilles d'une montre par rapport à l'axe horizontal à $t = 0$
z_c	position verticale du corps de la partie du corps du sol
θ	angle de rotation dans le sens des aiguilles d'une montre de la partie du corps à partir de la position verticale
M	masse de la partie du corps
m_w	masse de la roue
R	Rayon de la roue
L	longueur entre le corps et le centre de la roue
τ_0	couple appliqué
I	l'inertie de la partie du corps
I_w	inertie de la roue

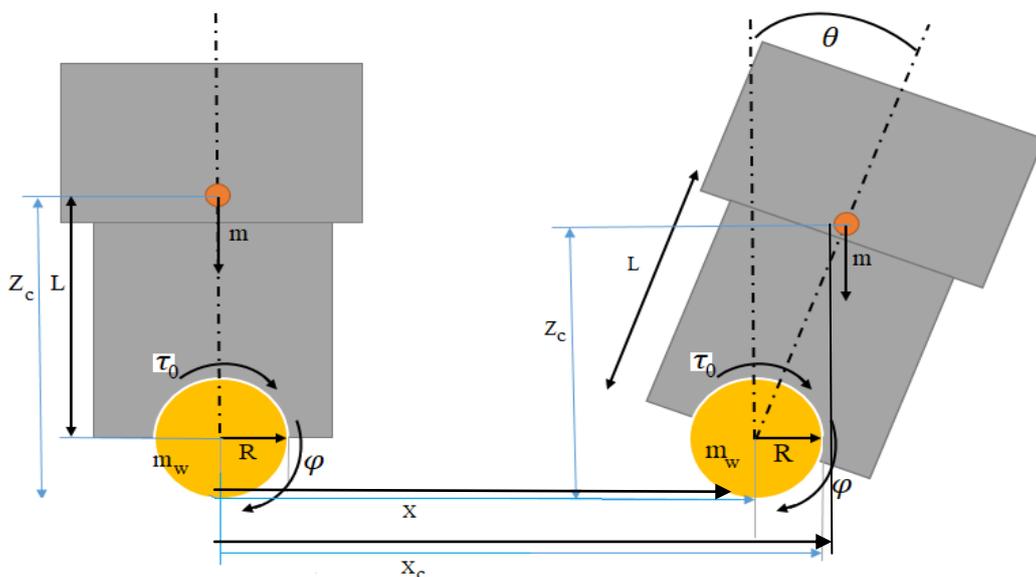


Figure 3.2: Schéma de carrosserie libre du sous-système mécanique du robot.

3.4.1. Dérivation des équations dynamiques du mouvement :

Nous avons utilisé la méthode de Lagrange pour dériver les équations dynamiques de ce système. Nous pouvons écrire x x_c z_c \dot{x} \dot{x}_c \dot{z}_c comme suit:

$$x = R \varphi \quad (1) \qquad \dot{x} = R \dot{\varphi} \quad (2)$$

$$x_c = R \varphi + L \sin \theta \quad (3) \qquad \dot{x}_c = R \dot{\varphi} + L \dot{\theta} \cos \theta \quad (4)$$

$$z_c = R + L \cos \theta \quad (5) \qquad \dot{z}_c = -L \dot{\theta} \sin \theta \quad (6)$$

En continuant, nous pouvons écrire l'énergie potentielle E_p (zéro est défini comme l'énergie potentielle à la position verticale) et l'énergie cinétique E_c comme suit:

$$E_p = mg(R + L \cos \theta) - mg(R + L) = mgL(\cos \theta - 1) \quad (7)$$

$$E_c = \frac{1}{2} m_w \dot{x}^2 + \frac{1}{2} I_w \dot{\varphi}^2 + \frac{1}{2} m \dot{x}_c^2 + \frac{1}{2} m \dot{z}_c^2 + \frac{1}{2} I \dot{\theta}^2 \quad (8)$$

$$= \frac{1}{2} (I_w + m_w R^2 + mR^2) \dot{\varphi}^2 + mRL \cos \theta \dot{\varphi} \dot{\theta} + \frac{1}{2} (I + mL^2) \dot{\theta}^2 \quad (9)$$

3.4.2. Dérivation du lagrangien :

Lagrangien L peut être écrit comme la différence entre l'énergie cinétique E_c et l'énergie potentielle E_p

$$L = E_c - E_p \quad (10)$$

$$= \frac{1}{2} (I_w + m_w R^2 + mR^2) \dot{\varphi}^2 + mRL \cos \theta \dot{\varphi} \dot{\theta} + \frac{1}{2} (I + mL^2) \dot{\theta}^2 - mgL(\cos \theta - 1) \quad (11)$$

Par conséquent, les équations dynamiques pour la coordonnée x et la coordonnée θ peuvent être dérivées comme suit:

Avec coordonné φ

$$\frac{\partial L}{\partial \varphi} = (I_w + m_w R^2 + mR^2) \dot{\varphi} + mRL \dot{\theta} \cos \theta \quad (12)$$

$$\frac{\partial L}{\partial \varphi} = 0 \quad (13)$$

$$\frac{d}{dt} \left(\frac{\partial L}{\partial \dot{\varphi}} \right) - \frac{\partial L}{\partial \varphi} = (I_w + m_w R^2 + mR^2) \ddot{\varphi} + mRL \cos \theta \ddot{\theta} - mRL \sin \theta \dot{\theta}^2 = \mu \quad (14)$$

CHAPITRE 03 MODELISATION ET EQUATION DYNAMIQUE

Avec coordonné θ

$$\frac{\partial L}{\partial \dot{\theta}} = mRL \cos \theta \dot{\phi} + (1+mL^2)\dot{\theta} \quad (15)$$

$$\frac{\partial L}{\partial \theta} = -mRL\dot{\phi} \sin \theta \ddot{\theta} + mgL \sin \theta \quad (16)$$

$$\frac{d}{dt} \left(\frac{\partial L}{\partial \dot{\theta}} \right) - \frac{\partial L}{\partial \theta} = (1+mL^2)\ddot{\theta} + mRL \cos \theta \ddot{\phi} - mgL \sin \theta = \chi \quad (17)$$

où μ et χ sont des forces généralisées (couples) pour chaque coordonnée

Ensuite, nous pouvons réécrire ces équations dynamiques non linéaires dans un style de matrice de second ordre comme suit:

$$\begin{bmatrix} Iw + mw R^2 + mR^2 & mRL \cos \theta \\ mRL \cos \theta & I + mL^2 \end{bmatrix} \begin{bmatrix} \ddot{\phi} \\ \ddot{\theta} \end{bmatrix} + \begin{bmatrix} 0 & -mRL \sin \theta \dot{\theta} \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \end{bmatrix} + \begin{bmatrix} 0 \\ -mgL \sin \theta \end{bmatrix} = \begin{bmatrix} \mu \\ \chi \end{bmatrix} \quad (18)$$

3.4.3. La commande PID :

La commande PID est dite aussi (correcteur, régulateur, contrôleur), se compose de trois termes P, I et D, d'où le 'P' correspond au terme proportionnel, 'I' pour terme intégral et 'D' pour le terme dérivé. Les régulateurs PID sont probablement les plus largement utilisés dans le contrôle robotique. Même les plus complexes systèmes de contrôle industriel peut comporter un réseau de contrôle dont le principal élément de contrôle est un module de contrôle PID.[40]

Le régulateur PID est une simple implémentation de retour d'information. Il a la capacité d'éliminer la compensation de l'état d'équilibre grâce à l'action intégrale, et il peut anticiper le futur grâce à une action dérivée

L'équation d'un régulateur PID est donnée comme suit:

$$\mathbf{R}(t) = K_p \cdot e(t) + K_i \cdot \int_0^t e(t) dt + K_d \cdot \frac{de(t)}{dt} \quad (19)$$

Le développement extraordinaire des microprocesseurs et microcontrôleurs a provoqué des changements importants dans la conception des systèmes de commande régulation des procédés industriels. Leur puissance de calcul et leur faible coût les rendent

aptes à prendre également en charge les aspects commande/régulation des procédés en question avec des performances nettement supérieures à celles des régulateurs analogiques. Autant dire que la régulation numérique va se substituer progressivement à la régulation analogique. Cependant la mise en œuvre de ces nouveaux systèmes numériques de contrôle commande nécessite de nouvelles compétences pour l'utilisateur.[41]

3.5. Organisation de système en boucle fermée :

Dans le système en boucle fermée, on trouve un capteur pour mesurer la sortie dans notre cas nous avons utilisé le gyroscope mpu6050 qui va permettre de mesurer l'inclinaison de corps du robot ensuite envoie les valeurs vers l'arduino qui nous avons utilisé comme comparateur pour élaborer l'erreur entre la consigne et la mesure de sortie aussi comme correcteur qui commande les moteurs en fonction de signal d'erreur.[42]

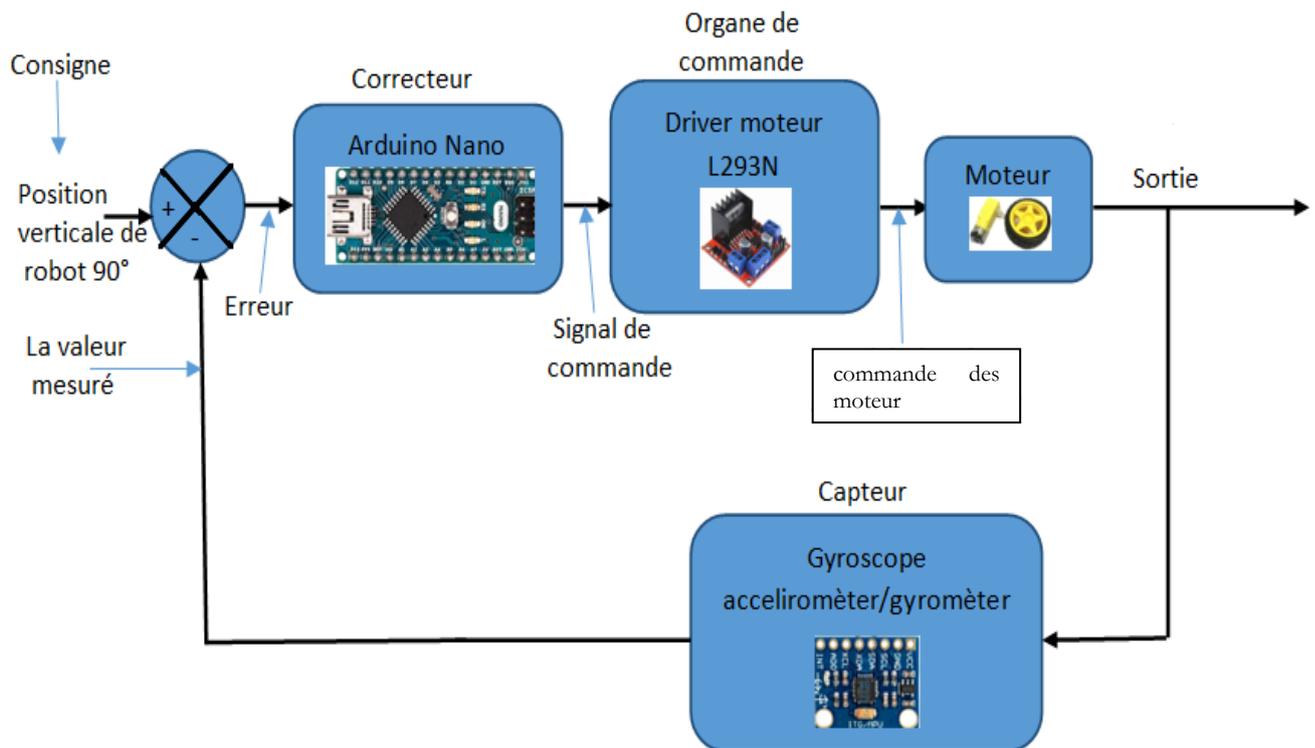


Figure 3.3:Schéma de régulation d'une boucle fermée.

