

الجمهورية الجزائرية الديمقراطية الشعبية
République Algérienne démocratique et populaire

وزارة التعليم العالي والبحث العلمي
Ministère de l'enseignement supérieur et de la recherche scientifique

جامعة سعد دحلب البليدة
Université SAAD DAHLAB de BLIDA

كلية التكنولوجيا
Faculté de Technologie

قسم الإلكترونيك
Département d'Électronique



Mémoire de Master

Filière Automatique

Spécialité Automatique et Informatique Industrielle

présenté par

LACHEHEB Moundhir

Thème

Stabilisation d'un Robot Mobile à Deux roues par un contrôleur PID basé sur myRio FPGA.

Proposé par : Mr BENNILA Nor-Eddine

Année Universitaire 2019-2020

Remerciements

Nous tenons à remercier d'abord dieu le tout puissant qui nous a donné la force et la patience pour accomplir ce modeste travail. Aussi, nous tenons à exprimer nos vifs remerciements à notre promoteur Mr BENNILA Nor-Eddine pour ces conseils, sa disponibilité et la confiance qu'il nous a accordé. Nous tenons aussi à remercier les membres de jury d'avoir accepté de juger notre travail. Sans oublier aussi de remercier tous les membres de notre familles respective pour leur soutien et leurs encouragements, particulièrement nos chères parents En fin nous tenons également à remercier toutes les personnes qui ont participé de près ou de loin à la réalisation de ce travail.

ملخص

هذه الرسالة عبارة عن مشروع لنهاية الدراسة للماجستير يتكون من تثبيت روبوت متحرك بعجلتين ، استنادًا إلى من نفس الشركة المصنعة. سنركز LabVIEW ، باستخدام برنامج National Instrument من myRio FPGA على مجال الروبوتات وخاصة الروبوتات المتنقلة. ووضح وشرح مبدأ التشغيل لكل أداة ضرورية لتطوير المشروع. على أجهزة المصنع وبرامج المصنع. ونقوم بتطوير المعادلات اللازمة لحركة وتثبيت الروبوت وتنتهي بالإدراك العملي للروبوت.

كلمات المفاتيح: تحكم PID; روبوت متأرجح ; تحكم غير خطي ; البندول المقلوب.

Résumé

Ce projet rentre dans le cadre de la stabilisation d'un Robot Mobile à deux roues, à base de myRio FPGA de National Instrument, en utilisant le logiciel LabVIEW du même constructeur, le domaine concerné est le robotique et spécialement le robotique mobile. Ce projet avait aussi pour but l'exposition est l'explication du principe de fonctionnement de chaque outil. Ainsi que la description du côté hardware and software, pour ensuite appliquer le développement des équations nécessaire pour le déplacement et la stabilisation du robot et pour finir avec une réalisation pratique du robot

Mots clés : Commande PID; robot balançoire; Commande non-linéaire; pendule inversé

Abstract

This dissertation is a master's 2 end-of-study project and consists of stabilizing a two-wheeled Mobile Robot, based on myRio FPGA from National Instrument, using LabVIEW software from the same manufacturer. we are going to focus on the field of robotics and in particular mobile robotics. and expose and explain the operating principle of each tool necessary for the development of the project. On the plant hardware and the plant software. and we develop the equations necessary for the movement and stabilization of the robot and ending with the practical realization of the robot.

Keywords : PID control; balancing robot; Non-linear control; inverted pendulum .

Listes Symboles et les opérateurs

ERR : Erreur.

DIR : Direction.

AMP : Ampère.

X(t) : Position du chariot.

$\theta(t)$: L'angle du pendule.

I : Lagrange.

EM : Energie cinétique du chariot.

Ecm : Energie cinétique du pendule.

F : La force généralisée.

$\ddot{\theta}$: Accélération angulaire du pendule.

\ddot{x} : Accélération du chariot.

$\dot{\theta}$: Vitesse angulaire du pendule.

\dot{x} : La vitesse du chariot.

g : La constante de gravité.

ki , kd , kp : Les gains du contrôleur PID.

f(t): Les perturbation.

a(t) : La commande du système (l'accélération du chariot) .

h(s) : Fonction de transfert.

p1,p2: Pôles .

$\theta(s)$: La sortie du système.

Listes des abréviations et acronymes

- MCC** : Moteur courant continu.
- PID** : Contrôleur Dérivé Intégral Proportionnel.
- PWM** : La modulation de largeur d'impulsion peut générer une tension analogique à partir de la sortie numérique en commutant entre 0 et 1 avec une fréquence élevée et un facteur de marche spécifique.
- FPGA** : Field Programmable Gate Arrays.
- E/S**: Entrée et sortie.
- RIO**: Reconfigurable input output.
- AI** : Analog input.
- AO**: Analog output.
- DIO**: Digital input output .
- AC** : Current alternative.
- DC**: Direct current.
- LSB** : The least significant bit.
- UART**: Universal Asynchronous Receiver Transmitter.
- MSP** : Mini System Port.
- MXP** : Metropolitan Exchange Point.
- VI** : Version instrumentale.
- NI**: National instrument
- MO** : Méga octet.
- MHZ** : Méga hertz .
- HDL** : Hardware description language .
- CLL** : Long life capacitor.

EP : Energie potentiel.

EC : Energie cinétique.

Cc : Courant continue.

Table des matières

Introduction général	1
Chapitre 1 généralité	2
1.1 Introduction.....	3
1.2 Définitions.....	3
1.2.1 Définition d'un robot	3
1.2.2 Définition de la robotique.....	3
1.3 L'histoire des robots	4
1.4 Types de robots.....	5
1.4.1 Robots mobiles	5
1.4.2 Robots Humanoïdes	5
1.4.3 Robots domestiques	6
1.4.4 Robots industriels	6
1.4.5 Robots collaboratifs.....	7
1.5 Constituants d'un robot.....	7
1.5.1 Structure mécanique	8
1.5.2 Les actionneurs.....	9
1.5.3 Les Capteurs	9
1.5.4 Système contrôle commande	9
1.6 Robot mobile	10
1.7 L'architecture des robots mobiles.....	10
1.8 Les différentes types des robot mobiles.....	11
1.8.1 Robot unicycle	11
1.8.2 Robot tricycle	11
1.8.3 Robot voiture.....	12
1.8.4 Robot omnidirectionnel.....	12
1.9 Robot pendule inversé (uni-cycle).....	12
1.9.1 Définition.....	12

1.9.2	Les caractéristique	13
1.9.3	La structure mécanique	13
1.10	Le domaine d'application.....	14
1.10.1	En médecine	14
1.10.2	En robotique.....	15
1.11	Conclusion	16
Chapitre 2 Outils nécessaire pour la réalisation du projet (software et hardware.....		17
2.1	Introduction.....	18
2.2	L'unité de traitement (myRio-1900 FPGA).....	18
2.2.1	Définition	18
2.2.2	Présentation du matériel.....	20
2.2.3	Brochage des connecteurs	21
2.2.4	Canaux d'entrée analogiques.....	24
2.2.5	Canaux de sortie analogique.....	26
2.2.6	Accéléromètre	26
2.2.7	Conversion des valeurs de données brutes en tension	26
2.2.8	Lignes DIO.....	27
2.2.9	Lignes UART	29
2.2.10	Utilisation du Bouton 0	29
2.2.11	Comprendre les indications des LED	29
2.2.12	Utilisation du port hôte USB Le port hôte USB.	31
2.2.13	Caractéristiques	31
2.3	Moteur à courant continue	35
2.3.1	Définition du moteur courant continu	35
2.3.2	Les caractéristiques du moteur courant continu.	36
2.3.3	Les avantages / inconvénients du moteur courant continu	37
2.4	Driver MD13S : 13 AMP DC MOTOR DRIVER	38
2.4.1	INTRODUCTION	38
2.4.2	Les caractéristiques de MD13S	38
2.4.3	disposition et spécifications de la carte	39
2.4.4	Borner - Connectez au moteur et à la source d'alimentation.	40
2.4.5	Le brochage	42

2.5	Capteur MPU-9250	42
2.5.1	Présentation du produit	42
2.5.2	Caractéristiques.....	44
2.5.3	Fonctionnalités supplémentaires.....	45
2.5.4	Applications.....	46
2.6	Le logiciel LABVIEW	46
2.6.1	Définition	46
2.6.2	Les langages de programmation de labview	46
2.6.3	Partie de la virtuel instrumentale (vi)	47
2.6.4	DATA FLOW	47
2.6.5	Mesures d'accéléromètre avec le myrio-1900	47
2.7	Conclusion	53
Chapitre 3 Développement du projet		54
3.1	Introduction.....	55
3.2	Coordonnées généralisées du système:	55
3.2.1	Modélisation du système chariot-pendule à deux degrés de liberté.....	56
3.2.2	Étude et développement de régulateur.....	60
3.2.3	Les coefficients du régulateur PID	66
3.3	Auto-équilibrage	68
3.3.1	PID	59
3.3.2	Calibrage contrôleur PID	69
3.4	Le branchement des outils	70
3.5	La programmation	73
3.5.1	Le fonctionnement de mpu 9250	73
3.5.2	La lecture du capteur mpu 9250	75
3.5.3	Utilisation de l'accéléromètre	77
3.5.4	Utilisation de gyroscope	78
3.5.5	Utilisation des intégrateurs et dérivateurs	79
3.5.6	Le signal de commande	80
3.5.7	Modulation de largeur d'impulsion (PWM)	81
3.5.8	La direction de la Rotation des Moteur.....	82
3.5.9	Flat Séquence Structure	83
3.6	Conclusion	84
Conclusion générale		85
Bibliographie		86

Liste des figures

Figure 1.1	Robot mobile	5
Figure 1.2	Robot Humanoïde.....	5
Figure 1.3	Robot domestique	6
Figure 1.4	Robot industriel	6
Figure 1.5	Robot collaboratif	7
Figure 1.6	Structure constituants unité opérationnelle et fonctionnelle d'un robot	8
Figure 1.7	Structure mécanique d'un robot	9
Figure 1.8	Architecture d'un robot mobile	10
Figure 1.9	Robot de type unicycle	11
Figure 1.10	Robot de type tricycle	11
Figure 1.11	Robot de type voiture	12
Figure 1.12	Robot omnidirectionnel	12
Figure 1.13	Schéma de l'ensemble chariot et pendule inversé	13
Figure 1.14	Le corps de l'être humain vu comme un double pendule	14
Figure 1.15	Le BIPED et sa représentation 3D.....	15
Figure 2.1	NI MYRIO_1900 national instruments	18
Figure 2.2	Composants de NI MYRIO_1900	19
Figure 2.3	Schéma fonctionnel du matériel	20
Figure 2.4	Signaux primaires/secondaires sur les connecteurs MXP A et B	21
Figure 2.5	Signaux primaires/secondaires sur le connecteur MSP C	23
Figure 2.6	Circuit d'entrée analogique NI MYRIO_1900.....	25
Figure 2.7	Circuit de sortie analogique NI MYRIO_1900	26
Figure 2.8	Lignes DIO <13..0> sur le connecteur MXP A OU B	28
Figure 2.9	Lignes DIO <15..14> sur le connecteur MXP A OU B.....	28
Figure 2.10	Lignes DIO <7..0> sur le connecteur MSP C	28
Figure 2.11	Moteurs courant continu	35
Figure 2.12	Mode de fonctionnement d'un moteur DC	36
Figure 2.13	Moteur driver MD13S	38
Figure 2.14	Schéma structurelle de la carte	39
Figure 2.15	Le branchement de la carte avec le moteur et l'alimentation	42
Figure 2.16	La carte MPU_9250	43
Figure 2.17	Moniteur USB	48
Figure 2.18	Test de l'appareil	50
Figure 2.19	L'écran d'accueil de labview	51
Figure 2.20	Point de départ pour le projet	51
Figure 2.21	Configuration du projet	53

Figure 3.1	Schéma de l'ensemble chariot et pendule inversé.....	55
Figure 3.2	La représentation des pôles.....	61
Figure 3.3	La représentation d'un système en boucle ouverte.....	62
Figure 3.4	La représentation d'un système en boucle fermé.....	62
Figure 3.5	Représentation des pôles du système avec un retour proportionnel avec $k_1 < 0$	63
Figure 3.6	Représentation des pôles du système avec un retour proportionnel avec $k_1 > 0$	64
Figure 3.7	Représentation des pôles du système avec un retour dérivé avec $k_2 < 0$	65
Figure 3.8	Représentation des pôles du système avec un retour dérivé avec $k_2 > 0$	65
Figure 3.9	Représentation des pôles du système avec un retour dérivée et proportionnel avec, $k_1 > 0$ et $K_2 > 0$	66
Figure 3.10	Schéma du contrôleur PID [7].....	67
Figure 3.11	Le schéma électrique du système.....	70
Figure 3.12	L'avant du robot.....	71
Figure 3.13	L'arrière du robot	71
Figure 3.14	Utilisation de la bibliothèque i2c	74
Figure 3.15	L'adressage des données	75
Figure 3.16	L'application de l'équation sur le programme	77
Figure 3.17	L'affichage des valeurs de gyroscope..	77
Figure 3.18	Le robot	77
Figure 3.19	Exécution dans le Block Diagram	78
Figure 3.20	L'affichage dans le Front panel	78
Figure 3.21	La boucle control and simulation	79
Figure 3.22	Utilisation de dérivateur.....	79
Figure 3.23	Utilisation de l' intégrateur.....	80
Figure 3.24	Le régulateur pid	80
Figure 3.25	La bibliothèque PWM sur LABVIEW.....	81

Figure 3.26	Configuration de la bibliothèque.....	82
Figure 3.27	La bibliothèque DIO sur LABVIEW.....	82
Figure 3.28	La configuration de la bibliothèque DIO.....	83
Figure 3.29	La structure flat sequence utilisé dans le programme	83

Liste des tableaux

Tableau 1	Description des signaux sur les connecteurs MXP A et B.....	22
Tableau 2	Description des signaux sur le connecteur MSP C	23
Tableau 3	Description des signaux sur les connecteurs audio	24
Tableau 4	Indications du voyant d'état	30
Tableau 5	Définition des composants de la carte	40
Tableau 6	Description des broches	40
Tableau 7	Description des broches de commande	41
Tableau 8	Table de vérité pour la logique de commande	41
Tableau 9	Le branchement du driver 1 avec myrio-1900.....	72
Tableau 10	Le branchement du driver 2 avec myrio-1900.....	72
Tableau 11	Le branchement du capteur mpu9250 avec myrio-1900.....	72
Tableau 12	Le branchement des drivers avec les moteurs et la batterie.....	72

Introduction générale

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.

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 MPU9250) et comparer cet angle avec l'angle cible qui est 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 envoie l'information au driver moteur qui contrôle les deux roues.

La rédaction du document, est scindée en trois 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, nous allons exposer et expliquer le principe de fonctionnement de chaque outil nécessaire au développement du projet.
- Dans le troisième chapitre, nous allons mettre au clair les étapes de conception en partant par le développement des équations nécessaires pour le déplacement et la stabilisation du robot et en terminant par la réalisation pratique du robot en assemblant tous les éléments et outils cités dans le chapitre 2.

Chapitre 1 Généralités

1.1 Introduction

Dans ce chapitre nous allons explorer le domaine de la robotique et notamment la robotique mobile, où nous portons une importance particulière du fait de son lien étroit avec le projet que nous allons développer dans les chapitres qui suivent. Nous évoquons pour clôturer le chapitre les problèmes liés à la stabilisation des robots verticaux.

1.2 Définitions

La robotique est l'ensemble des techniques permettant la conception et la réalisation de machines automatiques ou de robots.

De nos jours, les robots se déclinent dans plusieurs secteurs. Les drones et les minirobots surpuissants émergents et les **robots intelligents** font leur apparition au sein des ménages. L'intelligence artificielle s'immisce dans le secteur de l'automobile avec la voiture autonome et se mêle aux robots pour nous proposer des **robots** [1].

1.2.1 Définition d'un robot

Appareil automatique capable de manipuler des objets ou d'exécuter des opérations selon un programme fixe, modifiable ou adaptable [1].

1.2.2 Définition de la robotique

L'ensemble des techniques permettant la conception et la réalisation de machines automatiques ou de robots [1].

1.3 L'histoire des robots

L'origine du mot robot provient de la langue tchèque dans laquelle sont ancêtre "**robota**" signifie **travail forcé**.

Le terme de robotique quant à lui est apparu en 1942 dans le cycle universellement connu rédigé par *Isaac Asimov* et intitulé "**Les robots**". Voici une définition concise de ce que peut être un robot :

" Machine programmable qui imite des actions d'une créature intelligente."

Au cours de l'histoire on peut distinguer 3 types de robots correspondant en quelques sortes à l'évolution de cette "espèce" créée par l'Homme.

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

Le **second type** de robot correspond à ceux qui sont équipés de capteurs. On trouve des capteurs de température, photo-électrique, à ultrasons pour par exemple éviter les obstacles et/ou suivre une trajectoire. Ces capteurs vont permettre au robot une relative adaptation à son environnement afin de prendre en compte des paramètres aléatoires qui n'aurait pu être envisagés lors de leur programmation initiale. Ces robots sont donc bien plus autonomes que les automates mais nécessitent un investissement en temps de conception et en argent plus conséquent.

Enfin le **dernier type** de robot existant correspond à ceux disposant d'une intelligence dite "artificielle" et reposant sur des modèles mathématiques complexes tels que les réseaux de neurones. En plus de capteurs physiques comme leurs prédécesseurs, ces robots peuvent prendre des décisions beaucoup plus complexes et s'appuient également sur un apprentissage de leur erreur. Bien sûr il faudra attendre encore longtemps avant que le plus "intelligent" des robots ne soit égal, tant par sa faculté d'adaptation que par sa prise de décisions, à l'Homme [2].

1.4 Types de robots

Il y'a plusieurs types de robots : robots mobiles, robots Humanoïdes, Robots domestiques, Robots industriels, Robots collaboratifs :

1.4.1 Robots mobiles

Robots capables de se déplacer dans un environnement. Ils sont équipés ou non de manipulateurs suivant leur utilisation (**Figure 1.1**) [3].

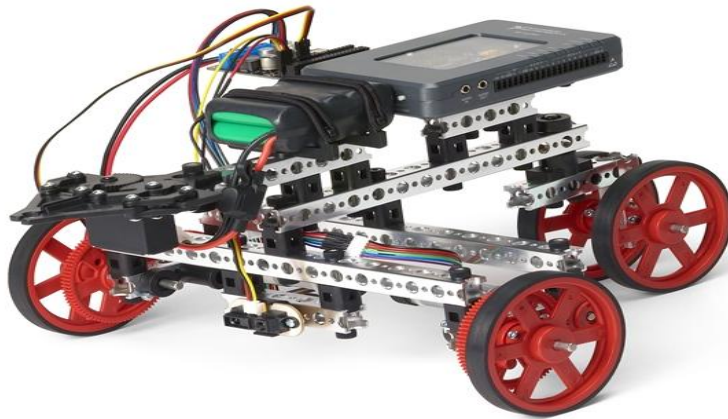


Figure 1.1 Robot mobile [3].

1.4.2 Robots Humanoïdes

Le terme humanoïde signifie « ressemblant à l'humain ». Il évoque la bipédie, la présence de deux bras et d'une tête (**Figure 1.2**) [3].

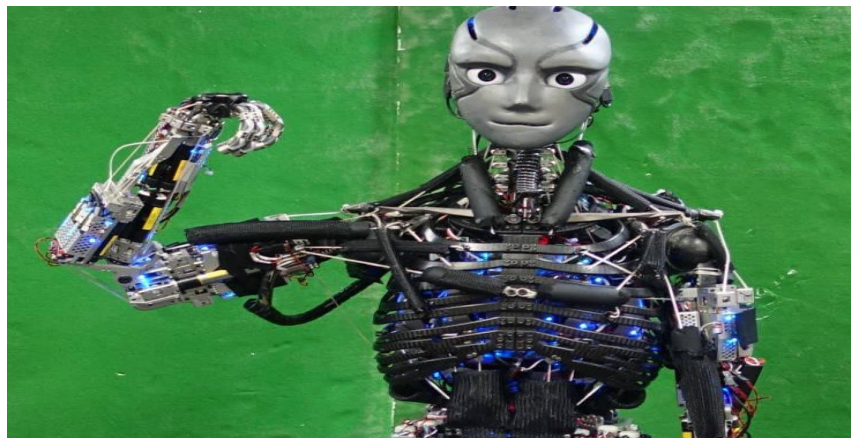


Figure 1.2 Rrobot Humanoïde [3].

1.4.3 Robots domestiques

Robots utilisés pour des tâches ménagères, par exemple en vaisselle, en repassage, en nettoyage (**Figure 1.3**) [3].



Figure 1.3 Robot domestique [3].

1.4.4 Robots industriels

Le robot industriel est officiellement défini comme un contrôle automatique

(**Figure 1.4**), reprogrammable, polyvalent, manipulateur programmable dans trois ou plusieurs axes [3].



Figure 1.4 Robot industriel [3].

1.4.5 Robots collaboratifs

Hommes et robots travaillent ensemble, les robots permettant de diminuer la pénibilité des manipulations manuelles, des efforts ou des mouvements réalisés par l'opérateur, (voire la Figure 1.5) [3].



Figure 1.5 Robot collaboratif [3].

1.5 Constituants d'un robot

Un robot est un assemblage complexe de pièces mécaniques et de pièces électroniques (figure 1.6), le tout pouvant être piloté par une intelligence artificielle [4].

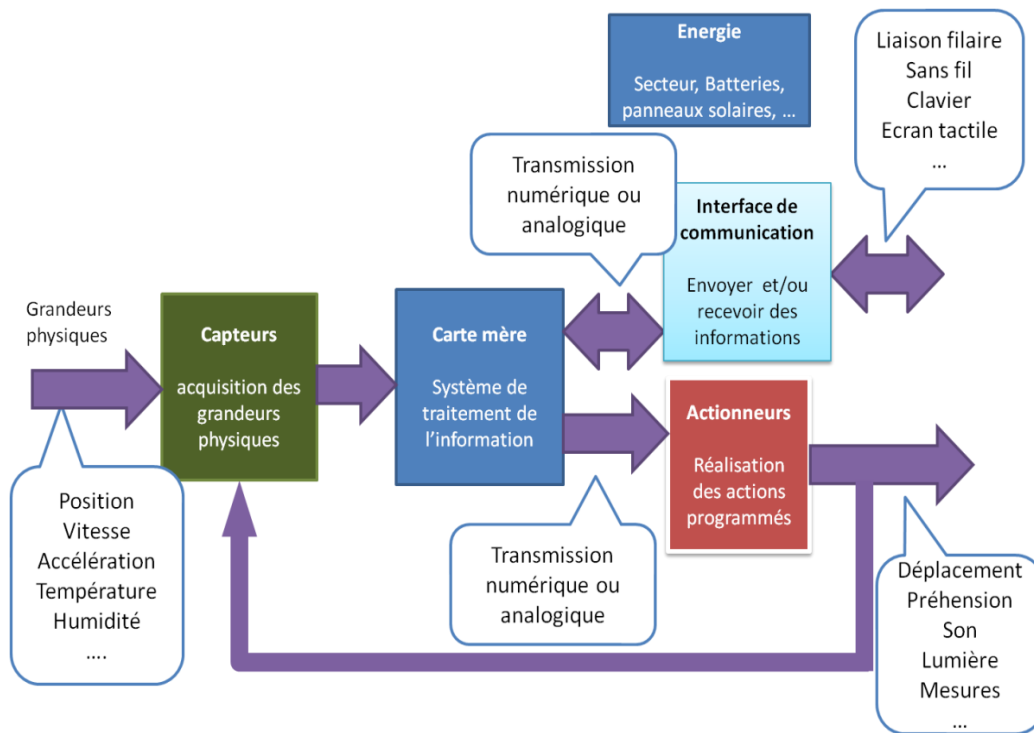


Figure 1.6 Structure Constituants Unité opérationnelle et fonctionnelle d'un robot [4].

Un robot est composé de quatre groupes de composantes principales :

1.5.1 Structure mécanique

C'est le squelette du robot (figure 1.7). Le choix des articulations permettra de déterminer les mouvements possibles et d'orienter son type d'utilisation. On compte un degré de liberté pour chaque axe de rotation selon lequel un membre (bras, patte) peut se mouvoir [5].

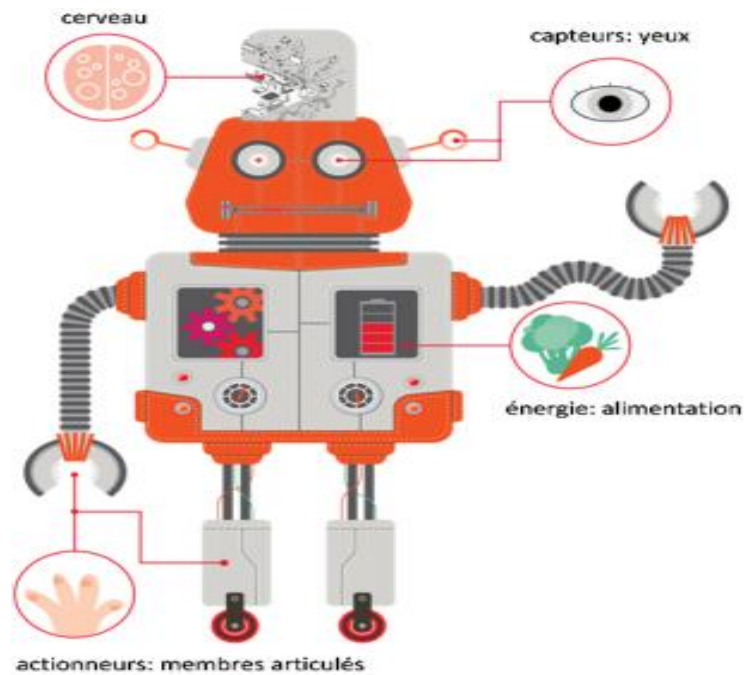


Figure 1.7 Structure mécanique d'un robot [5].

1.5.2 Les actionneurs

Les actionneurs permettront au robot d'effectuer concrètement les actions commandées par son système de commande, en fonction de ses objectifs et suite au traitement des informations recueillies par les capteurs. Les actionneurs sont des organes qui, sous commande, transforment l'énergie qui leur est fournie en actions physiques utilisables comme des mouvements [5].

1.5.3 Les Capteurs

Les capteurs s'occupent de la partie sensorielle du robot. Ils permettent d'acquérir des informations à propos de l'environnement du robot ou de ses composants internes [5].

1.5.4 Système contrôle commande

Le système contrôle commande du robot permet d'analyser les données provenant des capteurs et d'envoyer les ordres relatifs aux servo-moteurs. La partie commande est matérialisée physiquement par un microcontrôleur, qui est un cerveau électronique spécialement conçu pour interagir avec des capteurs et des actionneurs [5].

1.6 Robot mobile

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

1.7 L'architecture des robots mobiles

L'architecture des robots mobiles se structure en quatre éléments, (voire la **Figure 1.8**) [6] :

- ✓ La structure mécanique et la motricité.
- ✓ Les organes de sécurité.
- ✓ Le système de traitement des informations et gestion des tâches.
- ✓ Le système de localisation.

(Voire la **Figure1**)

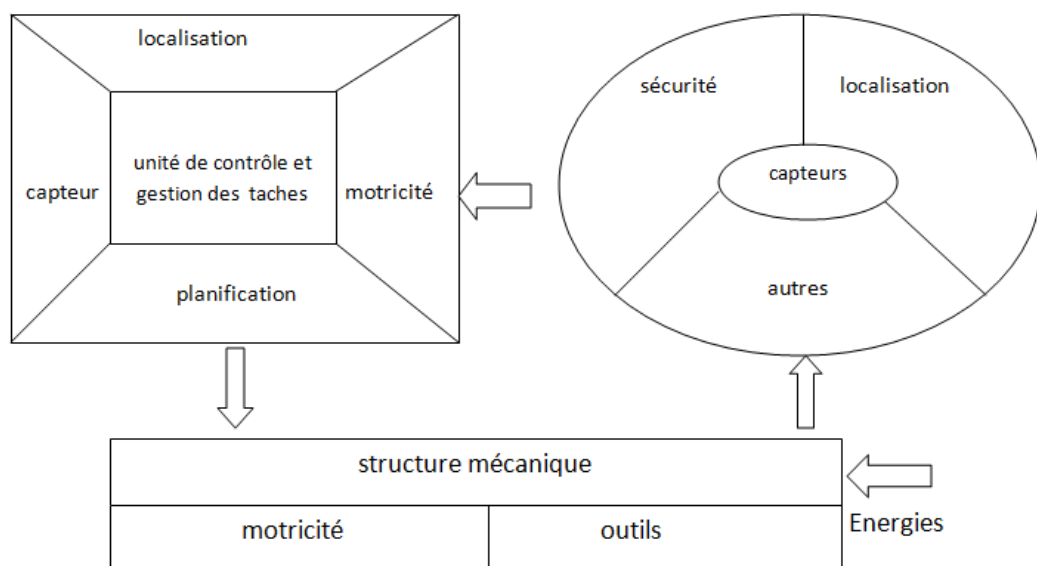


Figure 1.8 Architecture d'un robot mobile[6].

1.8 Les différents types des robots mobiles

1.8.1 Robot unicycle

Un robot de type uni-cycle est actionné par deux roues indépendantes, il possède éventuellement des roues pour assurer sa stabilité. Son centre de rotation est situé sur l'axe reliant les deux roues motrices (**Figure 1.9**) [6]:

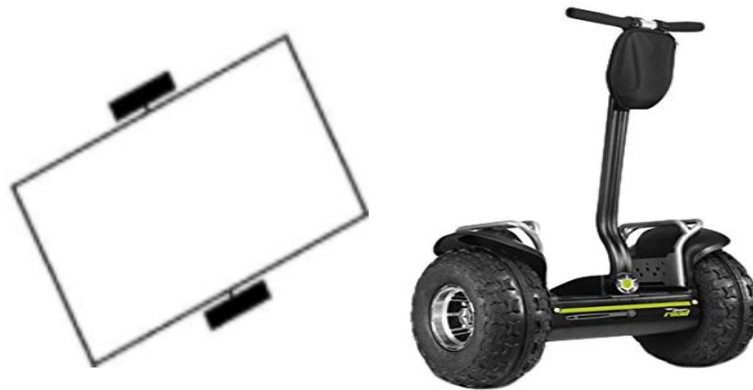


Figure 1.9 Robot de type unicycle [6].