3.6. L'organigramme pour le robot auto balance :

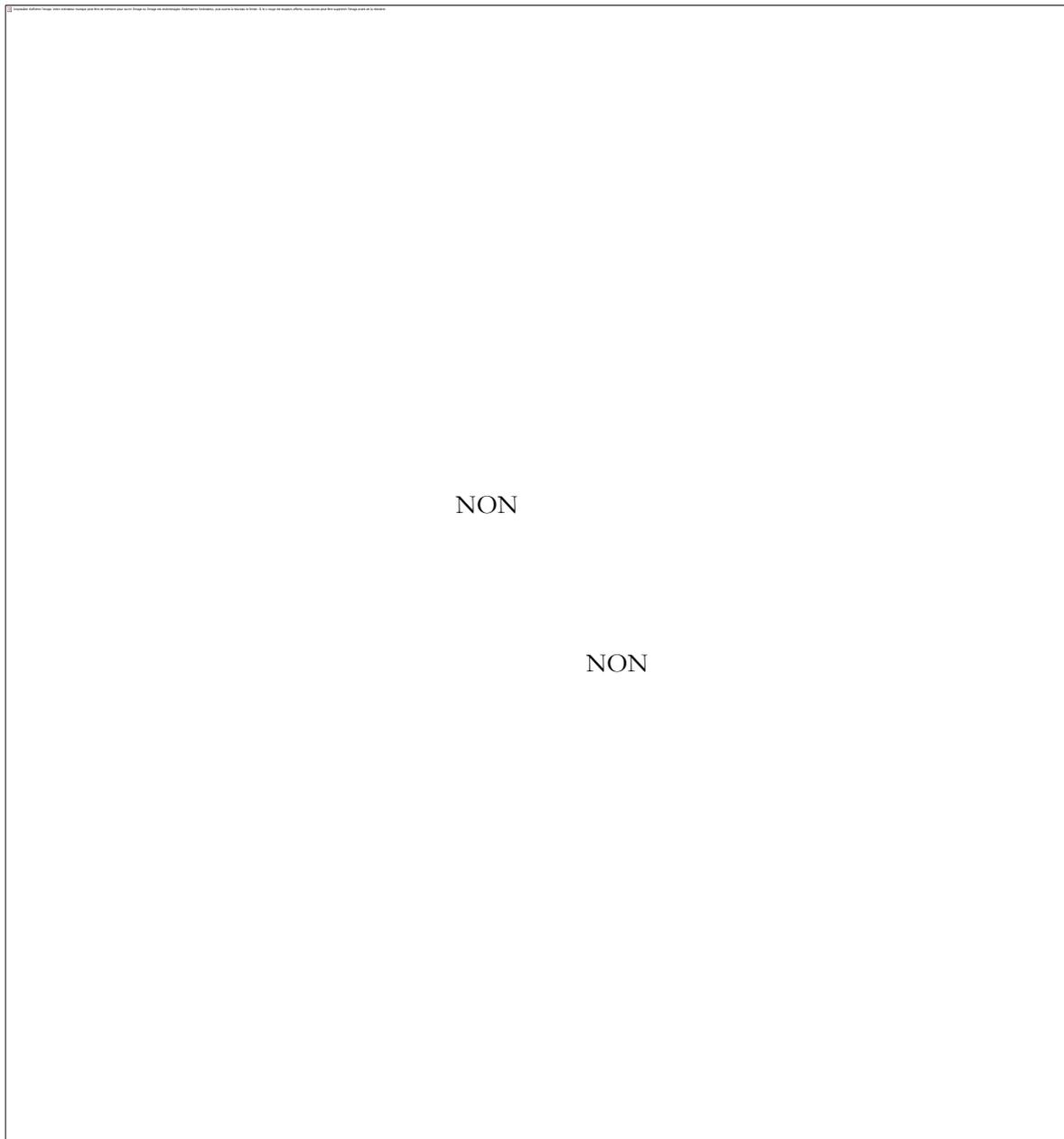


Figure 3.4: L'organigramme de fonctionnement de robot auto balance.

3.7. Schéma de fonctionnement :

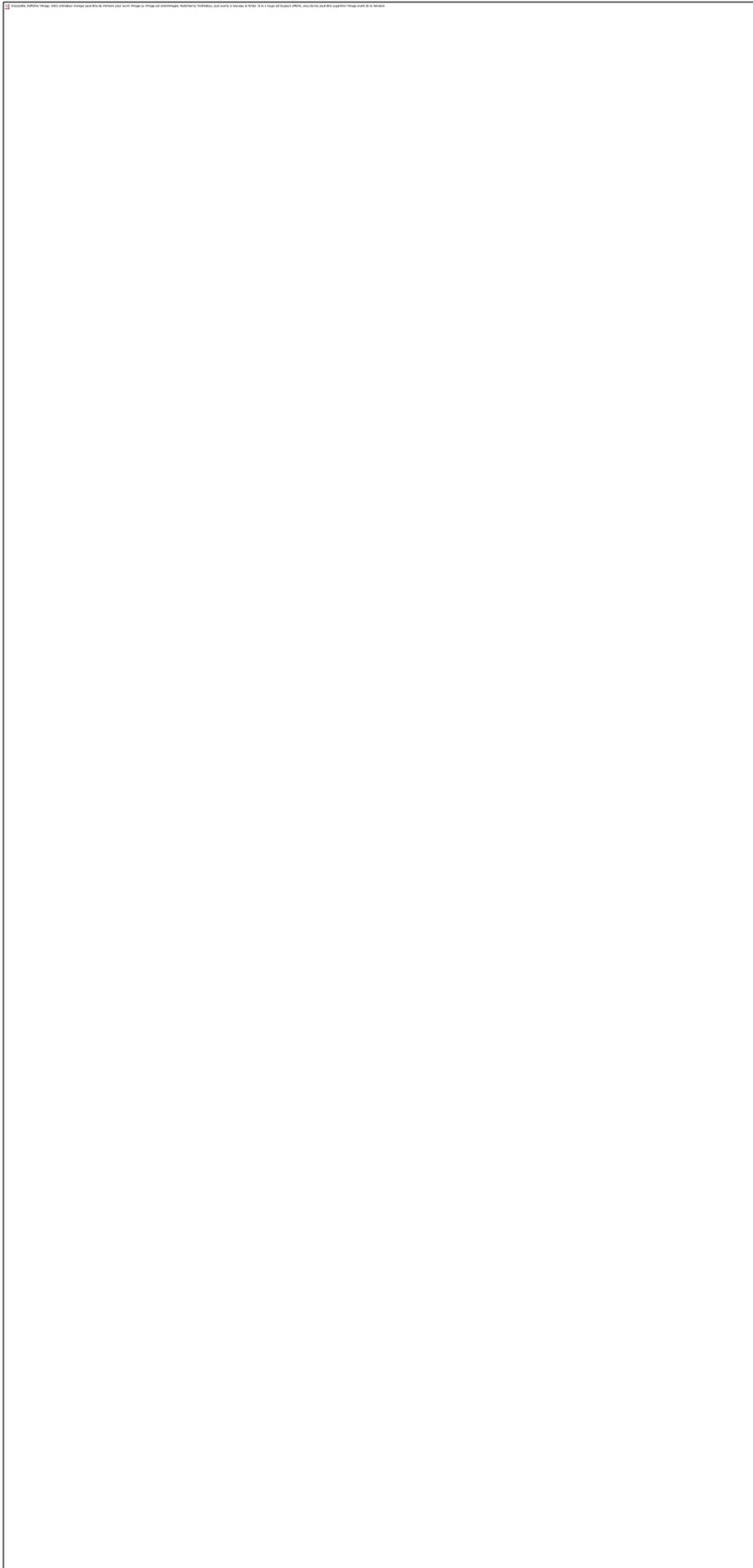


Figure 3.5 : Schéma de fonctionnement.

3.8. Implémentation de PID numérique sur l'arduino :

Le but de cette partie est justement d'essayer d'expliquer comment implémenter une régulation PID sans entrer dans les calculs ni sans utiliser les fonctions de transfert du système et autres équations plus compliqué. [43]

3.8.1. Le régulateur proportionnel (P : première règle) :

La commande de ce régulateur est proportionnelle à l'erreur.

$$\text{commande} = K_p * \text{erreur}$$

K_p est le coefficient de proportionnalité de l'erreur à régler de façon manuelle.

3.8.2. Le régulateur proportionnel intégral (PI : première et seconde règle) :

La commande de ce régulateur est proportionnelle à l'erreur, mais aussi proportionnelle à l'intégrale de l'erreur. On rajoute donc à la commande générée par le régulateur proportionnel, la somme des erreurs commises au cours du temps.

$$\text{commande} = K_p * \text{erreur} + K_i * \text{somme_erreurs}$$

K_i est le coefficient de proportionnalité de la somme des erreurs. Il faut aussi le régler de façon manuelle.