1.8.2 Robot tricycle

Un robot de type tricycle est constitué de deux roues fixes placées sur un même axe et d'une roue centrée orientable placée sur l'axe longitudinal. Le mouvement du robot est donné par la vitesse des deux roues fixes et par l'orientation de la roue orientable. Son centre de rotation est situé à l'intersection de l'axe contenant les roues fixes et de l'axe de la roue orientable (voire la **Figure 1.10**) [6].

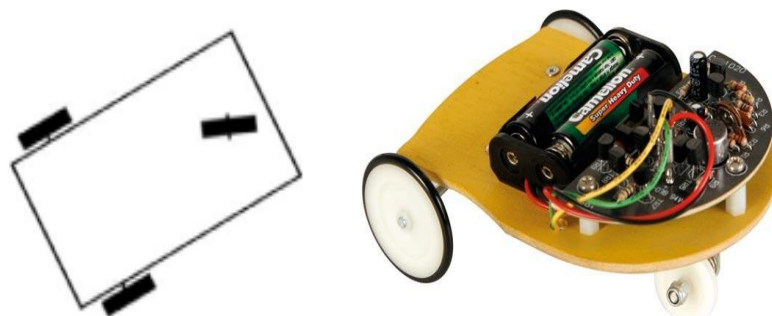


Figure 1.10 Robot de type tricycle [6].

1.8.3 Robot voiture

Un robot de type voiture est semblable au tricycle, il est constitué de deux roues fixes placées sur un même axe et de deux roues centrées orientables placées elles aussi sur un même axe (voire la **Figure 1.11**) [6].

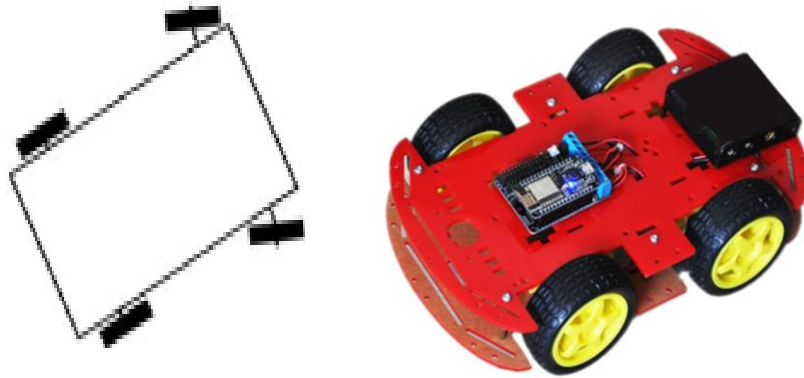


Figure 1.11 Robot de type voiture[6].

1.8.4 Robot omnidirectionnel

Un robot omnidirectionnel est un robot qui peut se déplacer librement dans toutes les directions. Il est en général constitué de trois roues décentrées orientables placées en triangle équilatéral (voire la **Figure 1.12**) [6].

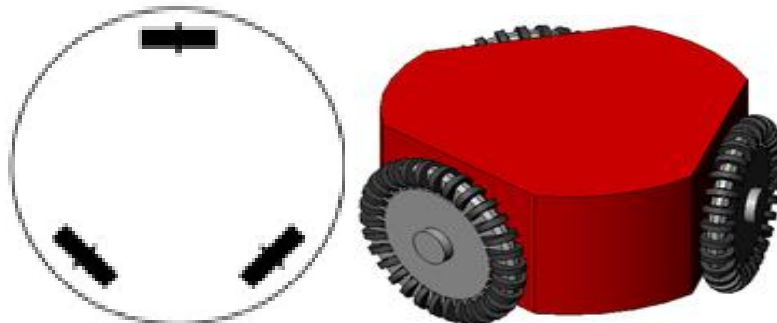


Figure 1.12 Robot de type omnidirectionnel [6].

1.9 Robot pendule inversé (uni-cycle)

1.9.1 Définition

Le pendule inversé présente une position d'équilibre instable que l'on cherche à stabiliser en utilisant un chariot mobile (**Figure1.13**). Plus simplement, une application

du pendule inversé est le système qui consiste à faire tenir un balai (la tête de celui-ci étant vers le haut) dans la paume de la main, le plus longtemps possible [8].

1.9.2 Les caractéristiques

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. L'angle d'inclinaison du pendule par rapport à la verticale est noté. 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 vis 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 [8]

1.9.3 La structure mécanique

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.

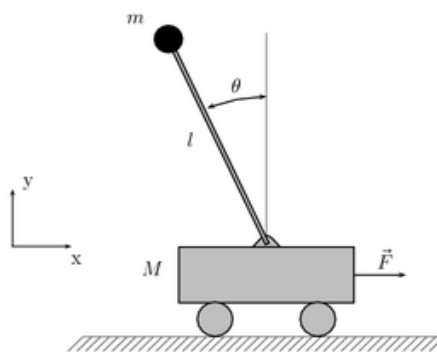


Figure 1.13 Schéma de l'ensemble chariot et pendule inversé[8].

1.10 Le domaine d'application

L'importance de l'étude des systèmes à pendule est illustrée par les exemples d'application cités ci-dessous.

1.10 En médecine

L'homme est vu comme un double pendule inversé en série, dont les deux axes de rotations principales 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 (remplacement chirurgical d'un organe ; la pièce ou l'appareil de remplacement : prothèse dentaire) pour les hanches sont amenés à utiliser le modèle de double pendule inversé pour calculer l'ensemble des contraintes qui sont soumises à la prothèse, comme le montre la figure suivante (voire la **Figure 1.14**) [9] .

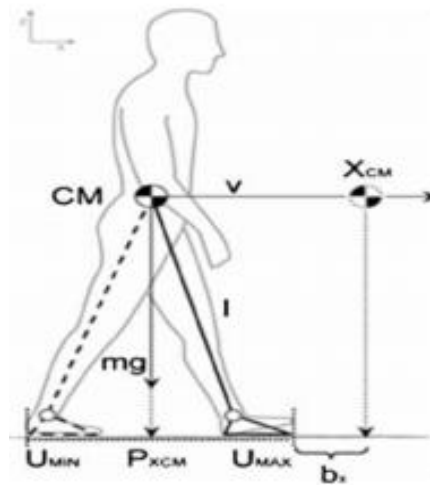


Figure 1.14 Le corps de l'être humain vu comme un double pendule[9].

Le premier pendule est articulé à la cheville et représente les membres inférieurs considérés groupés. Le second pendule est articulé à la hanche et représente la partie supérieure du corps. On accélère en se avançant et on ralentit en penchant en arrière. Le système est le même que le pendule inversé.

1.10.2 En robotique

Dans le même ordre d'idée, un robot est vu comme un pendule inversé. Pour maintenir le robot en équilibre il suffit de stabiliser le pendule inversé correspondant à sa position verticale au cours de son déplacement et en présence d'autres perturbations. A l'occasion nous citons quelques exemples des robots qui sont déjà étudiés et conçus à base de ce principe :

- Le robot BIPED : peut se présenter par deux doubles pendules inversés en série, dont les deux pendules sont couplés par une tige avec deux articulations (**figure 1.15**)[10].

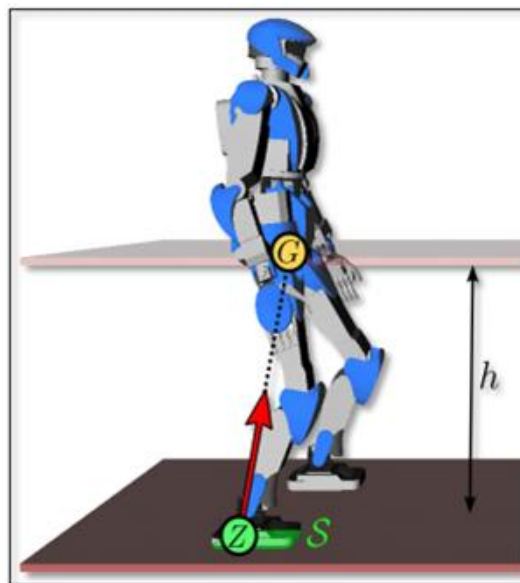


Figure 1.15 Le robot BIPED et sa représentation 3D[10].

- Le robot JOE : Il est vu comme un pendule inversé simple (une seule tige) se déplaçant à l'aide de deux roues

1.11 Conclusion

A travers ce chapitre on peut dire que dans le cadre de la robotique, la robotique mobile joue un rôle à part. Contrairement aux robots industriels manipulateurs qui travaillent de façon autonome dans un grand nombre d'usines automatisées, les robots mobiles sont très peu répandus. Cette situation n'est pas due au manque d'applications possibles, mais dès qu'on dispose de la mobilité, on peut imaginer des robots facteurs, nettoyeurs, gardiens, jardiniers et beaucoup d'autres. La faible diffusion est surtout due au fait que ces tâches ont une complexité bien supérieure à celles effectuées par des robots manipulateurs industriels. Le monde dans lequel un robot mobile doit se déplacer est souvent très vaste, partiellement ou totalement inconnu, difficilement caractérisable géométriquement et ayant une dynamique propre.

Dans le chapitre suivant on vas faire une présentation du matériels utilisé (hardware) et la présentation du logiciel LABVIEW (software).

Chapitre 2

Outils nécessaire pour la réalisation du projet (software et hardware)

2.1 Introduction

A travers ce chapitre nous allons exposer et expliquer le principe de fonctionnement de chaque outil nécessaire au développement du projet. Sur le plant hardware : l'unité de traitement (MYRIO FPGA), la motricité (actionneurs), et la perception (capteurs). Sur le plant software : le logiciel LABVIEW.

2.2 L'unité de traitement (MYRIO-1900 FPGA)

2.2.1 Définition

Le MYRIO-1900 de National Instruments est un périphérique d'E / S (RIO) portable reconfigurable qui les ingénieurs peuvent utiliser pour concevoir des systèmes de contrôle, de robotique et de mécatronique (voire **la Figure 2.1**) [11].

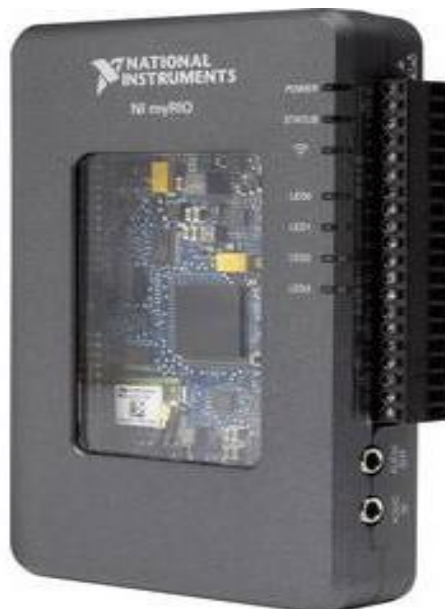


Figure 2.1 NI MYRIO-1900 National Instruments[11].

Il s'appuie sur l'architecture Zynq de Xilinx, et intègre donc sur la même puce un processeur double cœur ARM Cortex-A9 à 667 MHz et un FPGA reprogrammable. Cette puissance brute, associée au grand nombre d'E/S.

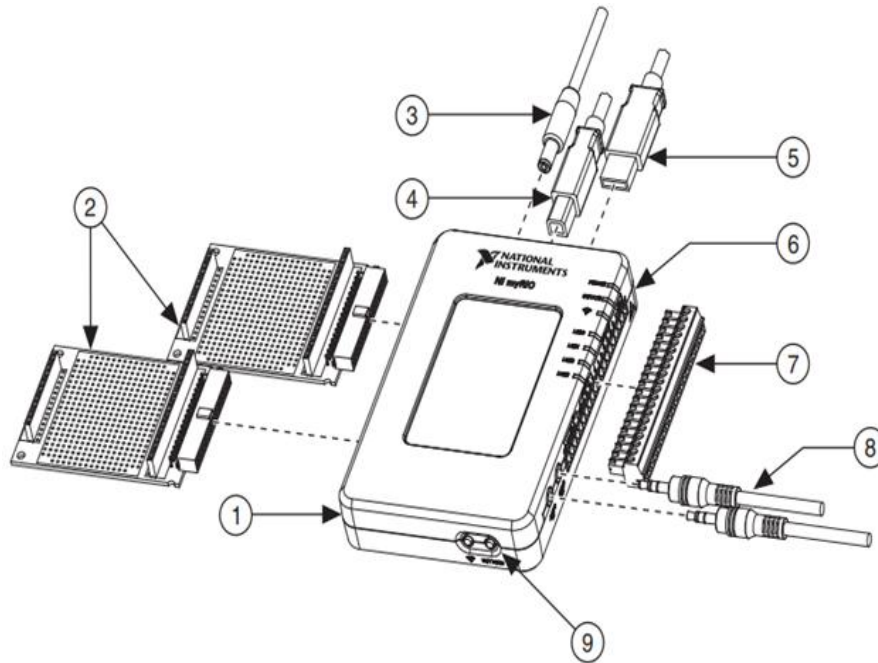


Figure 2.2 Composants de NI myRIO-1900 [11].

1. NI myRIO-1900
2. Sorties du port d'extension myRIO (MXP)
3. Câble d'entrée d'alimentation
4. Câble de périphérique USB
5. Câble USB Host
6. LEDS
7. Connecteur de borne à vis pour mini port système (MSP)
8. Câbles d'entrée / sortie audio
9. Bouton0 [11].

Voire la **Figure (2.2)**

2.2.2 Présentation du matériel

Le NI myrio 1900 fournit :

- 1) Entrées analogique (AI) ;
- 2) Sorties analogique (AO) ;
- 3) Entrées et sortie numérique (DIO) ;
- 4) Audio ;
- 5) Puissance de sortie dans un appareil compact intégré [11].

Le NI myrio 1900 se connecte à un ordinateur hôte via USB et sans fil 802.11b, g.n
(Figure 2.3).