3.8.3. Le régulateur proportionnel dérivé (PD : première et troisième règle) :

La commande de ce régulateur est proportionnelle à l'erreur, mais aussi proportionnelle à la dérivée de l'erreur. La dérivée de l'erreur correspond à la variation de l'erreur d'un échantillon à l'autre et se calcule simplement en faisant la différence entre l'erreur courante et l'erreur précédente (c'est une approximation linéaire et locale de la dérivée).

$$\text{commande} = K_p * \text{erreur} + K_d * (\text{erreur} - \text{erreur_précédente})$$

K_d est le coefficient de proportionnalité de la variation de l'erreur. Il faut régler ce coefficient expérimentalement.

3.8.4. Le régulateur proportionnel intégral dérivé (PID : première, seconde et troisième règle) :

La commande est à la fois proportionnelle à l'erreur, proportionnelle à la somme des erreurs et proportionnelle à la variation de l'erreur.

$$\text{commande} = K_p * \text{erreur} + K_i * \text{somme_erreurs} + K_d * (\text{erreur} - \text{erreur_précédente})$$

Nous allons donc faire une mesure sur notre système pour pouvoir calculer l'erreur et ainsi appliquer le PID. Cette mesure est à faire régulièrement à une certaine fréquence d'échantillonnage.

Tous les x millisecondes, faire :

$$\text{erreur} = \text{consigne} - \text{mesure};$$

$$\text{somme_erreurs} += \text{erreur};$$

$$\text{variation_erreur} = \text{erreur} - \text{erreur_précédente};$$

$$\text{commande} = K_p * \text{erreur} + K_i * \text{somme_erreurs} + K_d * \text{variation_erreur};$$

$$\text{erreur_précédente} = \text{erreur}$$

3.8.5. Régler les coefficients d'un PID :[44]

Le réglage des coefficients K_p , K_i et K_d d'un PID peut se faire expérimentalement par essais/erreurs. Tout d'abord, sachons qu'il ne sert à rien de vouloir régler les trois coefficients en même temps. Il y a trop combinaisons possibles et trouver un triplet performant relèverait de l'exploit. Il vaut mieux y aller par étape.

Tout d'abord, il faut mettre en place un simple régulateur proportionnel (les coefficients K_i et K_d sont donc nuls). Par essais/erreurs, il faut régler le coefficient K_p afin d'améliorer le temps de réponse du système. C'est-à-dire qu'il faut trouver un K_p qui permette au système de se rapprocher très vite de la consigne tout en faisant attention de garder la stabilité du système : il ne faut pas que le système réponde très vite tout en oscillant beaucoup

Une fois ce coefficient réglé, on peut passer au coefficient K_i . Celui-là va permettre d'annuler l'erreur finale du système afin que celui-ci respecte exactement la consigne. Il faut donc régler K_i pour avoir une réponse exacte en peu de temps tout en essayant de minimiser les oscillations apportées par l'intégrateur.

CHAPITRE 03 MODELISATION ET EQUATION DYNAMIQUE

Enfin, on peut passer au dernier coefficient K_d qui permet de rendre le système plus stable. Son réglage permet donc de diminuer les oscillations.

En général, pour régler ces coefficients, on donne au système une consigne fixe (exemple : l'évolution du nombre de tours par seconde du moteur au cours du temps). Le graphe résultant possède donc cette forme :

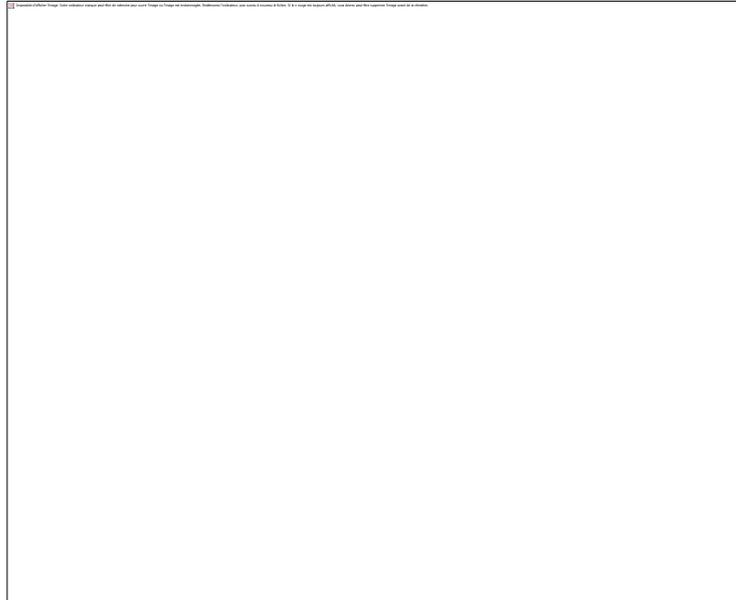


Figure 3.6 : Courbe pour régler le PID. [19]

Le PID parfait n'existe pas, tout est une question de compromis. Certaines applications autoriseront un dépassement afin d'améliorer le temps de stabilisation, alors que d'autres ne l'autoriseront pas (exemple, contrôler un stylo pour écrire sur une feuille. S'il y a dépassement dans le PID, le stylo traversera la feuille). Tout dépend donc du cahier des charges. Chacun des coefficients a un rôle à jouer sur la réponse à une consigne :

L'erreur statique, c'est l'erreur finale une fois que le système est stabilisé. Cette erreur doit être nulle. Pour diminuer l'erreur statique, il faut augmenter K_p et K_i .

Le dépassement, c'est le rapport entre le premier pic et la consigne. Ce dépassement diminue si K_p ou K_i diminuent ou si K_d augmente.

Le temps de montée est l'intervalle de temps correspondant à la croissance du signal de 10 % à 90 % de sa variation totale.

Le temps de montée diminue si K_p ou K_i augmentent ou si K_d diminue.

CHAPITRE 03 MODELISATION ET EQUATION DYNAMIQUE

Le temps de stabilisation, c'est le temps qu'il faut pour que le signal commette une erreur inférieure à 5% de la consigne. Ce temps de stabilisation diminue quand K_p et K_i augmentent.

3.9. Conclusion :

Dans ce chapitre nous avons vu le principe de fonctionnement de robot auto balance et les solutions pour les calculs et les équations dynamiques mais nous pour notre projet nous utilisons une méthode plus facile pour le faire fonctionner avec une implémentation de régulation PID sans faire des calculs compliqués.

Chapitre 04:

CONCEPTION ET ASSEMBLAGE DU CHASSIS

1.1. Introduction :

Les systèmes embarqués sont composés d'une partie logiciel « Software » et une partie « Hardware ». Dans le chapitre précédent, on s'est intéressés à la modélisation de notre robot. Dans ce chapitre, nous allons discuter de la partie hardware (les étapes pour réaliser le robot) et la partie software (à base de L'Arduino). En outre, nous allons donner les différentes solutions aux contraintes rencontrées au cours de ce projet.

Pour que le robot reste en équilibre, les moteurs doivent contrecarrer la chute du robot. Cette action nécessite une rétroaction et des éléments de correction. L'élément de rétroaction est un gyroscope plus un accéléromètre. Ces deux derniers sont incorporés dans un même boîtier à savoir le (MPU6050), qui fournit à la fois une accélération et une rotation dans les trois axes (bases du MPU6050 I2C). L'Arduino utilise ce signal pour connaître l'orientation actuelle du robot. L'élément correcteur est la combinaison moteur / roue.

Donc la première étape on va concevoir un montage afin de lire la valeur générée par le circuit mpu6050.

1.2. Mise en place et fonctionnement du mpu6050 :

Pour mesurer, une inclinaison, une accélération, le MPU6050 est la meilleure solution. Sa mise en place est facile, un composant facilement implantable dans tout projet sous Arduino, Raspberry ou pic. Il nous permettra de déterminer une inclinaison dans les trois axes, déterminer une accélération, par une première intégration de l'accélération en fonction du temps, il pourra donner la vitesse, par une seconde intégration de l'accélération en fonction du temps, il donnera une distance

Ce module communique avec la carte arduino via un bus de communication I2C, une brève introduction est donnée dans la section suivante.

1.3. Bus I2C :

I2C est un bus de données qui a émergé de la « guerre des standards » lancée par les acteurs du monde électronique. Conçu par Philips pour les applications de domotique et d'électronique domestique, il permet de relier facilement un microprocesseur et différents circuits, notamment ceux d'une télévision moderne : récepteur de la télécommande, réglages des amplificateurs basses fréquences, tuner, horloge, gestion de la prise péritel, etc.[45]

Il existe d'innombrables périphériques exploitant ce bus, il est même implémenté par logiciel

Dans n'importe quel microcontrôleur. Le poids de l'industrie de l'électronique grand public a permis des prix très bas grâce à ces nombreux composants. Ce bus porte parfois le nom de TWI (TwoWire Interface) chez certains constructeurs

I2C est un bus série synchrone bidirectionnel half-duplex. Plusieurs équipements, soit maîtres, soit esclaves, peuvent être connectés au bus. Les échanges ont toujours lieu entre un seul maître et un (ou tous les) esclave(s), toujours à l'initiative du maître (jamais de maître à maître ou d'esclave à esclave). Cependant, rien n'empêche un composant de passer du statut de maître à esclave et réciproquement.

La connexion est réalisée par l'intermédiaire de deux lignes :

- SDA (Serial Data Line) : ligne de données bidirectionnelle,
- SCL (Serial Clock Line) : ligne d'horloge de synchronisation bidirectionnelle.

Il ne faut également pas oublier la masse qui doit être commune aux équipements, les 2 lignes sont tirées au niveau de tension VDD à travers des résistances de pull-up (Rp).