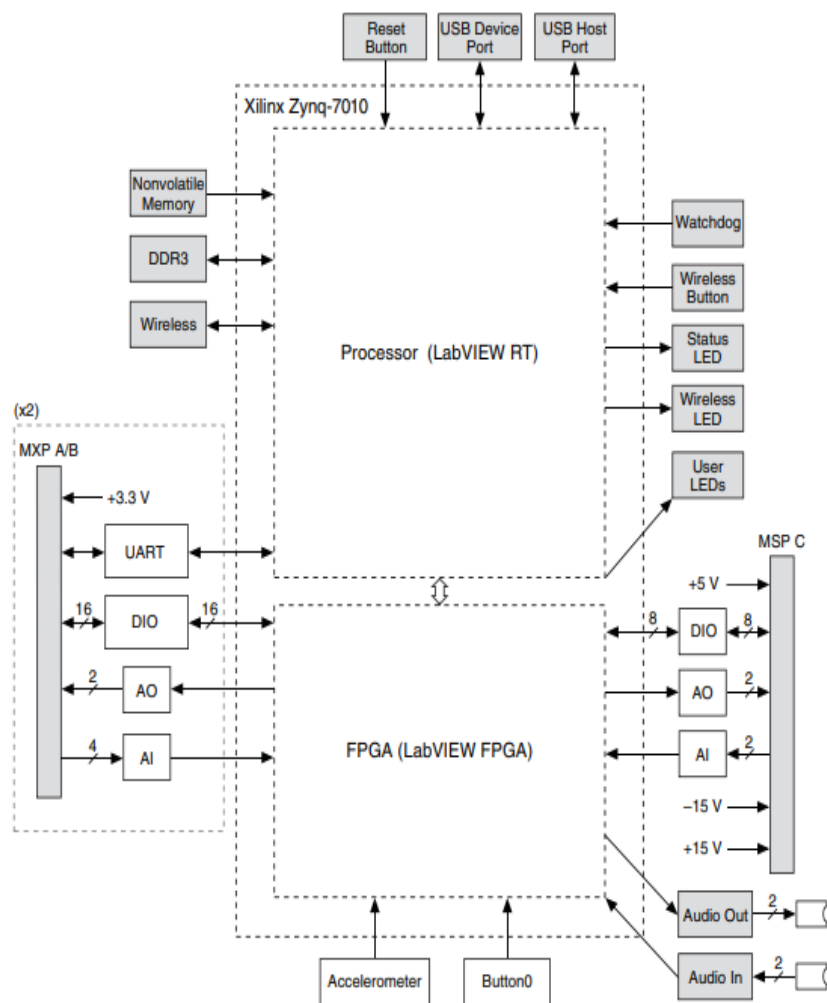


Figure 2.3 Schéma fonctionnel du matériel [11].

2.2.3 BROCHAGE DES CONNECTEURS

- les connecteurs A et B du port d'extension transportent des ensembles de signaux identiques.
 - les signaux sont distingués dans le logiciel par le nom du connecteur. **La figure (2.4)** et le tableau suivant montrent : les signaux sur les connecteurs MXP A et B.
- Notez que certaines broches portent des fonctions secondaires ainsi que des fonctions primaires [11] .

DIO15 / I2C.SDA	34	33	+3.3 V
DIO14 / I2C.SCL	32	31	DIO10 / PWM2
DGND	30	29	DIO9 / PWM1
DGND	28	27	DIO8 / PWM0
DIO13	26	25	DIO7 / SPI.MOSI
DGND	24	23	DIO6 / SPI.MISO
DIO12 / ENC.B	22	21	DIO5 / SPI.CLK
DGND	20	19	DIO4
DIO11 / ENC.A	18	17	DIO3
DGND	16	15	DIO2
UART.TX	14	13	DIO1
DGND	12	11	DIO0
UART.RX	10	9	A13
DGND	8	7	A12
AGND	6	5	A11
AO1	4	3	A10
AO0	2	1	+5V

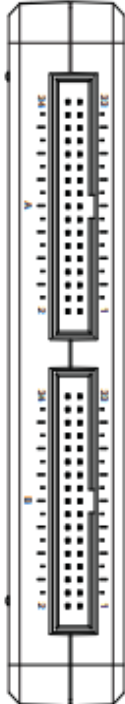


Figure 2.4 Signaux primaires / secondaires sur les connecteurs MXP A et B [11].

Signal Name	Reference	Direction	Description
+5V	DGND	Output	puissance de sortie +5v
AI <0..3>	AGND	Input	0-5 V, analogique référencé, asymétrique canaux d'entrée.
AO <0..1>	AGND	Output	0-5 V, analogique référencé, asymétrique canaux de sortie .
AGND	N/A	N/A	Référence pour entrée et sortie
+3.3V	DGND	Output	Puissance de sortie +3,3 V.
DIO <0..15>	DGND	Input or Output	Lignes numériques polyvalentes avec Sortie 3,3 V, compatible 3,3 V / 5 V contribution
UART.RX	DGND	Input	Entrée de réception UART. Les UART sont électriquement identique aux DIO.
UART.TX	DGND	Output	Sortie de transmission UART. Les UART sont électriquement identique aux DIO.
DGND	N/A	N/A	Référence pour les signaux numériques, +5 V, et +3,3 V.

Tableau 1 Descriptions des signaux sur les connecteurs MXP A et B[11].

La figure (2.5) et le tableau suivant montrent les signaux sur le connecteur C Mini System Port (MSP). Que certaines broches portent des fonctions secondaires ainsi que des fonctions primaires.

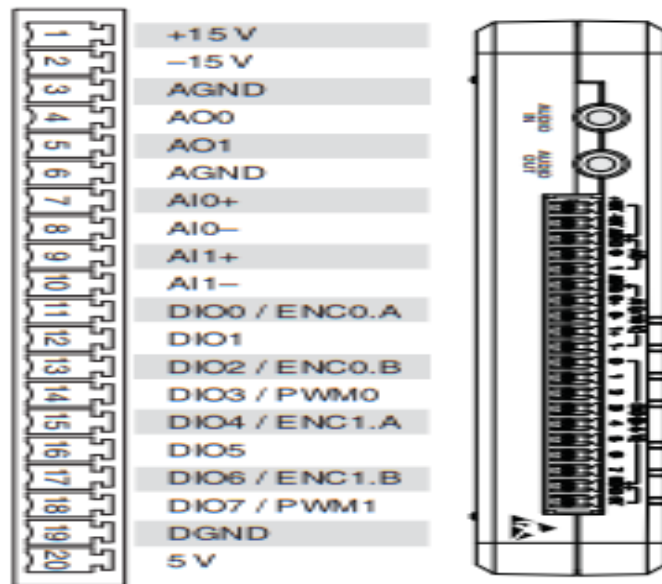


Figure 2.5 Signaux primaires / secondaires sur le connecteur MSP C [11].

Signal Name	Reference	Direction	Description
+15V/-15V	AGND	Output	Puissance de sortie +15 V / -15 V.
AI0+/AI0-; AI1+/AI1-	AGND	Input	± 10 V, entrée analogique différentielle canaux
AO <0..1>	AGND	Output	± 10 V référencé, analogique asymétrique canaux de sortie
AGND	N/A	N/A	Référence pour entrée et sortie analogiques et puissance de sortie +15 /- 15V.
+5V	DGND	Output	Puissance de sortie +5 V
DIO <0..7>	DGND	Input or Output	Lignes numériques polyvalentes avec Sortie 3,3 V, compatible 3,3 V / 5 V contribution.
DGND	N/A	N/A	Référence pour les lignes numériques et +5V puissance de sortie.

Tableau 2 Descriptions des signaux sur le connecteur MSP C [11].

Signal Name	Reference	Direction	Description
AUDIO IN	N/A	Input	Entrées audio gauche et droite sur stéréo connecteur
AUDIO OUT	N/A	Output	Sorties audio gauche et droite sur stéréo connecteur.

Tableau 3 Descriptions des signaux sur les connecteurs audio [11].

2.2.4 Canaux d'entrée analogiques

Le NI myRIO-1900 possède des canaux d'entrée analogiques sur les connecteurs A du port d'extension myRIO (MXP) et B, le connecteur C du mini port système (MSP) et un connecteur d'entrée audio stéréo (voire la Figure 2.6).

Les entrées analogiques sont multiplexées vers un seul convertisseur analogique-numérique (A-DC) qui échantillonne tous les canaux.

Les connecteurs MXP A et B ont quatre canaux d'entrée analogiques asymétriques par connecteur, AI0-AI3, que vous pouvez utiliser pour mesurer des signaux 0-5 V.

Le connecteur MSP C a deux hautes impédances, canaux d'entrée analogiques différentiels, AI0 et AI1, que vous pouvez utiliser pour mesurer des signaux jusqu'à ± 10 V.

Les entrées audio sont des entrées de niveau ligne stéréo gauche et droite avec une plage pleine échelle de $\pm 2,5$ V [11].

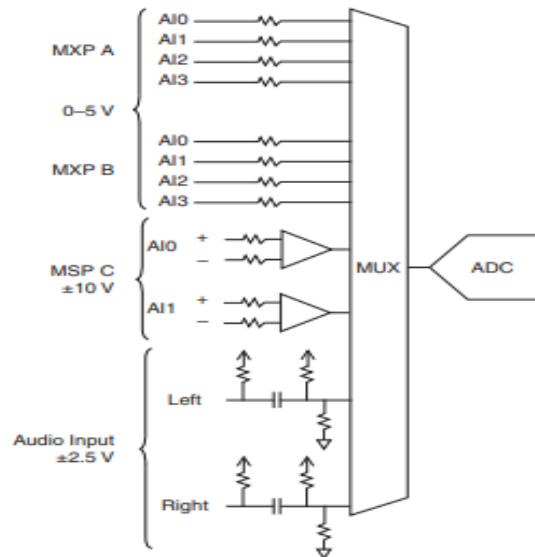


Figure 2.6 Circuit d'entrée analogique NI myRIO-1900 [11].

Le NI myRIO-1900 possède des canaux de sortie analogiques sur les connecteurs du port d'extension myRIO (MXP) A et B, connecteur C du mini port système (MSP) et un connecteur de sortie audio stéréo.

Chaque canal de sortie analogique a un convertisseur numérique-analogique (D-AC) dédié, afin qu'ils puissent tous mettre à jour simultanément.

Les D-AC des canaux de sortie analogiques sont contrôlés par deux bus de communication du FPGA.

Les connecteurs MXP A et B partagent un bus et MSP le connecteur C et les sorties audio partagent un deuxième bus. Par conséquent, la fréquence de mise à jour maximale est

Spécifié comme un chiffre global dans la section Sortie analogique des Spécifications.

Les connecteurs MXP A et B ont deux canaux de sortie analogiques par connecteur, AO0 et AO1, qui vous pouvez utiliser pour générer des signaux 0-5 V.

Le connecteur MSP C a deux canaux de sortie analogiques, AO0 et AO1, que vous pouvez utiliser pour générer des signaux jusqu'à ± 10 V.

Les sorties audio sont à gauche et à droite sorties stéréo de niveau ligne capables de piloter des écouteurs (Voire **la Figure 2.7**) [11].

2.2.5 Canaux de sortie analogique

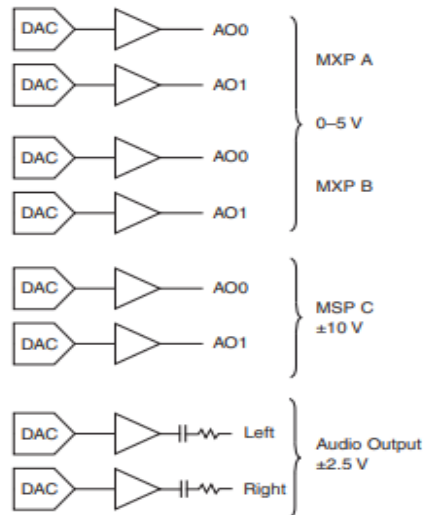


Figure 2.7 Circuit de sortie analogique NI myRIO-1900 [11].

2.2.6 Accéléromètre

Le NI myRIO-1900 contient un accéléromètre à trois axes. L'accéléromètre échantillonne chaque axe en continu et met à jour un registre lisible avec le résultat.

2.2.7 Conversion des valeurs de données brutes en tension

Vous pouvez utiliser les équations suivantes pour convertir les valeurs de données brutes en volts :

$$V = \text{Valeur des données brutes} * \text{Poids LSB}$$

$$\text{Poids LSB} = \frac{\text{Plage nominale}}{2^{\text{Résolution}} \text{ADC}}$$

Quand la valeur des données brutes est la valeur renvoyée par le nœud d'E / S FPGA,

Le poids LSB : est la valeur en volts de l'incrément entre les valeurs de données.
La plage nominale : est la valeur absolue en volts de la plage nominale complète, crête à crête de la chaîne.

La résolution ADC : est la résolution de l'ADC en bits. (Résolution ADC = 12).

- Pour les canaux AI et AO sur les connecteurs MXP:

$$\text{Poids LSB} = 5 \text{ V} \div 2^{12} = 1,221 \text{ mV}$$

$$\text{Lecture maximale} = 4095 * 1,221 \text{ mV} = 4,999 \text{ V}$$

- Pour les canaux AI et AO sur les connecteurs MSP:

$$\text{Poids LSB} = 20 \text{ V} \div 2^{12} = 4,883 \text{ mV}$$

$$\text{Lecture positive maximale} = +2047 * 4,883 \text{ mV} = 9,995 \text{ V}$$

$$\text{Lecture négative maximale} = -2048 * 4,883 \text{ mV} = -10\,000 \text{ V}$$

- Pour l'entrée / sortie audio :

$$\text{Poids LSB} = 5 \text{ V} \div 2^{12} = 1,221 \text{ mV}$$

$$\text{Lecture positive maximale} = +2047 * 1,221 \text{ mV} = 2,499 \text{ V}$$

$$\text{Lecture négative maximale} = -2048 * 1,221 \text{ mV} = -2,500 \text{ V}$$

- Pour l'accéléromètre:

$$\text{Poids LSB} = 16 \text{ g} \div 2^{12} = 3,906 \text{ mg}$$

$$\text{Lecture positive maximale} = +2047 * 3,906 \text{ mg} = +7,996 \text{ g}$$

$$\text{Lecture négative maximale} = -2048 * 3,906 \text{ mg} = -8\,000 \text{ g} , [11].$$

2.2.8 Lignes DIO

Le NI myRIO-1900 possède des lignes DIO à usage général de 3,3 V sur les connecteurs MXP et MSP.

Les connecteurs MXP A et B ont 16 lignes DIO par connecteur.

Sur les connecteurs MXP, chaque DIO la ligne de 0 à 13 a une résistance pull-up de 40 k Ω à 3,3 V (**Figure 2.8**) et les lignes DIO 14 et 15 ont 2,1 k Ω résistances pull-up à 3,3 V (**Figure 2.9**).

Le connecteur MSP C a huit lignes DIO, Chaque ligne MSP DIO a un 40 k Ω résistance à la masse, DGND est la référence pour toutes les lignes DIO (**Figure 2.10**).

Vous pouvez tout programmer les lignes individuellement comme entrées ou sorties [11].

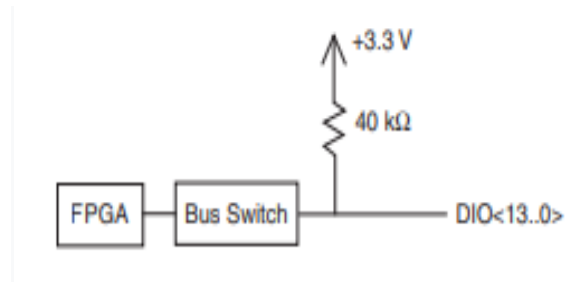


Figure 2.8 Lignes DIO <13..0> sur le connecteur MXP A ou B [11].

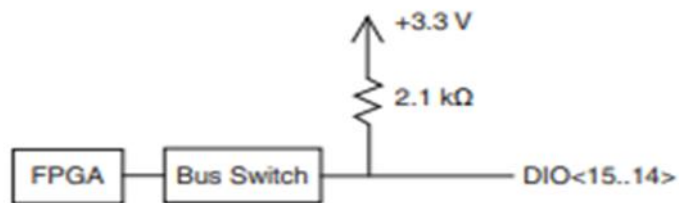


Figure 2.9 Lignes DIO <15..14> sur le connecteur MXP A ou B [11].

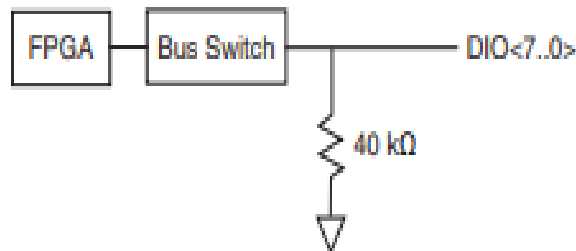


Figure 2.10 Lignes DIO <7..0> sur le connecteur MSP C [11].

Lorsqu'une ligne DIO flotte, elle flotte dans le sens de la résistance de tirage.

Une ligne DIO peut être flottante dans l'une des conditions suivantes :

- au démarrage de l'appareil myRIO
- lorsque la ligne est configurée comme entrée
- lorsque l'appareil myRIO s'éteint Vous pouvez ajouter une résistance plus forte à une ligne DIO pour la faire flotter dans la direction opposée [11].

2.2.9 Lignes UART

Le NI myRIO-1900 a une ligne d'entrée de réception UART et une ligne de sortie de transmission UART sur chaque connecteur MXP.

Les lignes UART sont électriquement identiques aux lignes DIO 0 à 13 sur le MXP connecteurs. Comme ces lignes, UART.RX et UART.TX ont des résistances pull-up(rappel) de 40 kΩ à 3,3 V. Utilisez LabVIEW Real-Time pour lire et écrire sur les lignes UART [11].

2.2.10 Utilisation du Bouton 0

Le bouton 0 produit une logique TRUE lorsqu'il est enfoncé et un logique FALSE lorsqu'il n'est pas enfoncé. Le bouton n'est pas anti-rebond [11].

2.2.11 Comprendre les indications des LED

a) LED d'alimentation

Le voyant d'alimentation est allumé lorsque le NI myRIO-1900 est sous tension. Cette LED indique que le l'alimentation électrique connectée à l'appareil est adéquate.

b) LED d'état

Le voyant d'état est éteint pendant le fonctionnement normal. Le NI myRIO-1900 exécute un autotest à la mise sous tension (POST) lorsque vous mettez le périphérique sous tension. Pendant le POST, les voyants d'alimentation et d'état sur. Lorsque le voyant d'état s'éteint, le POST est terminé. Le NI myRIO-1900 indique des conditions d'erreur spécifiques en faisant clignoter la LED d'état un certain nombre de fois toutes les quelques secondes, comme indiqué dans le tableau.

Nombre de flashes Toutes les quelques secondes	Indication
2	L'appareil a détecté une erreur dans son logiciel. Cela généralement se produit lorsqu'une tentative de mise à niveau du logiciel est interrompue. Réinstallez le logiciel sur l'appareil.
3	L'appareil est en mode sans échec.
4	Le logiciel s'est écrasé deux fois sans redémarrage ni cycle puissance entre les accidents. Cela se produit généralement lorsque l'appareil fonctionne Mémoire insuffisante. Passez en revue votre VI RT et vérifiez l'utilisation de la mémoire. Modifiez le VI si nécessaire pour résoudre le problème d'utilisation de la mémoire
Clignotant continuellement ou solide	L'appareil a détecté une erreur irrémédiable.

Tableau 4 Indications du voyant d'état [11].

c) LED 0 à 3

Vous pouvez utiliser les voyants 0 à 3 pour vous aider à déboguer votre application ou à récupérer facilement l'état de l'application. La logique TRUE allume une LED et la logique FALSE éteint une LED [11].

2.2.12 Utilisation du port hôte USB

NI myRIO-1900 prend en charge les caméras Web conformes à la vidéo USB Protocole de classe d'appareil (UVC) ainsi que des caméras de vision industrielle conformes à l'USB3 Vision standard et rétro compatible USB 2.0. Le port hôte USB NI myRIO-1900 prend en charge les caméras Basler ace USB3[11].

2.2.13 Caractéristiques

Les spécifications suivantes sont typiques pour la plage de températures de fonctionnement de 0 à 40 ° C [11].

A. Processeur

Type de processeur Xilinx Z-7010
La vitesse du processeur.....667 MHz
Cœurs de processeur 2

B. Mémoire

Mémoire non volatile 512 Mo
Mémoire DDR3 256 Mo
Fréquence d'horloge DDR3 533 MHz
Largeur du bus de données DDR3 16 bits

C. FPGA

Type FPGA Xilinx Z-7010

D. Caractéristiques sans fil

Mode radio IEEE 802.11 b, g, n
Bande de fréquence..... ISM 2,4 GHz
Largeur de canal À 20 MHz
Canaux USA 1 à 11, International 1 à 13
Puissance d'émission +10 dBm max (10 mW)
Gamme extérieure jusqu'à 150 m (ligne de visée)
Directivité de l'antenne Omnidirectionnel
Sécurité..... WPA, WPA2, WPA2-Entreprise

E. Ports USB

Port hôte USB USB 2.0 haute vitesse
Port de périphérique USB USB 2.0 haute vitesse
Entrée analogique
Fréquence d'échantillonnage agrégée 500 kS / s
Résolution..... 12 bits
Protection contre les surtensions ± 16 V

F. Connecteurs MXP

Configuration Quatre canaux asymétriques par
connecteur Impédance d'entrée > 500 k Ω acquisition à 500
kS / s 1M Ω sous tension et inactif
4,7 k Ω hors tension
Impédance de source recommandée 3 k Ω ou moins
Plage nominale 0 V à +5 V
Précision absolue ± 50 mV
Bande passante > 300 kHz

G. Connecteur MSP

Configuration Deux canaux différentiels
Impédance d'entrée Jusqu'à 100 nA de fuite sous tension ;
4,7 k Ω hors tension
Plage nominale ± 10 V
Tension de travail
(Signal + mode commun) ± 10 V de AGND
Précision absolue ± 200 mV
Bande passante 20 kHz minimum, > 50 kHz typique

H. Entrée audio

Configuration Une entrée stéréo composée de deux couplés AC,
canaux asymétriques
Impédance d'entrée 10 k Ω à DC
Plage nominale $\pm 2,5$ V
Bande passante 2 Hz à > 20 kHz

I. Sortie analogique

Taux de mise à jour maximaux agrégés

Tous les canaux AO sur les connecteurs MXP 345 kS / s

Tous les canaux AO sur le connecteur MSP et canaux de sortie audio 345 kS / s

Résolution 12 bits

Protection contre les surcharges ± 16 V

Tension de démarrage 0 V après l'initialisation

FPGA

J. Connecteurs MXP

Configuration Deux canaux asymétriques par connecteur

Gamme 0 V à +5 V

Précision absolue 50 mV

Variateur de courant 3 mA

Vitesse de balayage 0,3 V / μ s

K. Connecteur MSP

Configuration Deux canaux asymétriques

Gamme ± 10 V

Précision absolue ± 200 mV

Variateur de courant 2 mA

Vitesse de balayage 2 V / μ s

L. Sortie audio

Configuration Une sortie stéréo composée de deux canaux asymétriques couplés en courant alternatif

Impédance de sortie 100 Ω en série avec 22 μ F

Bande passante 70 Hz à > 50 kHz sous une charge de 32 Ω ;

2 Hz à > 50 kHz dans une charge à haute impédance

M. E / S numériques

Nombre de lignes

Connecteurs MXP 2 ports de 16 lignes DIO (un port par connecteur) ; une ligne UART.RX et une ligne UART.TX par connecteur

Connecteur MSP 1 port de 8 lignes DIO
Commande de direction Chaque Ligne DIO
programmable individuellement comme entrée ou sortie
Niveau logique Entrée LVTTTL compatible 5 V ; 3,3 V LVTTTL
output.

N. Niveaux logiques d'entrée Entrée

Basse tension, VIL 0 V min ; 0,8 V max
Entrée haute tension, VIH 2,0 V min ; 5,25 V max

O. Niveaux logiques de sortie

Sortie haute tension, VOH
Source 4 mA 2,4 V min ; 3,465 V max
Sortie basse tension, VOL
Descendant 4 mA 0 V min ; 0,4 V max
Largeur d'impulsion minimale 20 ns

P. Fréquences maximales pour les fonctions numériques secondaires

SPI 4 MHz
PWM 100 kHz
Entrée codeur en quadrature 100 kHz
I2C 400 kHz

Q. Lignes UART

Vitesse de transmission maximale 230400 bps
Bits de données..... 5, 6, 7, 8
Bits d'arrêt 1, 2
Parité..... Impair, Pair, Marque, Espace
Contrôle de flux XON / XOFF

R. Accéléromètre

Nombre d'axes 3
Gamme ± 8 g
Résolution..... 12 bits
Taux d'échantillonnage 800 S / s
Bruit..... 3,9 mgrms typique à 25 ° C.

S. Puissance de sortie

Puissance de sortie +5 V

Tension de sortie 4,75 V à 5,25 V

Courant maximum sur chaque connecteur 100 mA

Puissance de sortie +3,3 V

Tension de sortie 3,0 V à 3,6 V

Courant maximum sur chaque connecteur 150 mA

2.3 Moteur à courant continu

2.3.1 Définition du moteur courant continu

Un appareil à courant continu est un appareil électrique. Les moteurs courant continu sont des convertisseurs de puissance (voire la **Figure 2.11**) :

Soit ils convertissent l'énergie électrique absorbée en énergie mécanique lorsqu'ils sont capables de fournir une puissance mécanique suffisante pour démarrer puis entraîner une charge en mouvement. On dit alors qu'ils ont un fonctionnement en moteur [12].



Figure 2.11 Moteurs courant continu [12].

Soit ils convertissent l'énergie mécanique reçue en énergie électrique lorsqu'ils subissent l'action d'une charge entraînée. On dit alors qu'ils ont un fonctionnement en générateur (voire la **Figure 2.12**).

- En mode « moteur », l'énergie électrique est convertie en énergie mécanique.
- En mode « générateur » ou « dynamo », l'énergie mécanique est convertie en une énergie électrique capable de se comporter comme un frein.

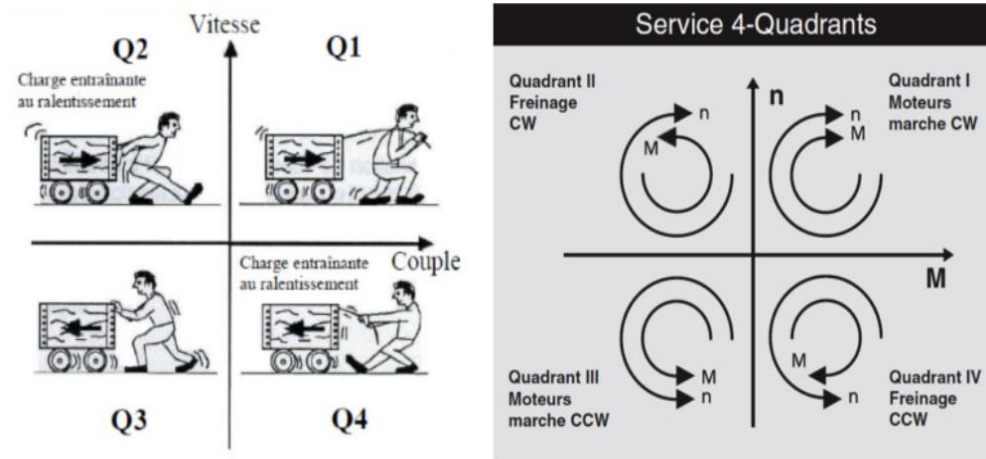


Figure 2.12 Mode de fonctionnement d'un moteur DC .

- Les quadrants Q1 et Q3 traduisent un fonctionnement de la machine en moteur dans les deux sens de rotation
- Les quadrants Q2 et Q4 traduisent un fonctionnement de la machine en générateur dans les deux sens de rotation.

Le nombre de quadrants de fonctionnement est exclusivement limité par le système de commande pilotant le moteur [12].

2.3.2 Les caractéristiques du moteur courant continu

Le moteur courant continu (DC) est caractérisé par une constante de vitesse, et une pente vitesse/couple. Le courant est proportionnel à la charge ; et la vitesse est proportionnelle à la tension d'alimentation.

Le moteur courant continu sans fer ne présente pas de couple magnétique résiduel et les agitations électromagnétiques sont dérisoires. Son rendement, qui atteint 90%, surpasse celui des autres technologies de moteurs.

Son rotor en forme de cloche donne la possibilité d'accélération très importantes et d'un couple de retenue inexistant.

Cela permet d'obtenir des positionnements précis et des vitesses faibles.

Le moteur est traditionnellement composé d'un stator à aimant permanent en terre rare de type Al-nico, samarium cobalt ou néodyme fer bore qui expliquent les caractéristiques dynamiques très élevées [12].

2.3.3 Les avantages / inconvénients du moteur courant continu :

L'avantage principal des moteurs à courant continu réside dans leur adaptation simple aux moyens permettant de régler ou de faire varier leur vitesse, leur couple et leur sens de rotation : les variateurs de vitesse, voire leur raccordement direct à la source d'énergie : batteries d'accumulateur, piles, etc. Ils ne nécessitent pas d'électronique pour les piloter, et peuvent être branchés directement sur une alimentation, des batteries, un variateur de vitesse, ou une carte de positionnement associée à un signal de copie.

- Possibilité d'entraîner de très fortes inerties ;
- Forte constante de temps mécanique ;
- Forte capacité à entraîner des surcharges élevées imprévisibles ralentissant le moteur : puisque son courant est proportionnel au couple, le moteur courant continu peut franchir des pointes de couple, et ainsi éviter les phénomènes de décrochage[12].

Inconvénients

La commutation du moteur à balais nécessite la mise en œuvre d'un ensemble de pièces mécaniques pour faire la liaison par frottement entre les charbons et le collecteur. Il en découle que :

- Plus la vitesse de rotation est élevée, plus la pression des balais doit augmenter pour rester en contact avec le collecteur donc plus le frottement est important;
- Aux vitesses élevées les balais doivent donc être remplacés très régulièrement ;

- Le collecteur imposant des ruptures de contact provoque des arcs, qui usent rapidement le commutateur et génèrent des parasites dans le circuit d'alimentation, ainsi que par rayonnement électromagnétique (réduit dans le cas des moteurs maxon par le système CLL (long life capacitor) [12].

2.4 Driver MD13S : 13 AMP DC MOTOR DRIVER

2.4.1 Introduction

MD13S est conçu pour entraîner un moteur CC à balais à courant élevé jusqu'à 13A en continu (voire la figure 2.13). Ça offre plusieurs améliorations telles que la prise en charge à la fois du PWM verrouillé d'antiphase et de magnitude de signe signal ainsi que l'utilisation de composants à semi-conducteurs complets qui se traduisent par un temps de réponse plus rapide et éliminer l'usure du relais mécanique[13].



Figure 2.13 Moteur driver MD13S [13].