Le nombre maximal d'équipements est limité par le nombre d'adresses disponibles, 7 bits pour l'adresse et un bit pour définir si on écrit ou on lit, soit 128 périphériques, mais il dépend également de la capacité (CB) du bus (dont dépend la vitesse maximale du bus). Il faut savoir que des adresses sont réservées pour diffuser des messages en broadcast et que de nombreuses adresses sont déjà attribuées par les fabricants ce qui limite grandement le nombre d'équipements (une variante d'adressage sur 10 bits existe également).[46]

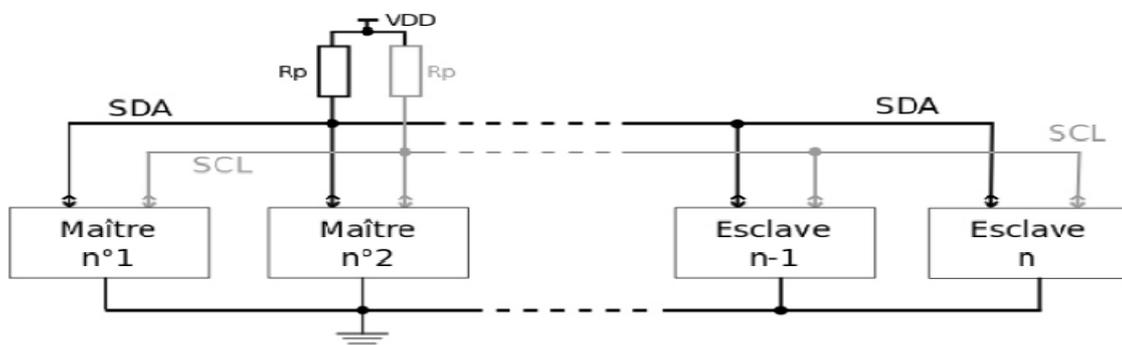


Figure 4.1 : Les lignes SDA et SCL. [20]

Pour l'instant on n'a qu'un capteur connecté en I2C à l'Arduino. On n'a donc pas besoin d'ajouter des résistances. On connecte donc directement A4 de l'Arduino à SDA (les données) et A5 à SCL (l'horloge).

Par ailleurs l'adresse par défaut d'un gyro MPU-6050 est **0x68**. On peut donc dire à l'arduino où il est puis, dans `setup()`, lancer la communication sachant qu'on a défini l'adresse du gyro par **`const int MPU_addr=0x68;`**

Ici notre Arduino nano est le « maître » et le gyroscope est « esclave ». L'Arduino écrira (`write`) pour envoyer des données au gyroscope, ou lira (`read`) pour en recevoir des données. Par exemple :

```
Wire.write(0); // set to zero (wakes up the MPU-6050)

AcX=Wire.read()<<8|Wire.read(); // 0x3B (ACCEL_XOUT_H) & 0x3C (ACCEL_XOUT_L)
```

Pour communiquer en suivant le protocole I2C, on a inclus la bibliothèque `ArduinoWire`, avec **`#include<Wire.h>`** .

- **comprendre la lecture du gyroscope**

C'est assez compliqué car il faut lire dans des registres de 16 bits les valeurs mesurées.

`int16_t AcX,AcY,AcZ,Tmp,GyX,GyY,GyZ;` définit que les variables mesurées sont codées sur 16 bits et sont signées (peuvent être négatives). Elles peuvent donc prendre une valeur allant de -32 768 à 32 767

On demande au gyroscope de commencer à fournir les données à partir du registre 0x3B qui contient une partie de l'accélération en x et on lui indique qu'on aura besoin de lire 14 registres en tout avec les lignes **`Wire.write(0x3B);`** et **`Wire.requestFrom(MPU_addr,14,true);`**

Ensuite on lit les registres deux par deux car chaque valeur que l'on recherche est codée sur deux registres différents. Par exemple l'accélération en x est codée dans 0x3B (ACCEL_XOUT_H) et 0x3C (ACCEL_XOUT_L).

Pour les lire et les transformer en une seule valeur, on utilise, des « bit-shift », ou déplacements de bits .

Le premier `Wire.read()` lit le premier registre : 0x3B puisqu'on a dit que c'est là qu'il faut commencer avec la commande `Wire.write(0x3B);` . Le deuxième lit le suivant (0x3C). L'accélération en x est calculée par la ligne `AcX=Wire.read()<<8|Wire.read();`

Chaque `Wire.read()` renvoie un entier de 16 bits et le suivant se décale d'un registre. La ligne `AcX=Wire.read()<<8|Wire.read();` place xxxxxxxxXXXXXXXX dans l'entier de 16 bits AcX par le mécanisme illustré dans cette image :

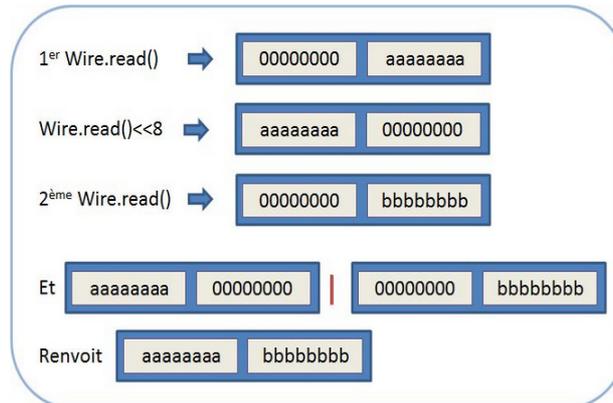


Figure 4.2 :Mécanisme des adresses. [21]

On ne s'intéresse qu'à l'accélération en x et y et l'angle en z. On ne va donc lire que AcX, AcY et GyZ du programme précédent .Pour ce là, on se positionne d'abord sur le registre 0x3B, on déclare qu'on ne veut lire que 4 registres avec les lignes suivantes :

```
Wire.write(0x3B); // starting with register 0x3B (ACCEL_XOUT_H)
Wire.endTransmission(false);
Wire.requestFrom(MPU6050, 4, true);
```

On lit AcX et AcY puis on passe à la lecture de GyZ, dont le premier registre est en 0x47, et on ne lit que les deux registres correspondants.

```
Wire.write(0x47);
Wire.endTransmission(false);
Wire.requestFrom(MPU6050, 2, true); // request a total of 2 registers
```

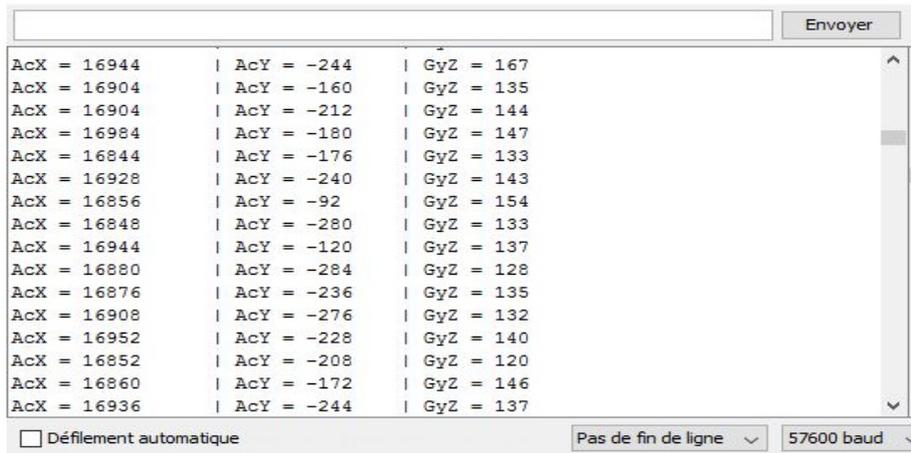


Figure 4.3 :Affichage des valeurs de gyroscope. [22]

4.4. Positionnement du gyroscope :

La figure 4.4 illustre un prototype de fonctionnement du gyroscope sur lequel on s'est inspiré pour l'intégrer dans notre projet. Ainsi, le gyroscope sera donc placé à l'horizontale, avec la flèche X pointant vers l'avant du robot et la flèche Y dans le même axe que les roues.[47]

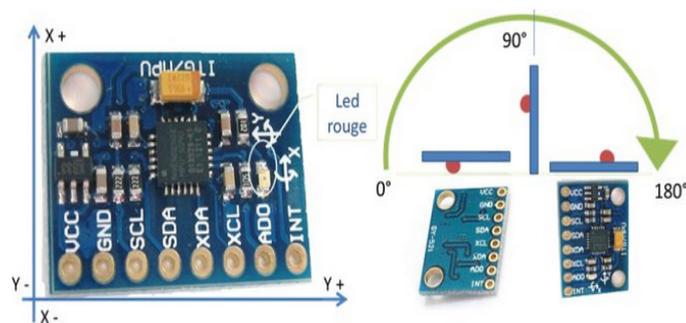


Figure 4.4 :Orientation de gyroscope. [23]

4.5. Modulation de largeur d'impulsion (PWM) :

Un signal de modulation de largeur d'impulsion (*Pulse Width Modulation* - PWM) est un signal à onde carrée pour lequel on contrôle la durée durant laquelle le signal reste à l'état haut. Cette portion de temps est nommée le cycle actif (*Duty Cycle*). La fréquence de ce signal doit être significativement plus élevée que la fréquence d'échantillonnage utilisée pour la commande numérique du système afin que le niveau de tension puisse être perçu comme une moyenne durant une période d'échantillonnage. De plus, lorsque ce type de signal est

utilisé pour contrôler un moteur, il est souhaitable d'utiliser une fréquence au-delà de la plage de fréquences audibles par l'oreille humaine afin d'éviter de produire des sifflements.[48]

Dans notre projet nous avons utilisé deux moteurs à courant continu et nous avons besoin de varier ses vitesses et le sens de rotation. Ceci dit, nous avons utilisé comme solution le circuit de référence L298N. Ce circuit offre un moyen de piloter deux moteurs CC ou un moteur pas à pas ou même un moteur brushless. Il peut délivrer jusqu'à 2A en pointe et 20W en continu. Il possède son propre circuit d'alimentation logique (permettant d'alimenter l'Arduino, ses spécifications et ses avantages sont :

- Composant de contrôle en puissance: L298N
- Alimentation de la charge: de +6V à +35V
- Tension de commande logique Vss: de +5 à +7V (alimentation interne de +5V)
- Courant de commande logique: de 0 à 36mA
- Tensions pour contrôle du sens: Low -0.3V~1.5V, high: 2.3V~Vss
- Tensions pour contrôle "Enable": Low -0.3V~1.5V, high: 2.3V~Vss
- Puissance Max: 25W (Température 75°C)
- Température de fonctionnement: de -25°C à +130°C
- Dimensions: 60mm x 54mm
- Poids: ~48g

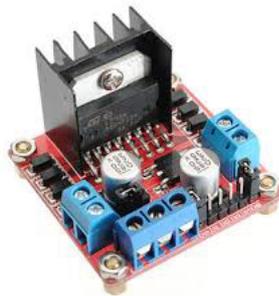


Figure 4.5 : Pilot moteur L298n. [23]

4.5.1. Activation moteur : [49]

- ENA raccordés à un niveau haut (HIGH) activera MOTORA.
- ENB raccordés à un niveau haut (HIGH) activera MOTORB.

Pour le contrôler la vitesse, connecter ENA(ENB) est connectée sur une sortie PWM.