2.4.2 Les caractéristiques de MD13S

- Contrôle bidirectionnel pour un moteur CC à balais.
- La tension du moteur de soutien varie de 6V à 30V.

- Courant maximum jusqu'à 13A en continu (sans dissipateur à 25 °C) et 30A en crête (10 seconde).
- Limitation de courant à 30A.
- Entrée de niveau logique 3,3V et 5V.
- Connecteur compatible GROVE.
- Les composants à semi-conducteurs offrent un temps de réponse plus rapide et éliminent l'usure de relais mécanique.
- Entièrement NMOS HBridge pour une meilleure efficacité et aucun dissipateur thermique n'est requis.
- Fréquence PWM de contrôle de vitesse jusqu'à 20 KHz (la fréquence de sortie réelle est la même que fréquence d'entrée).
- Prise en charge à la fois du fonctionnement PWM verrouillé en biphasé et en magnitude de signe.
- Prend en charge TTL PWM du microcontrôleur, pas PWM du récepteur RC.
- Compatible SMD
- Dimension : 61 mm x 33 mm [13].

2.4.3 Disposition et spécifications de la carte

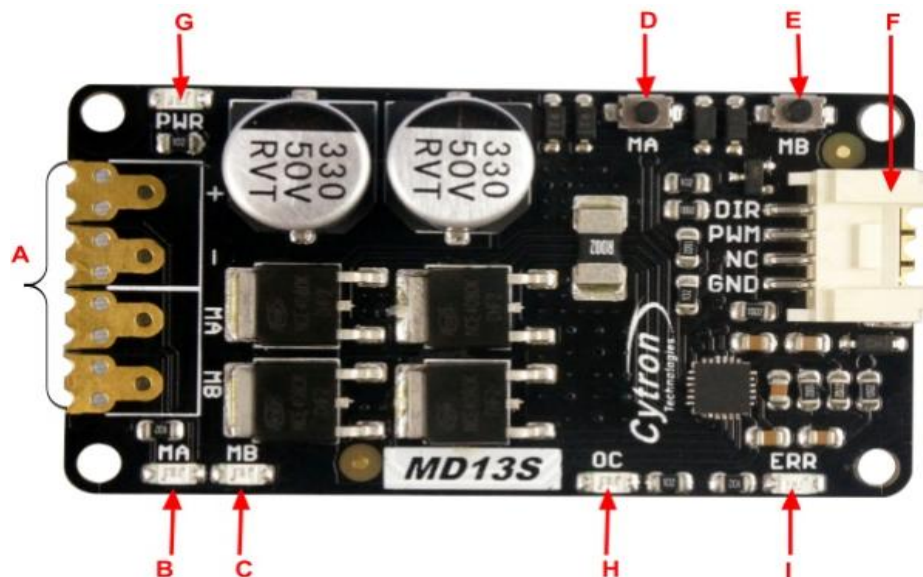


Figure 2.14 Schéma structurelle de la carte [13].

Etiquète	Fonction
A	Borner
B	Indicateur LED MA
C	Indicateur LED MB
D	Commutateur de test MA
E	Commutateur de test MB
F	Broche d'entrée
G	Indicateur LED d'alimentation
H	Indicateur LED OC
I	Indicateur LED ERR

Tableau 5 Définition des composants de la carte[13].

2.2.4 Borner - Connectez au moteur et à la source d'alimentation.

N ° de broche	Nom de la broche	La description
1	PUISSANCE +	Alimentation positive (6V à 30V)
2	PUISSANCE -	Alimentation négatif
3	Sortie moteur A	Connectez à la borne A du moteur
4	Sortie moteur B	Alimentation positive (6V à 30V)

Tableau 6 Description des broches [13].

Indicateur LED MA - S'allume lorsque la sortie A est haute et la sortie B est basse.

Indique que le courant circule de la sortie A vers B.

Indicateur LED MB - S'allume lorsque la sortie A est basse et la sortie B est haute.

Indique que le courant circule de la sortie B vers A.

Les entrées

N ° de broche	Nom de la broche	La description
1	GND	Terre logique
2	PWM	Entrée PWM pour le contrôle de la vitesse
3	DIR	Contrôle de direction

Tableau 7 Description des broches de commande [13].

- La table de vérité pour la logique de commande est la suivante

Pin 2 (PWM)	Pin 3 (DIR)	Sortie A	Sortie B
LOW	X	LOW	LOW
HIGH	LOW	HIGH	LOW
HIGH	HIGH	LOW	HIGH

Tableau 8 Table de vérité pour la logique de commande [13].

2.2.5 Le brochage

Voire la **Figure 2.15**

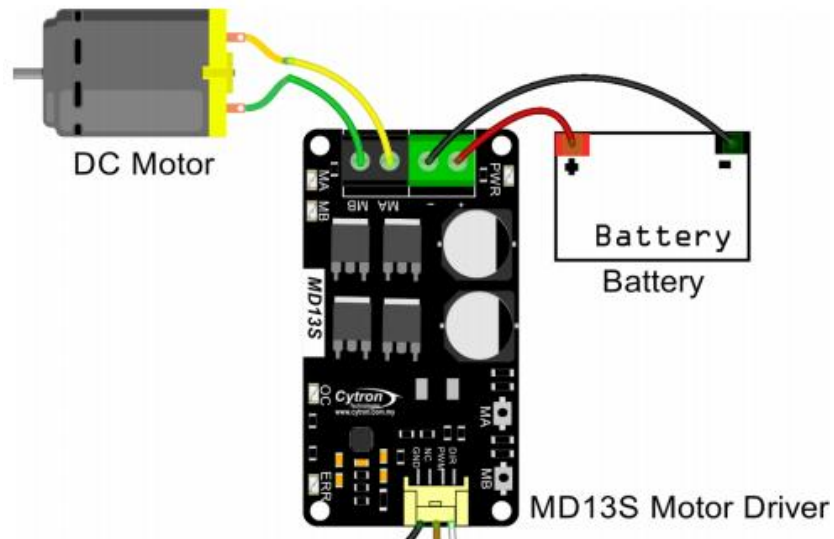


Figure 2.15 Le branchement de la carte avec le moteur et l'alimentation [13].

2.5 Capteur MPU-9250

2.5.1 Présentation du produit

Le MPU-9250 est un module multi-puces (MCM) composé de deux matrices intégrées dans un seul boîtier QFN. Une matrice abrite le gyroscope 3 axes et l'accéléromètre 3 axes. L'autre matrice abrite l'AK8963 3 axes magnétomètre d'Asahi Kasei Micro devices Corporation. Par conséquent, le MPU-9250 est un suivi de mouvement à 9 axes appareil qui combine un gyroscope à 3 axes, un accéléromètre à 3 axes, un magnétomètre à 3 axes et un mouvement numérique Processor™ (DMP) le tout dans un petit boîtier 3x3x1mm disponible en tant que mise à niveau compatible avec les broches à partir du MPU6515. Avec son bus de capteur I2C dédié, le MPU-9250 fournit directement un Motion Fusion™ à 9 axes . Le dispositif de suivi de mouvement MPU-9250, avec son intégration 9 axes, Motion Fusion™ sur puce et le micro logiciel d'étalonnage d'exécution, permet aux fabricants d'éliminer la sélection, la qualification, et l'intégration au niveau du système de dispositifs discrets, garantissant des performances de mouvement optimales pour les consommateurs. Le

MPU-9250 est également conçu pour s'interfacer avec plusieurs capteurs numériques non inertiels, tels que des capteurs de pression, sur son auxiliaire Port I2C (voire la **Figure 2.16**) [14].

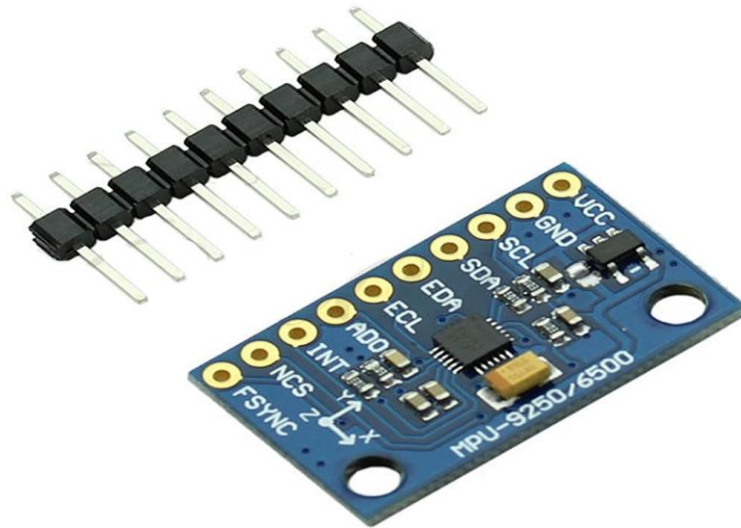


Figure 2.16 La carte MPU-9250 [14].

- Le MPU-9250 comprend trois convertisseurs analogique-numérique (ADC) 16 bits pour la numérisation des sorties du gyroscope, trois ADC 16 bits pour numériser les sorties de l'accéléromètre et trois ADC 16 bits pour numériser le magnétomètre les sorties. Pour un suivi précis des mouvements rapides et lents, les pièces disposent d'une gamme pleine échelle du gyroscope de ± 250 , ± 500 , ± 1000 et ± 2000 ° / s (dps), un accéléromètre programmable par l'utilisateur gamme pleine échelle de $\pm 2g$, $\pm 4g$, $\pm 8g$ et $\pm 16g$, et une gamme pleine échelle du magnétomètre de $\pm 4800\mu T$.
- Les autres caractéristiques de pointe de l'industrie comprennent des filtres numériques programmables, une horloge de précision avec une dérive de 1% à partir de -40 ° C à 85 ° C, un capteur de température intégré et des interruptions programmables. L'appareil comporte I2C et SPI interfaces série, une plage de fonctionnement VDD de 2,4 V à 3,6 V et une alimentation E / S numérique séparée, VDDIO à partir de 1,71 V à VDD.
- La communication avec tous les registres de l'appareil est effectuée en utilisant soit I2C à 400 kHz ou SPI à 1 MHz. Pour applications nécessitant des

communications plus rapides, le capteur et les registres d'interruption peuvent être lus à l'aide de SPI à 20 MHz [14] .

2.5.2 Caractéristiques

a Caractéristiques du gyroscope

Le gyroscope MEMS à trois axes du MPU-9250 comprend un large éventail de fonctionnalités: • Capteurs de vitesse angulaire (gyroscopes) à sortie numérique des axes X, Y et Z avec une plage pleine échelle programmable par l'utilisateur de ± 250 , ± 500 , ± 1000 et ± 2000 ° / s et des ADC 16 bits intégrés

- Filtre passe-bas programmable numériquement
- Courant de fonctionnement du gyroscope : 3,2 mA
- Courant du mode veille : 8 μ A
- Facteur d'échelle de sensibilité calibré en usine
- Autotest [14].

b Caractéristiques de l'accéléromètre

L'accéléromètre MEMS à trois axes du MPU-9250 comprend un large éventail de fonctionnalités :

- Accéléromètre à trois axes à sortie numérique avec une plage pleine échelle programmable de $\pm 2g$, $\pm 4g$, $\pm 8g$ et $\pm 16g$ et ADC 16 bits intégrés
- Courant de fonctionnement normal de l'accéléromètre : 450 μ A
- Courant du mode accéléromètre basse puissance : 8,4 μ A à 0,98 Hz, 19,8 μ A à 31,25 Hz
- Courant du mode veille : 8 μ A
- Interruptions programmables par l'utilisateur

- Interruption Wake-on-Motion pour un fonctionnement à faible consommation d'énergie du processeur d'applications

- Autotest [14] .

2.5.3 Fonctionnalités supplémentaires

Le MPU-9250 comprend les fonctionnalités supplémentaires suivantes [14] :

- Bus I2C maître auxiliaire pour la lecture des données de capteurs externes (par ex. Capteur de pression) ;

- Courant de fonctionnement de 3,5 mA lorsque les 9 axes de détection de mouvement et le DMP sont activés ;

- Plage de tension d'alimentation VDD de 2,4 à 3,6 V ;

- Tension de référence VDDIO pour l'auxiliaire Appareils i2C ;

- Boîtier QFN le plus petit et le plus fin pour les appareils portables : 3x3x1mm ;

- Sensibilité transversale minimale entre les axes de l'accéléromètre, du gyroscope et du magnétomètre ;

- Le tampon FIFO de 512 octets permet au processeur d'applications de lire les données en rafales ;

- Capteur de température à sortie numérique ;

- Filtres numériques programmables par l'utilisateur pour gyroscope, accéléromètre et capteur de température ;

- 10 000 g de tolérance aux chocs ;

- I2C Fast Mode 400 kHz pour communiquer avec tous les registres ;

- Interface série SPI 1MHz pour communiquer avec tous les registres ;

- Interface série SPI 20 MHz pour la lecture des registres de capteur et d'interruption.

2.5.4 Applications

- Services basés sur la localisation, points d'intérêt et l'estime ;
- Combiné et jeux portables ;
- Contrôleurs de jeu basés sur le mouvement ;
- Télécommandes 3D pour DTV et décodeurs connectés à Internet, souris 3D ;
- Capteurs portables pour la santé, le fitness et le sport [14] .

2.6 Le logiciel LABVIEW

2.6.1 Définition

Est un environnement de développement spécialisé en informatique industrielle et scientifique. Sa particularité est qu'il s'appuie sur le langage G, créé par National Instruments, qui est entièrement graphique. Il permet de créer des logiciels complexes tout en facilitant la programmation et donc de diminuer les délais de développement. Grâce à ses bibliothèques de fonctions dédiées à l'acquisition de données, l'instrumentation, à l'analyse mathématique des mesures, mais également grâce à la création rapide d'interfaces graphiques de qualité et le codage simplifié, l'ingénieur a plus de temps pour se concentrer sur les fonctions métiers de l'instrumentation et du traitement des mesures[15].

LabVIEW est particulièrement recommandé pour développer des systèmes de contrôle, supervision et les bancs de test et mesure.

2.6.2 Les langages de programmation utilisable dans labview

1. DATA FLOW
2. SIMULINK
3. C
4. HDL
5. STATECHARTS
6. MATLAB [15].

2.6.3 Partie de la virtuel instrumentale (vi)

a) Front panel (supervision) : c'est l'interface utilisateur graphique.

- Contrôle : entrées
- Indication : sorties

b) Bloc diagram (le circuit) : c'est le code source graphique.

- Les objets du panneau avant apparaissent sous forme de bornes.
- icon /connector pane : c'est une représentation graphique d'une version instrumentale (VI).
- Moyens de connexion [15].

2.6.4 DATA FLOW

- L'exécution du schéma fonctionnel (block diagram) dépend du flux de données (FLOW DATA).
- Le block diagram ne s'exécute pas de gauche à droite.
- Le nœud (NODE) s'exécute lorsque les données sont disponibles pour tous les terminaux d'entrée
- Le nœud (NODE) fournit des données à toutes les bornes de sortie une fois terminé
- L'ordinateur exécutant ce code avait plusieurs processeurs, ces morceaux de code peuvent fonctionner indépendamment sans codage supplémentaire[15].