4.5.2. Rotation Moteur : [50]

- IN1 raccordés à 5V et IN2 à GND MOTORA tournera dans le sens direct de l'horloge.
- IN1 raccordés à GND et IN2 à 5V MOTORA tournera dans le sens inverse de l'horloge.

La partie software correspondant à cette connectivité est la suivante ;

```
int ENA=5; //Connecté à Arduino pin 5(sortie pwm)
int IN1=2; //Connecté à Arduino pin 2
pinMode(ENA,OUTPUT);//Configurer le A broches comme sortie
digitalWrite(ENA,LOW);// Moteur A - Ne pas tourner (désactivation moteur)
analogWrite(ENA,255); // pour faire varier la vitesse de moteur
```

4.5.3. la librairie PID sur arduino :

Afin de faciliter l'intégration de la régulation PID dans notre robot, nous avons utilisé la bibliothèque PID_V1 compatible avec la carte Arduino. Pour installer la bibliothèque il suffit de suivre les étapes suivantes:

Dans l'IDE Arduino, nous avons le menu Croquis/Importer bibliothèque/Ajouter la bibliothèque.ZIP. Pour utiliser cette bibliothèque, au début du programme en écrit :
`include<PID.V1.h`

Les trois correcteurs k_p , k_i et k_d peuvent être déterminés mathématiquement ou expérimentalement.

Pour notre cas, nous avons essayé de déterminer quelles valeurs seraient les plus efficaces. Nous nous sommes inspirés d'un projet similaire, où nous avons donc tenté de trouver ces valeurs avec l'utilisation de trois potentiomètres. Les valeurs qu'ils renvoyaient étaient attribuées à celles des constantes recherchées. Les tests effectués devaient ainsi nous permettre de déterminer à quel moment notre robot était le plus stable, rapide et précis.

Pour conclure cette méthode, récapitulons les effets importants de chaque correcteur :

- k_p et k_i augmentent le temps de stabilisation

- ki diminue considérablement l'erreur statique
- kd diminue le temps de stabilisation et diminue les risques que la mesure dépasse la consigne.[51]

4.6. Hardware :

4.6.1. Réalisation du châssis :

Pour la réalisation de châssis, notre première tentative était de le réaliser avec le plexiglas, mais les essais expérimentales n'ont pas été concluantes et le robot n'arrive pas à avoir son équilibre à cause du poids qui influence sur la rapidité de déplacement. Face à cette contrainte, nous avons décidé de minimiser au maximum le poids, en optant d'une part au minimum plexiglas quand c'est nécessaire et d'utiliser des matériaux plus légers.

Etape 1 : les composants à utiliser dans la réalisation:

- ❖ Une feuille de plexiglass sur laquelle nous allons installer nos matériaux
- ❖ Des vis et Des écrous pour relier les plates-formes
- ❖ Des fils de raccordement
- ❖ 2x moteurs DC avec un réducteur motorisation 300 tr/min 6V
- ❖ 1x L298N H-bridge
- ❖ 1x Arduino Nano
- ❖ 1x gyroscope-module (MPU-6050)
- ❖ 3x batteries 3.7v
- ❖ 1x interrupteur
- ❖ 2 roues 65 mm

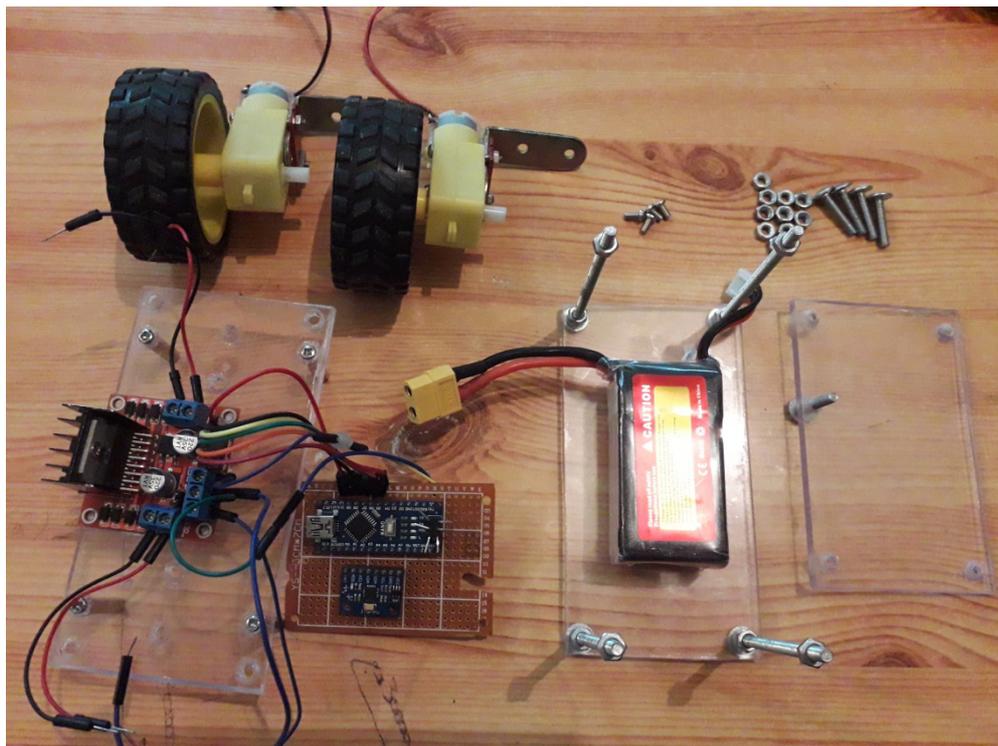


Figure 4.6 : Les composants utilisés.

Etape 2 : Le montage :

La figure suivante illustre le câblage des composants électrique en utilisant le Logiciel Fritzing..

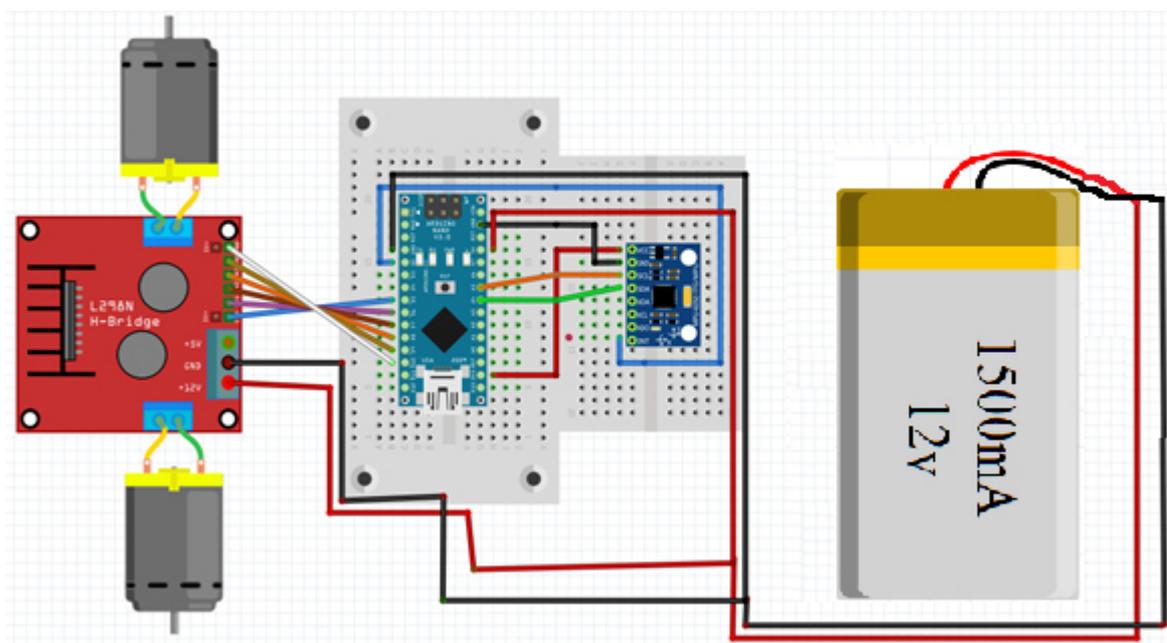


Figure 4.7 : Montage réalisé

Etape 3 : Raccordement des composants:

L'une des contraintes du montage des composants du robot est la répartition du poids autour de l'axe verticale, c'est-à-dire que le centre de gravité doit être superposé à cet axe. En outre le centre de gravité doit être le plus bas possible.

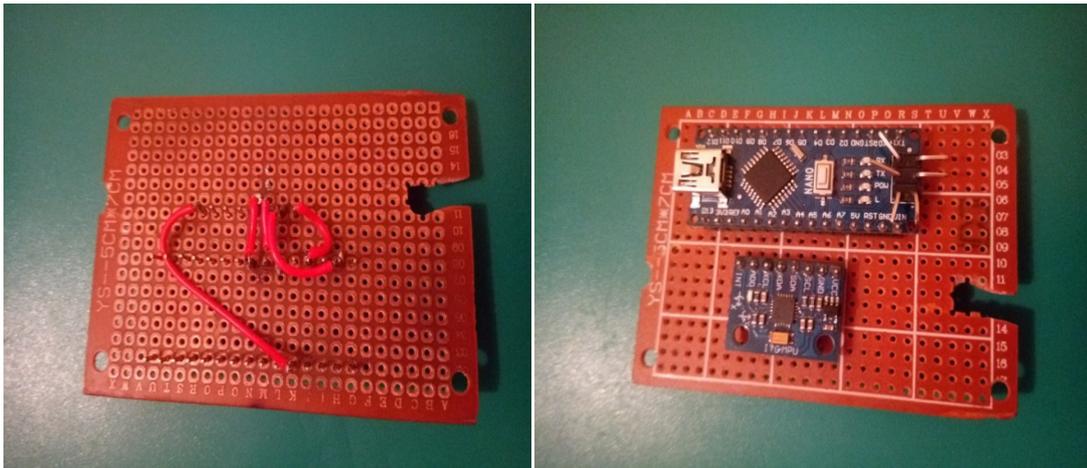


Figure 4.8 : Raccordement du module MPU6050 avec l'arduino.

Etape 4 : Positionnement du gyroscope :

Pour avoir une bonne mesure d'angle d'inclinaison du châssis il faut que le gyroscope soit posé de façon plate au centre de la rotation des roues et au plus proche des roues. A cet effet, et pour toutes ces considérations nous avons réservé le premier étage pour le gyroscope et la carte Arduino. Pour faire régler l'horizontalité de cette carte qui contient le gyroscope on a ajouté une visse afin de la positionner

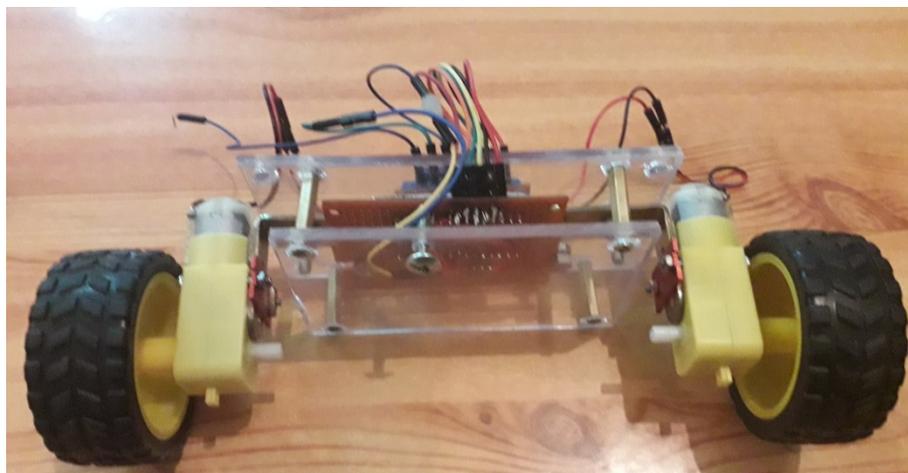


Figure 4.9 : Placement du gyroscope sur le châssis

Toujours dans le souci du centre de gravité qui doit être sur l'axe, la batterie doit être placée dans l'étage supérieur à celui contenant le gyroscope et la carte arduino. La figure suivante illustre le placement de la batterie.



Figure 4.10 : Placement de la batterie dans un deuxième étage.

Etape 5 : Montage du corps du robot:

- Les deux étages avec le châssis sont connectés via des vis et des écrous comme l'illustre la figure suivante.



Figure 4.11 : Corps du robot.

- connecter les fils du moteur C.C et les fils des batteries avec le driver moteur.
- Relier les fils entre Arduino Nano et le module MPU-6050.
- Rassembler les fils de driver moteur et l'Arduino Nano.

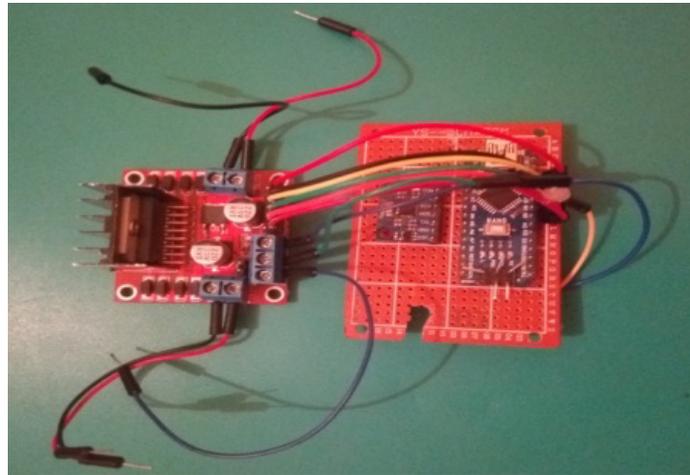


Figure 4.12 : Partie électronique

La structure finale du projet :

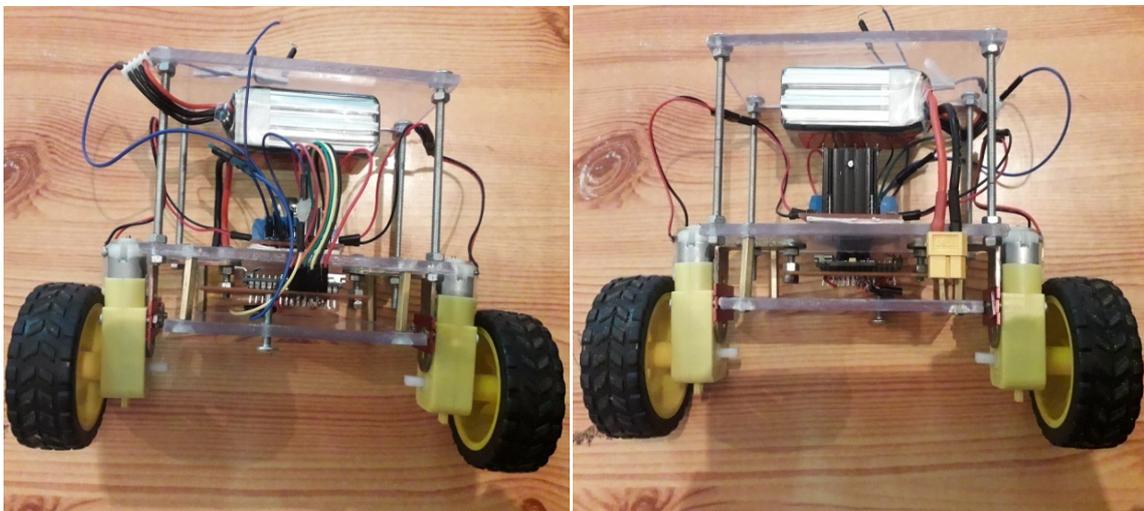


Figure 4.13 : Version finale du robot

La contrainte de l'alimentation réside dans la recharge de la batterie. Au départ nous avons utilisé trois batteries lithium en séries mais on a trouvé un problème de charge, pour les charger il faut les alimenter avec une source de tension de 12 volts. Mais le fait que ces batteries sont en séries pose le problème de recharge où chaque batterie nécessite un contrôleur de charge indépendant. Pour trouver une solution à cette contrainte, nos recherches nous ont

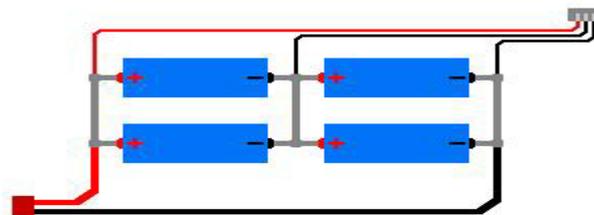
menés vers la batterie LIPO, pour « Lithium Polymer Batterie » qui est très utilisés dans le domaine de la robotique, les Smartphones, et les Ordinateurs portables et les tablettes.

4.6.2. Constitution d'une batterie Lipo:



Figure 4.14 :Constitution d'un batterie LiPo.

- Le taux de décharge: il est ici de 50C, ce qui veut dire que cet accumulateur peut donner 50 fois sa capacité (C). En somme, on obtient un taux de décharge de $50 \times 5600 = 280$ A en continu.
- L'ampérage que la batterie peut fournir en continu et en pointe: c'est en fait la retranscription du calcul effectué plus haut. Ainsi, l'accu peut donner 280A en continu et 560 en pointe (on considère qu'une pointe dure pendant un maximum de 2 secondes...)
- La capacité: c'est simplement la "taille" de la batterie. Plus la capacité est importante plus la batterie va durer longtemps lors de son utilisation.
- Enfin, la constitution interne: il est ici question d'une batterie 2S2P. C'est à dire qu'il y a 2 éléments en série (cumul des tensions) et deux éléments en parallèle.



2 éléments en série et 2 éléments en parallèle

Figure 4.15 :Constitution interne de la batterie.

Le schéma ci-dessus illustre la constitution interne de la batterie. Nous avons donc deux séries de deux éléments de 2600 mAh de tension 7,4V qui sont mis en parallèle et on obtient donc deux séries d'une capacité de 5200 mAh et de tension de 7,4V. En mettant en série ces deux séries de 5200 mAh 7,4V on va obtenir un voltage général de l'accumulateur de 7,4V et de capacité de 5200 mAh. Donc 2 éléments en série composés de séries de deux éléments en parallèle

Pour un accumulateur en 3S1P on aura ainsi:

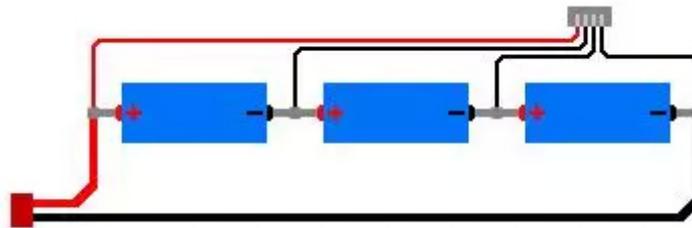


Figure 4.16 :Schéma de l'équilibrage.

Pour ce qui est du câblage interne de la fiche d'équilibrage. On a, plus haut, expliqué que l'équilibrage tendait à égaliser les voltages entre les différentes cellules qui forment l'accumulateur. Sur le schéma on peut observer le schéma de l'équilibrage. Il y a un fil rouge qui se retrouve au pôle + de la batterie et un fil noir (qui n'est pas la borne -) qui se retrouve entre chaque connexion entre deux cellules et cela permettra (à votre chargeur, un testeur de batterie ou vous) de connaître le voltage de chaque cellule en devenant soit un pôle + soit un pôle -.

Ce schéma nous permettra de mieux comprendre comment mesurer la tension de chaque élément de votre batterie.[7]

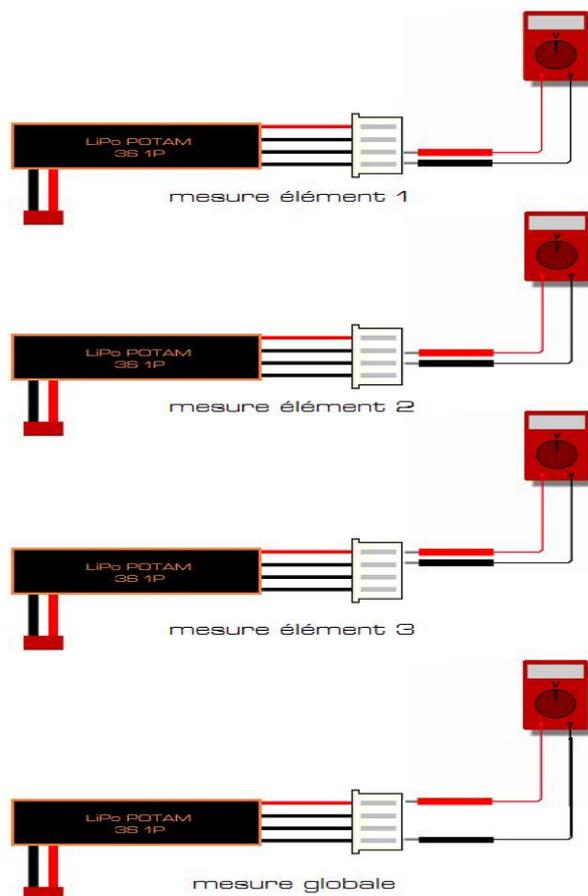


Figure 4.17 : Mesure de la tension de batterie.