2.6.5 Mesures d'accéléromètre avec le myrio-1900

Il faut configurer et mettre à jour le NI MYRIO-1900 avant de continuer avec le reste

c Configuration du myrio-1900

- **Installation du matériel : Connexion de NI myRIO**

Un des objectifs de myRIO est de simplifier l'installation du matériel. Pour y parvenir, NI myRIO dispose d'un utilitaire d'installation et de configuration différent de NI Measurement & Automation Explorer (MAX). Vous pouvez toutefois utiliser MAX si

vous le souhaitez et si vous êtes plus à l'aise avec cet environnement. NI myRIO dispose par exemple d'un utilitaire de surveillance des connexions USB, qui s'exécute quand vous connectez l'appareil à l'ordinateur hôte[15] .

- **NI myRIO USB Monitor**

Faites bien attention à alimenter NI myRIO avec l'adaptateur fourni. Branchez le câble USB de type B sur NI myRIO et l'autre extrémité sur un port USB de l'ordinateur hôte. Sans avoir à démarrer LabVIEW ou NI MAX, à condition que l'appareil soit alimenté, le système d'exploitation devrait reconnaître NI myRIO et installer les pilotes nécessaires. Une fois cette étape accomplie, sous un système d'exploitation de type Windows, Windows devrait automatiquement lancer l'utilitaire NI myRIO USB Monitor

Ci-dessous (voire la figure 2.17) [15] .

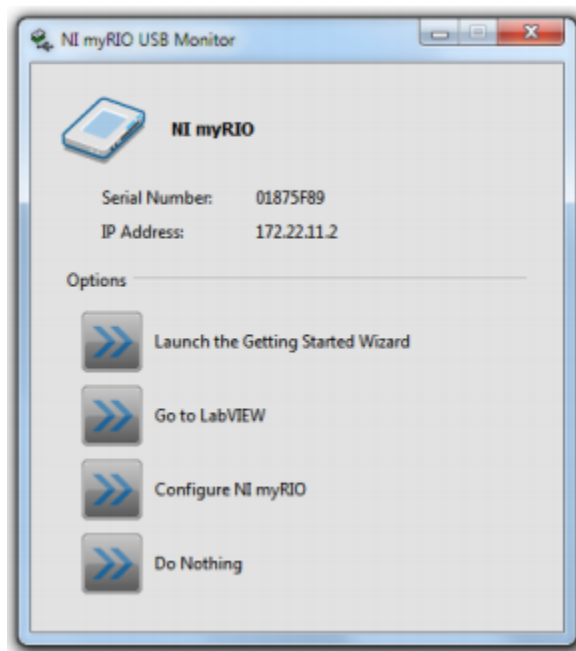


Figure 2.17 Moniteur USB [15].

En même temps que le numéro de série et l'adresse IP, vous disposez de quatre options une fois que NI myRIO est détecté[15] :

1. Launch the Getting Started Wizard

L'assistant Getting Started Wizard vous permet d'observer rapidement le statut fonctionnel de l'unité NI myRIO. L'assistant cherche et détecte des appareils NI myRIO, se connecte aux unités sélectionnées, vérifie que la partie logicielle est à jour, suggère une mise à jour s'il en éprouve le besoin, permet de renommer l'appareil puis affiche une face-avant qui vous permet d'observer les valeurs mesurées par l'accéléromètre, allumer ou éteindre les LEDs en façade, et tester le bouton reprogrammable par l'utilisateur.

L'écran final de l'assistant Getting Started Wizard propose deux options:

➤ **Start my first project now**

Sélectionner cette option lance un tutorial affiché dans un navigateur web.

2. Go to LabVIEW

Sélectionner cette option lance LabVIEW et son écran d'accueil.

3. Configure NI myRIO

Sélectionner cette option ouvre l'utilitaire de configuration de NI myRIO dans un navigateur web.

4. Do Nothing

Si LabVIEW est déjà ouvert et qu'un projet est configuré pour cibler NI myRIO, vous pouvez sélectionner cette option pour fermer NI myRIO USB Monitor quand une unité est reconnectée à l'ordinateur de développement.

- **Lancement de l'assistant Getting Started Wizard**

Depuis NI myRIO USB Monitor, sélectionnez **Launch the Getting Started Wizard**.

Cliquez sur **Next** et l'assistant se connecte à l'unité myRIO, vérifie les mises à jour logicielles et vous propose de renommer l'appareil.

Si aucun logiciel n'est installé sur NI myRIO, l'assistant installera automatiquement la version la plus récente.

Ensuite, une fenêtre de diagnostic apparaît (figure 2.18). Elle vous permet de visualiser les valeurs mesurées par l'accéléromètre 3 axes, tester le bouton reprogrammable par l'utilisateur, et allumer ou éteindre les LEDs en façade.

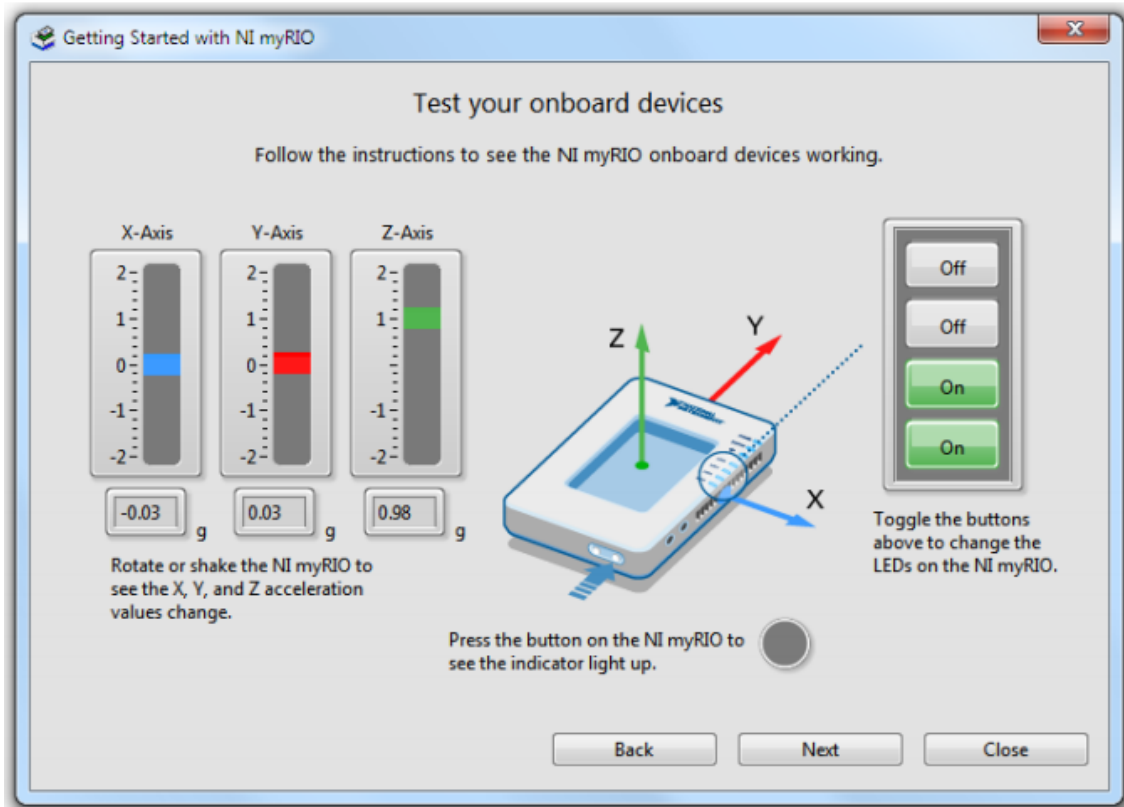


Figure 2.18 Test de l'appareil [15].

Maintenant que vous avez installé et configuré NI myRIO, vous allez pouvoir créer des VIs temps réel et les exécuter sur le processeur (de la même manière qu'un VI Windows) en profitant du FPGA pour libérer la vraie puissance du calcul parallèle. Dans la prochaine section, vous créez un projet par défaut et explorez les fonctionnalités d'une architecture basique.

d **Crée un projet myrio**

1. Depuis l'écran d'accueil de LABVIEW (**Figure 2.19**), cliquez sur le bouton **Create Project**.

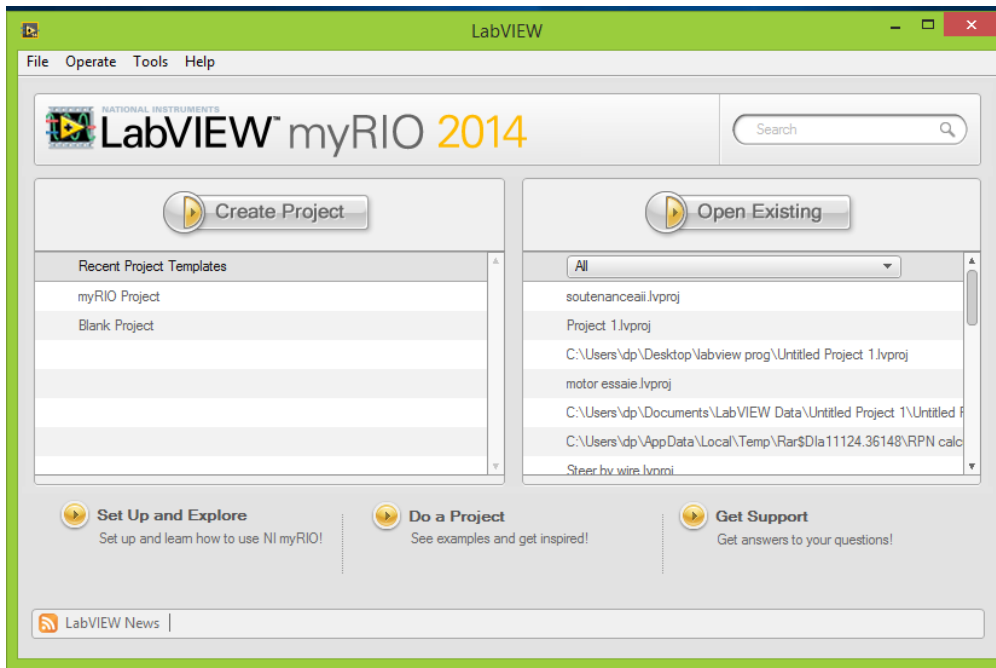


Figure 2.19 L'écran d'accueil de LabVIEW [15].

2. Dans le panneau de gauche de la boîte de dialogue "Créer un projet", sélectionnez **myRIO** dans l'arborescence des modèles de conception (voir Figure 2.20).

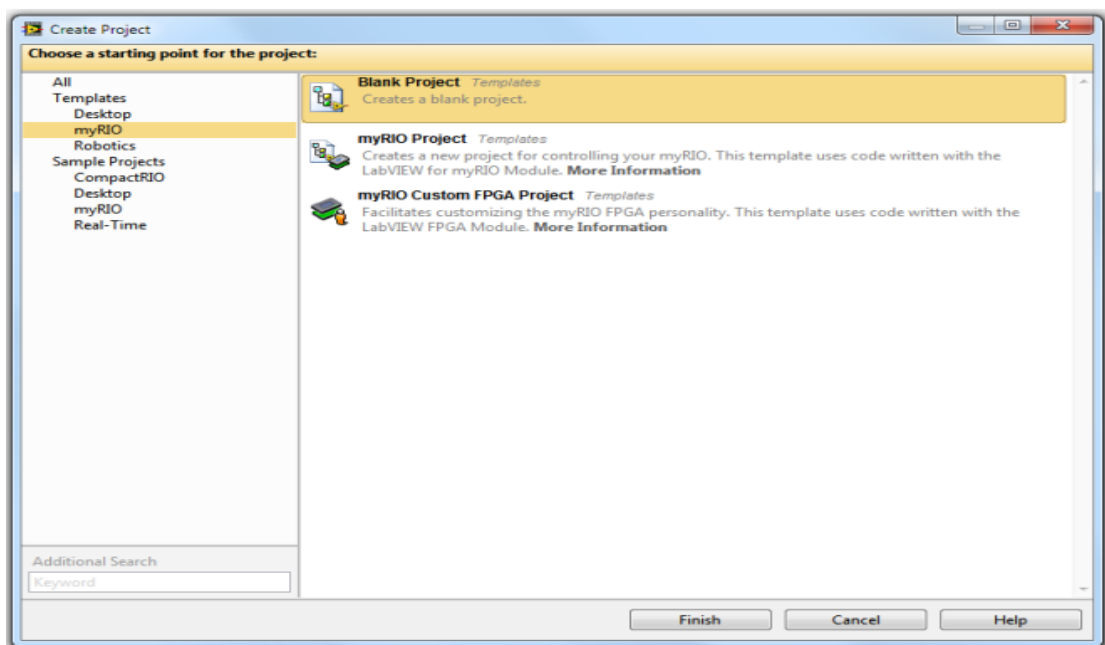


Figure 2.20 Point de départ pour le projet [15] .

3. Le panneau de droite affiche maintenant trois options : un projet standard vide, un projet myRIO et un projet de personnalisation du FPGA.

- a. Utilisez le modèle de conception myRIO Project pour créer un projet avec la personnalité FPGA par défaut. Ce modèle est utile pour les projets qui n'ont pas besoin des fonctionnalités étendues offertes par la personnalisation du FPGA.
- b. Utilisez le modèle de conception myRIO Custom FPGA Project pour personnaliser la définition du FPGA sur NI myRIO. Par exemple, les trois connecteurs offrent par défaut huit lignes d'E/S numériques de type PWM. Si vous avez besoin de connecter plus d'actionneurs pilotés en PWM, ou plus d'entrées, il est possible de reconfigurer le FPGA pour offrir le support du PWM sur un nombre supérieur de lignes d'E/S numériques.

4. Sélectionnez **myRIO Project** dans la liste et cliquez sur le bouton **Suivant**.

5. Donnez un nom approprié au projet et sélectionnez un répertoire pour y sauvegarder le projet. Vérifiez que la case à cocher **Plugged into USB** est sélectionnée et que l'unité NI myRIO porte le bon nom dans la liste de droite.