Par sécurité nous recommandons l'usage de chargeur de batterie LiPo simple, qui charge uniquement via la prise d'équilibrage des batteries. Certes ils chargent plus lentement que d'autres, mais il n'y a aucun réglage à faire sur le chargeur et donc pas de risque d'erreur, et la charge se fait en toute sécurité.

La plupart des chargeurs de ce type ont des leds indicatives qui sont rouges lorsque la batterie est en charge/équilibre. Ces leds deviennent vertes une fois que la batterie est prête à l'emploi.



Figure 4.18 :LiPo/ LiFe balance charger.

4.7. Software :

Etape 1 :chargement du programme sur la carte arduino Nano. IDE Arduino est le logiciel de programmation de notre robot pendule inversé

la 1^{ère} fenêtre qui s’affiche quand on ouvre le logiciel

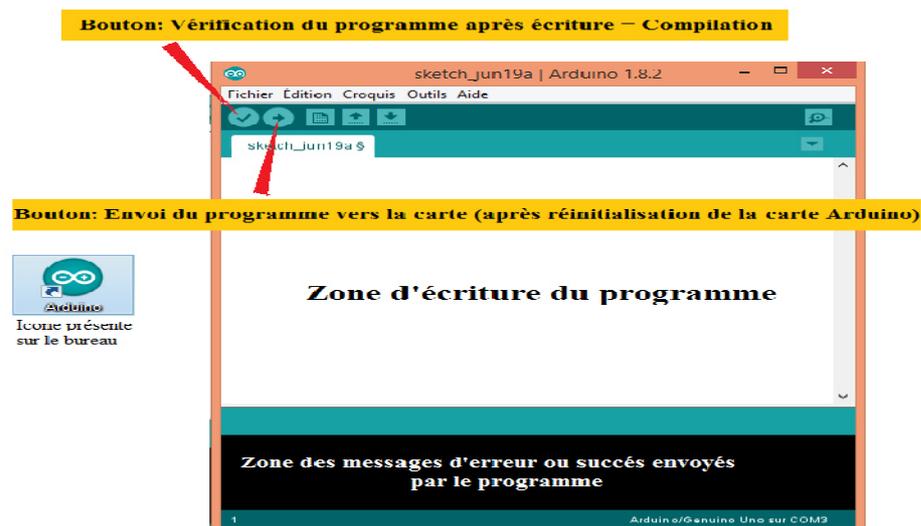


Figure 4.19 :IDE arduino.

Un code source est une suite d’instructions élémentaires sous forme textuelle, ligne par ligne. La carte lit puis effectue les instructions les unes après les autres, dans l’ordre

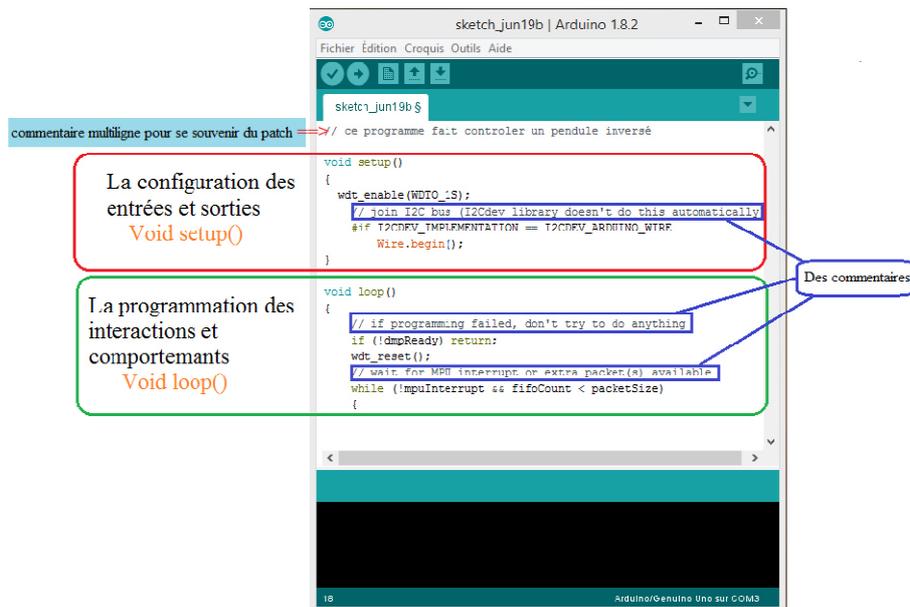


Figure 4.20 : Structure d'un programme Arduino

Une fois la dernière ligne exécutée, la carte revient au début de la troisième phase et recommence sa lecture et son exécution des instructions successives. Et ainsi de suite.

Cette boucle se déroule des milliers de fois par seconde et anime la carte.

Etape 2:

- Fermer l'interrupteur.
- Connecter l'arduino au PC via un câble USB.
- Ouvrir le code source de robot auto balance.
- Vérifier et compiler le code source.
- Attendre quelques secondes et débranché le câble USB.
- Ouvrir l'interrupteur.



Figure 4.21 : Télécharger le programme vers l'Arduino.

4.8. Conclusion :

Nous avons présenté dans ce chapitre les étapes de réalisation du robot auto balance. Dans ce chapitre nous avons également relevé toutes les contraintes rencontrées ainsi que les solutions proposées.

Conclusion générale

Ce projet nous a permis d'aborder les aspects d'implantation expérimentale des techniques que nous avons étudiées durant notre formation théorique. L'équilibrage d'un robot autonome à deux-roues de type pendule inverse, n'est pas chose facile. Si les simulations permettent aisément de réaliser beaucoup d'études théoriques, construire un prototype et le contrôler est une autre paire de manche.

Une stratégie de contrôle a été mise en œuvre pour résoudre le problème du contrôle autour de l'équilibre du système. Pendant le test, le robot est presque capable de maintenir sa position verticale en se balançant sur ses roues.

La technique du filtre complémentaire a été appliquée avec succès. La dérive des valeur de gyromètre a été effectivement éliminée permettant une estimation relativement correcte de l'angle d'inclinaison et de sa dérivée.

Plus de recherches sont nécessaires pour étudier les effets de la linéarisation de la dynamique du système afin d'améliorer la stabilité et la robustesse du robot. Une tentative de contrôler le système en utilisant des méthodes non-linéaires est fortement recommandé pour des recherches futures. De nouveaux composants, très compacts et complets ont vu le jour, comme les IMU (Inertia Measurement Units) MEMS et pourront avantageusement remplacer des composants discrets, mal positionnées sur le corps du robot, afin d'augmenter la précision de la mesure de l'angle et des vitesses angulaires et linéaires.

Références Bibliographiques

- [1] : www.cite-sciences.fr/fr/ressources/bibliotheque-en-ligne/dossiers-documentaires/des-robots-pour-tout-des-robots-pour-tous/applications/ consulté le :14/06/2018 a 6:20 PM
- [2] : <https://sites.google.com/site/typesderobots/home/types-de-robots> consulté le :16/06/2018 a 9:50 PM
- [3]: <https://assistancerobotisee.wordpress.com/domaines-dapplications/> consultée le : 17/06/2018 a 5:10 PM
- [4]: <https://www.roboticbeast.com/les-differents-types-de-robot-1/> consulté le :20/06/2018 a 6: 20 PM
- [5] : V. Sukontanakarn and M. Parnichkun.Real-time optimal control for rotary inverted pendulum. American Journal of Applied Sciences, (6) :1106–1115, 2009.
- [6] : http://eavr.u-strasbg.fr/~bernard/education/3a_robmob/3a_robmob_slides.pdf consulté le :27/06/2018 a 11:40 PM
- [7] : <https://www.techno-science.net/definition/9026.html> consulté le :28/06/2018 a 9:50 AM
- [8] : TOUATI Billal ,MOUSSAOUI Mahdi,<< Commande d'un pendule inversé par mode glissant >>, Master en Automatique, Université Abderrahmane Mira – Bejaia,2017
- [9] : H. Ludvigsen A. Shiriaev, A. Pogromsky and O. Egeland.On global properties of passivity based control of an inverted pendulum. International Journal Of Robust And Nonlinear Control, (10) :283–300, 2000
- [10]: Khaoula BRAHIM,<< Analyse et modélisation : du pendule inversé sur deux roues au fauteuil roulant automatisé FreeMove>>, Rapport de Stage de Master 2 Recherche EEA,2014
- [11]:Samer RIACHY<<Contribution à l'estimation et à la commande de systèmes mécaniques sous-actionnés>>, pour obtenir : Docteur de l'Ecole Centrale de Lille Discipline Automatique et Informatique Industrielle,01 décembre 2008.
- [12]: K.H. Lundberg and J.K. Roberge.Classical dual-inverted-pendulum control. In The 2003 IEEE Conference On Decision And Control, 2003.
- [13]: D.ChoiandJ.H.Oh. Human-friendlymotioncontrolofawheeledinvertedpendulumbyreducedorder disturbance observer.In IEEE International Conference on Robotics and Automation, Pasadena, CA, USA, 2008.
- [14] : R.Olfati-Saber. Global stabilization of a flat underactuatedsystem : the inertia wheel pendulum. In 40th Conference on Decision and Control, Orlando,FL, 2001.
- [15] : H.Kwakernaak and R.Sivan. Linear optimal control systems.Number 575.Wiley Interscience, 1972.
- [16]: S. Marteau J. Lottin. Le pendule inversé :un système hybride in CETSIS'2005, Nancy, France, 2005.

Références Bibliographiques

- [17] : Noauthorslisted. Independence(tm)ibot(tm)3000mobilitysystem:astair-climbing wheel chair. In Issues EmergHealthTechnol, 2004.
- [1] [18]: Michele Portolan, « Electronique, Electrotechnique, Automatique, Télécommunications, Signal », Doctora le institut national polytechnique , Le 06 Décembre 2006.
- [19] : <https://www.techno-science.net/definition/6737.html> consulté le :02/07/2018 a 3:00 PM
- [20] : <http://www.ludilab.fr/> consulté le :04/07/2018 a 5:55PM
- [21] : <https://www.automation-sense.com/blog/electronique/tutoriel-arduino.html> consulté le :07/07/2018 a 08:30 AM
- [5] [22] : <https://zestedesavoir.com/tutoriels/686/arduino-premiers-pas-en-informatique-embarquee/> consulté le :14/07/2018 a 7:45 PM
- [23] : <http://blewando.dlinkddns.com/elv/Promo2016/th8/pag2.html> consulté le :13/07/2018 a 10:00 PM
- [24] : <http://dictionnaire.sensagent.leparisien.fr/Arduino/fr-fr/> consulté le :05/07/2018 a 10:00 AM
- [25] : <https://www.tunisia-sat.com/forums/threads/3299045/> consulté le :09/07/2018 a 4:27 AM
- [26] : <https://www.gotronic.fr/> consulté le :16/07/2018 a 11:37 AM
- [27] : <http://projet.eu.org/pedago/sin/tutos/arduino.pdf> consulté le :18/07/2018 a 01:22 AM
- [28] : <https://www.infothema.fr/forum/index.php?topic=961.0> consulté le :19/07/2018 a 03:57 AM
- [29] : A. R. Hambley «*Electrical Engineering Principles and Applications*» Sixth edition Department of Electrical and Computer Engineering Michigan Technological University.
- [30] : https://fr.wikipedia.org/wiki/Machine_%C3%A0_courant_continu consulté le :23/07/2018 a 6:05 AM
- [31] : <https://www.avrfreaks.net/sites/default/files/Motor%20Driver.pdf> consulté le : 06/08/2018 a 3:10 PM
- [32] : <https://howtomechatronics.com/tutorials/arduino/arduino-dc-motor-control-tutorial-1298n-pwm-h-bridge/> consulté le :09/08/2018 a 5:30 PM
- [33] : Louzeau mathieu , << commande par supervision de systèmes mécatroniques via internet >> école polytechnique de montréal, septembre 2009 .
- [34] : <https://www.invensense.com/products/motion-tracking/6-axis/mpu-6050/> consulté le : 15/08/2018 a 7:50 PM

Références Bibliographiques

- [35] : <https://docplayer.fr/58522250-Le-pendule-inverse-stabilise-par-volant-d-inertie-un-systeme-non-lineaire-sous-actionne.html> consulté le :21/08/2018 a 12:00 AM
- [36] : Cédric anthierens << conception d'un micro robot a actionneur asservi électropneumatique pour l'inspection intra tubulaire>> le grade de docteur, devant l'institut national des sciences appliquées de Lyon, 7 décembre 1999.
- [37] : https://www.pobot.org/IMG/pdf/Pendule_inverse_en_robotique.pdf consulté : 23/08/2018 a 01:15 PM
- [38] : <https://docplayer.fr/45325429-Sciences-industrielles-pour-l-ingenieur-partie-i-analyse-systeme.html> consulté le :28/06/2018 a 09:52 PM
- [39] : DELAMARE Fabrice « Robot à pendule inversé » Laboratoire d'Automatique, école polytechnique, Référence du projet : STPI/P6-3/2011 – 2012
- [40] : <https://www.developpez.net/forums/d1410658/general-developpement/programmation-systeme/embarque/implementation-langage-c-d-regulateur-pid/> consulté le :01/09/2018 a 02:00 AM
- [41] : http://zone.ni.com/reference/fr-XX/help/371361L-0114/lvpidmain/pid_vi_algs/ consulté le :02/09/2018 a 07:30 PM
- [42]: <http://www.specialautom.net/introduction.htm> consulté le :05/09/2018 a 8:49 AM
- [43] : <https://ieeexplore.ieee.org/document/6226963> consulté le :09/09/2018 a 01:50 AM
- [44] : ATMANI BILLAL << Optimisation des paramètres du Contrôleur PID par Algorithme Génétique Multi-objectifs>> , obtention du diplôme de Master en Automatique, Université Abderrahmane Mira de Béjaïa,Septembre 2011
- [45] : <https://la-programmation.surleweb-france.fr/arduino-utiliser-mpu6050/> consulté le : 16/09/2018 a 4:42 PM
- [46] : <https://docplayer.fr/42519577-Bus-i2c-classe-de-terminale-si.html> consulté le : 19/09/2018 a 2:35 AM
- [47] : <https://knowledge.parcours-performance.com/ajouter-gyroscope-a-robot-arduino/> consulté le :23/09/2018 a 8:40 PM
- [48] : <http://digital.ni.com/public.nsf/allkb/4291A7936721A53C86257EA10074E0A7> consulté le :29/09/2018 a 4:18 AM
- [49] : <http://wiki.t-o-f.info/Transducteur/MoteurCC> consulté le :02/10/2018 a 3:27 AM
- [50] : http://projet.eu.org/pedago/sin/term/6-asservissement_arduino.pdf consulté le : 04/10/2018 a 5:10 AM
- [51] : <http://blog.patrickmodelisme.com/post/qu-est-ce-qu-une-batterie-lipo> consulté le : 09/10/2018 a 4:40 AM

Références Bibliographiques

Référence des Figures :

- [1] : <http://www.hellopro.fr/robots-cartesiens-5-axes-5x-line-1001257-5749215-produit.html> consulté le :17/06/2018 a 6:50 PM
- [2] : <http://www.orion-industry.com/apollo/apollo-seiko-J-CAT-STELLAR.html>consulté le : 18/06/2018 a 8:40 PM
- [3] : <http://www.machine-outil.com/actualites/t490/a7357-fanuc-sepem.html> consulté le : 20/06/2018 a 04:36 PM
- [4] : <http://www.thefrog.com/robots/sankyo/index.html> consulté le :24/06/2018 a 07:13 PM
- [5] : <https://www.foundrymag.com/stub-229/gallery?slide=1> consulté le :27/06/2018 a 6:32 PM
- [6] : <http://www.directindustry.com/prod/kawasaki-robotics-gmbh/product-18836-585505.html> consulté le :01/07/2018 a 10:30 AM
- [7] : <https://www.exapro.fr/robot-industriel-abb-irb-6620-irc5-doccasion-p30212180/> consulté le :03/07/2018 a 11:20 PM
- [8] : V. Sukontanakarn and M. Parnichkun.Real-time optimal control for rotary inverted pendulum.American Journal of Applied Sciences, (6) :1106–1115, 2009.
- [9] : H. Ludvigsen A. Shiriaev, A. Pogromsky and O. Egeland.On global properties of passivitybased control of an inverted pendulum. International Journal Of Robust And Nonlinear Control, (10) :283–300, 2000
- [10] : Khaoula BRAHIM,<< Analyse et modélisation : du pendule inversé sur deux roues au fauteuil roulant automatisé FreeMove>>, Rapport de Stage de Master 2 Recherche EEA,2014
- [11] Samer RIACHY<<Contribution à l'estimation et à la commande de systèmes mécaniques sous-actionnés>>, pour obtenir : Docteur de l'Ecole Centrale de Lille Discipline Automatique et Informatique Industrielle,01 décembre 2008.
- [12] : K.H. Lundberg and J.K. Roberge.Classical dual-inverted-pendulum control. In The 2003 IEEE Conference On Decision And Control, 2003.
- [13] S. Marteau J. Lottin. Le pendule inversé :un système hybride? In CETSIS'2005, Nancy, France, 2005.
- [14] <https://www.elprocus.com/different-types-of-arduino-boards/> consulté le :26/08/2018 a 7:53 PM
- [15] A. R. Hambley «Electrical Engineering Principles and Applications» Sixth edition Department of Electrical and Computer Engineering Michigan Technological University.
- [16] https://fr.wikipedia.org/wiki/Machine_%C3%A0_courant_continu consulté le : 01/09/2018 a 2:47 AM

Références Bibliographiques

[17] <https://docplayer.fr/22075221-Introduction-sur-le-l298n.html> consulté le :05/09/2018 a 3:53 AM

[18] <https://www.elementsonline.com/mpu6050-gy-521-3-axis-analog-gyro-sensors-accelerometer-module> consulté le :07/09/2018 a 9:54 AM

[19] <https://learn.ni.com/teach/resources/123/pid-position-control> consulté le :13/09/2018 a 5:35 PM

[20] <https://slideplayer.fr/slide/1168015/> consulté le :19/09/2018 a 2:45 AM

[21] https://fr.gearbest.com/other-accessories/pp_226466.html consulté le :28/09/2018 a 1:23 AM

[22] <http://blog.patrickmodelisme.com/post/qu-est-ce-qu-une-batterie-lipo> consulté le :02/10/2018 a 00:06 AM

[23] <https://www.digikey.com/en/maker/blogs/2018/introduction-to-the-arduino-ide> consulté le :09/10/2018 a 3:49 PM

Résumé :

Dans ce travail on a construit un robot à auto-équilibre à deux roues. Le but de ce projet est de l'équilibrer en position verticale.

Le robot d'équilibre à deux roues est un problème de type pendule inversé. Pour garder le robot en équilibre, il doit mesurer l'angle d'inclinaison et contrôler les roues pour ajuster le robot en position verticale. Dans notre cas, nous utilisons l'accéléromètre pour mesurer l'angle d'inclinaison et le gyromètre pour compléter les lectures de l'accéléromètre. Les deux roues sont connectées avec deux moteurs à CC contrôlé par le module L298N.

Abstract:

In this work we built a two-wheel self-balancing robot. The goal of this project is to balance it in a vertical position.

The balancing robot with two wheels is a problem of the inverted pendulum type. To keep the robot in balance, it must measure the tilt angle and control the wheels to adjust the robot to the upright position. In our case, we use the accelerometer to measure the angle of inclination and the gyrometre to complete the readings of the accelerometer. Both wheels are connected with two DC motors that is controlled by the L298N module.

ملخص :

في هذا العمل ، قمنا ببناء روبوت ذاتي التوازن ثنائي العجلات. الهدف من هذا المشروع هو تحقيق التوازن في الوضع الرأسي. إن روبوت الإلتزان مع عجلتين هو مشكلة في نوع البندول المقلوب. للحفاظ على التوازن في الروبوت ، يجب قياس زاوية الميل والتحكم في العجلات لضبط الروبوت إلى الوضع المستقيم. في حالتنا ، نحن نستخدم مقياس التسارع لقياس زاوية الميل و gyromètre لإكمال قراءات مقياس التسارع. ترتبط كلتا العجلتين بمحركين DC يتم التحكم فيهما بواسطة وحدة L298N.