6. Cliquez sur **Terminer** quand tout semble configuré correctement (**Figure 2.21**) [15] .

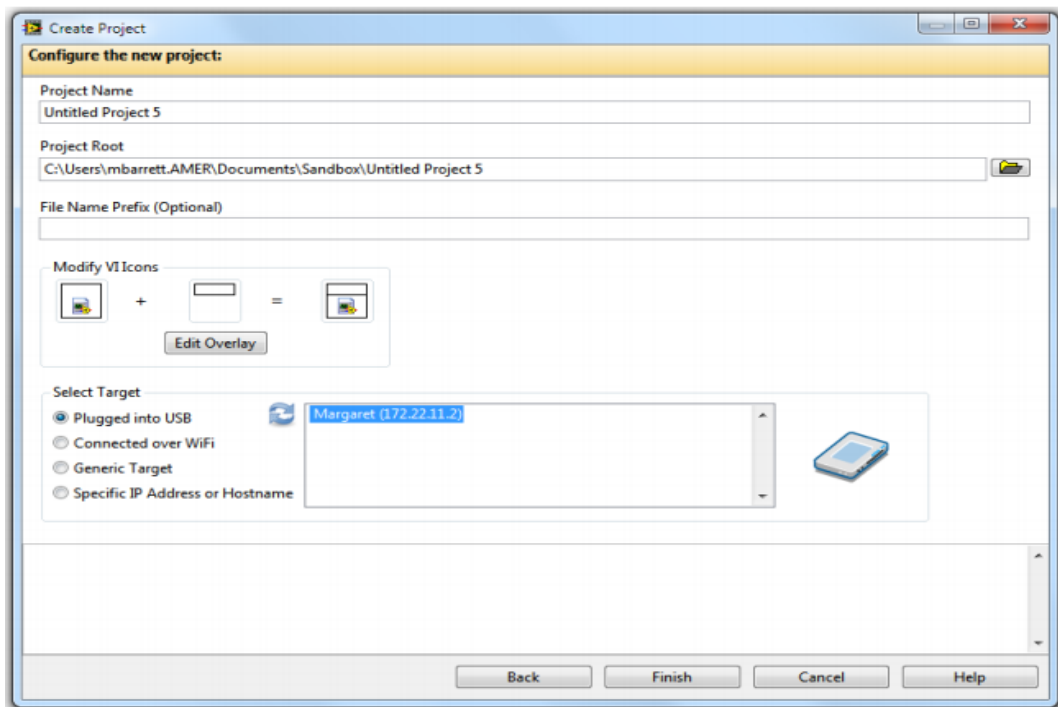


Figure 2.21 Configuration du projet[15] .

2.7 Conclusion

Nous avons présenté dans ce chapitre les définitions ,les caractéristiques et le mode de fonctionnement de tous le matériel utilisé (Hardware) qui est :

- NI MYRIO-1900
- le capteur mpu9250
- le driver moteur MD13S
- le moteur à courant continue .

Dans ce chapitre nous avons également relevé la partie software :

- La définition
- les langages de programmation
- le guides d'utilisation du logiciel LABVIEW.

Dans le chapitre suivant nous allons faire la modélisation de notre système ainsi que la programmation avec le logiciel Labview.

Chapitre 3

Développement du projet

3.1 Introduction

Dans ce chapitre nous allons mettre au clair les étapes de conception en partant par le développement des équations nécessaires pour le déplacement et la stabilisation du robot et en terminant par la réalisation pratique du robot en assemblant tous les éléments et outils cités dans le chapitre 2 et nous clôturons ce chapitre par les résultats obtenus.

3.2 Coordonnées généralisées du système

L'ensemble du chariot pendule à deux degrés de liberté qui sont représentées par deux coordonnées généralisées [16] [17]:

X : Pour le déplacement horizontal du chariot

θ : Pour la rotation du pendule.

La direction positive de x est le sens à droite en mètre et celui de l'angle est le sens des aiguilles d'une montre en radian.

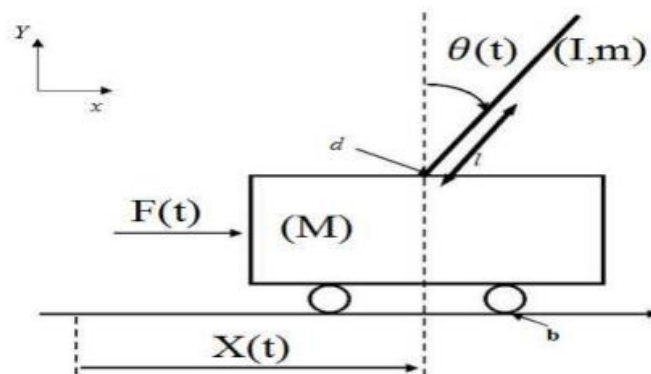


Figure 3.1 Schéma de l'ensemble chariot et pendule inversé [16].

Soit :

m: masse du pendule (kg)

x(t): position du chariot (m)

M : masse du chariot (kg)

b: frottements de déplacement du chariot(Ns/m)

L : la longueur du pendule (m)

$\theta(t)$: l'angle du pendule(rad)

g : intensité de pesanteur (m/s^2)

d : frottements du pendule (N ms/rad)

F(t) : force exercée sur le chariot (N)

3.2.1 Modélisation du système chariot-pendule à deux degrés de liberté

Les équations du mouvement du pendule sont déterminées par le formalisme d'Euler-Lagrange qui est basé sur le principe de la conservation de l'énergie mécanique plutôt que sur le concept de force, comme dans le principe de Newton.

Le Lagrangien est défini comme étant la différence entre l'énergie cinétique et l'énergie potentielle E_p du système, l'avantage de ce formalisme réside dans l'élimination des efforts d'interaction, Il s'exprime par [16] [17] :

$$L = E_c - E_p \quad (3.1)$$

Pour simplifier la modélisation, on suppose tout au long de notre étude que la masse du pendule est concentrée au bout de la tige.

Afin de pouvoir écrire les équations dynamiques régissant le mouvement du pendule inversé, on considère les coordonnées $q_1 = x$ et $q_2 = \theta$ qui représente les deux degrés de libertés du pendule

❖ Energie cinétique du système en mouvement

Le système en mouvement comporte le chariot qui se déplace et le pendule qui se balance sur son axe de rotation[16] [17].

L'énergie cinétique du chariot en mouvement est donnée par l'équation :

$$EM = 1/2 M \dot{x}^2 \quad (3.2)$$

L'énergie cinétique du pendule est exprimée par l'équation :

$$E_{cm} = 1/2 m \overrightarrow{v}_c^2 \quad (3.3)$$

v_c : La vitesse de centre de gravité du pendule

La position du centre de gravité du pendule, notée r_c à partir de ces coordonnées est donnée par :

$$\overrightarrow{Rc} = (x + L \sin \theta) \vec{i} + L \cos \theta \vec{j} \quad (3.4)$$

D'où la vitesse du centre de gravité du pendule est :

$$\overrightarrow{v}_c = \frac{d\overrightarrow{Rc}}{dt} = (\dot{x} + L \cos(\theta) \dot{\theta}) \vec{i} - L \dot{\theta} \sin \theta \vec{j} \quad (3.5)$$

En substituant les équations (3.5) dans l'équation (3.3) on trouve :

$$E_{cm} = 1/2 m (\dot{x}^2 + 2\dot{x} L \cos(\theta) \dot{\theta} + L^2 \cos^2(\theta) \dot{\theta}^2 + L^2 \sin^2(\theta) \dot{\theta}^2) \quad (3.6)$$

Après calcul, l'équation (3.6) de l'énergie cinétique du pendule prend alors la forme :

$$E_{cm} = 1/2 m (\dot{x}^2 + 2L \cos(\theta) \dot{\theta} \dot{x} + L^2 \dot{\theta}^2) \quad (3.7)$$

L'énergie cinétique de l'ensemble chariot (3.2) et pendule (3.7) est exprimée par :

$$E_c = EM + E_{cm} = 1/2 M \dot{x}^2 + 1/2 m (\dot{x}^2 + 2L \cos(\theta) \dot{\theta} \dot{x} + L^2 \dot{\theta}^2) \quad (3.8)$$

❖ Energie potentielle du système

Le chariot étant en mouvement sur un rail horizontal, seul le pendule en mouvement possède une énergie potentielle. L'énergie potentielle du centre de gravité du pendule est donnée par :

$$E_p = m g L \cos(\theta) \quad (3.9)$$

L'équation de Lagrange :

D'après (3. 1) :

$$l = E_c - E_p \quad (3.10)$$

$$l = \frac{1}{2} (M+m)\dot{x}^2 + m L \cos(\theta) \dot{\theta} \dot{x} + \frac{1}{2} m L^2 \dot{\theta}^2 - m g L \cos(\theta) \quad (3.11)$$

L'équation d'Euler_lagrange :

$$\frac{d}{dt} \left(\frac{\partial l}{\partial \dot{q}_i} \right) - \left(\frac{\partial l}{\partial q} \right) = F_i \quad (3.12)$$

Avec :

l : Le lagrangien

∂q_i : La i éme composante de la vitesse généralisée.

∂q_i : La i éme composante du vecteur de coordonnées généralisées.

F_i : La force généralisée associée a q_i

$$\frac{d}{dt} \left(\frac{\partial l}{\partial \dot{x}} \right) - \left(\frac{\partial l}{\partial x} \right) = F \quad (3.13)$$

L'équation de Lagrange pour le degré de liberté $q(t) = x(t)$:

La dérivée partielle du lagrangien suivant \dot{x} et t s'écrit :

$$\frac{d}{dt} (M\dot{x} + m\dot{x} + m L \cos(\theta) \dot{\theta}) = F \quad (3.14)$$

D'où la première équation de Lagrange :

$$(M+m)\ddot{x} + mL \cos(\theta) \ddot{\theta} - mL \sin(\theta) \dot{\theta}^2 = F \quad (3.15)$$

L'équation de Lagrange pour le degré de liberté $q(t) = \theta(t)$:

$$\frac{d}{dt} \left(\frac{\partial L}{\partial \dot{\theta}} \right) - \left(\frac{\partial L}{\partial \theta} \right) = 0 \quad (3.16)$$

La dérivée partielle du lagrangien suivant $\dot{\theta}$ et θ s'écrit :

$$\frac{d}{dt} (mL \dot{x} \cos(\theta) + mL^2 \dot{\theta}) - (-mL \dot{x} \sin(\theta) \dot{\theta} + mgL \sin(\theta)) = 0 \quad (3.17)$$

Donc la deuxième équation de Lagrange est :

$$mL^2 \ddot{\theta} + mL \ddot{x} \cos(\theta) - mgL \sin(\theta) = 0 \quad (3.18)$$

Finalement, le modèle de connaissance du système chariot-pendule est donné par le système d'équations suivant :

$$\begin{cases} (M+m)\ddot{x} + mL \cos(\theta) \ddot{\theta} - mL \sin(\theta) \dot{\theta}^2 = F \\ mL^2 \ddot{\theta} + mL \ddot{x} \cos(\theta) - mgL \sin(\theta) = 0 \end{cases} \quad (3.19)$$

Ce modèle peut être écrit sous la forme de Gauchy (modèle d'état non linéaire).

Pour cela, on définit les variables d'état comme suit :

$x_1 = x$ (Position rectiligne du chariot).

$x_2 = \dot{x}$ (vitesse du chariot).

$x_3 = \theta$ (Position angulaire de la tige).

$x_4 = \dot{\theta}$ (vitesse angulaire de la tige).

Le modèle d'état non linéaire s'écrit sous la forme suivante :

$$\dot{x}(t) = f(x(t)) + g(x(t)) u(t) \quad (3.20)$$

$$\begin{pmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \end{pmatrix} = \begin{pmatrix} \frac{-mg \cos(x_3) \sin(x_2) + mLx_4^2 \sin(x_3)}{M + m \sin^2(x_3)} \\ x_4 \\ \frac{(M+m)g \sin(x_3) - mLx_4^2 \cos(x_3) \sin(x_3)}{(M+m \sin^2(x_3))L} \\ x_4 \end{pmatrix} + \begin{pmatrix} 0 \\ 1 \\ \frac{0}{M + m \sin^2(x_3)} \\ \frac{\cos(x_3)}{(M + m \sin^2(x_3))L} \end{pmatrix} F \quad (3.21)$$

Conclusion

Le pendule inversé est un système non linéaire et sous-actionné. Il est souvent utilisé pour tester l'efficacité de nouvelles commandes. C'est pour cette raison que nous l'avons choisi comme application, afin de mettre en évidence l'efficacité de notre stratégie de commande.

3.2.2 Étude et développement de régulateur

D'après les équations précédentes qu'on a trouvées à (3.17) [18]

$$\begin{cases} (M+m)\ddot{x} + mL \cos(\theta) \ddot{\theta} - mL \sin(\theta) \dot{\theta}^2 = F \\ mL^2 \ddot{\theta} + mL \ddot{x} \cos(\theta) - mgL \sin(\theta) = 0 \end{cases} \quad (3.22)$$

On a

$$(M+m)\ddot{x} + mL \cos(\theta) \ddot{\theta} - mL \sin(\theta) \dot{\theta}^2 = F \quad (3.23)$$

On obtient

$$\ddot{x} = \frac{F}{M+m} - \frac{mL}{M+m} (\dot{\theta}^2 \sin(\theta) - \ddot{\theta} \cos(\theta)) \quad (3.24)$$

Avec \ddot{x} est l'accélération du chariot, alors $\ddot{x} = a$

Et on a

$$mL^2 \ddot{\theta} + mL \ddot{x} \cos(\theta) - mgL \sin(\theta) = 0 \quad (3.25)$$

Après avoir ajouté les perturbations $f(t)$ à cette équation et remplacer \ddot{x} par a nous obtenons :

$$mL^2 \ddot{\theta} + mL a \cos(\theta) - mgL \sin(\theta) - L f = 0 \quad (3.26)$$

$$L \frac{d^2\theta(t)}{dt^2} = g \sin[\theta(t)] + L f(t) - a(t) \cos[\theta(t)] \quad (3.27)$$

Pour les petites valeurs de θ ($\theta \approx 0$) on peut faire les approximations suivante pour linéariser le système

$$\text{Donc} \quad \sin[\theta(t)] \approx \theta(t) \quad (3.28)$$

$$\text{Et} \quad \cos[\theta(t)] \approx 1 \quad (3.29)$$

Alors

$$L \frac{d^2\theta(t)}{dt^2} - g \theta(t) = L f(t) - a(t) \quad (3.30)$$

Après avoir appliqué la transformée de Laplace nous obtenons :

$$\theta(s) = \frac{1}{Ls^2 - g} [L f(s) - a(s)] \quad (3.31)$$

On a deux pole $p_1 = -\sqrt{g/L}$ et $p_2 = +\sqrt{g/L}$

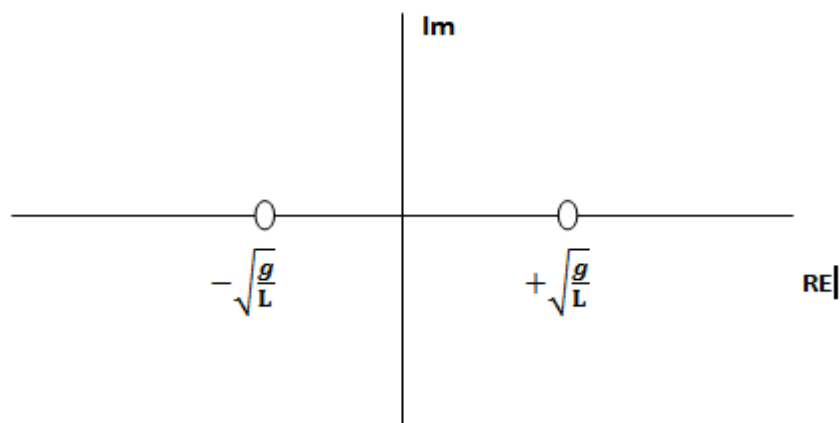


Figure 3.2 La représentation des pôles

Donc le système il peut être stable.

Schéma de principe et fonction système pour le système en boucle ouverte

$$\theta(s) = H(s) [L f(s) - a(s)] \quad (3.32)$$

$$H(s) = \frac{1}{Ls^2 - g} \quad (3.33)$$

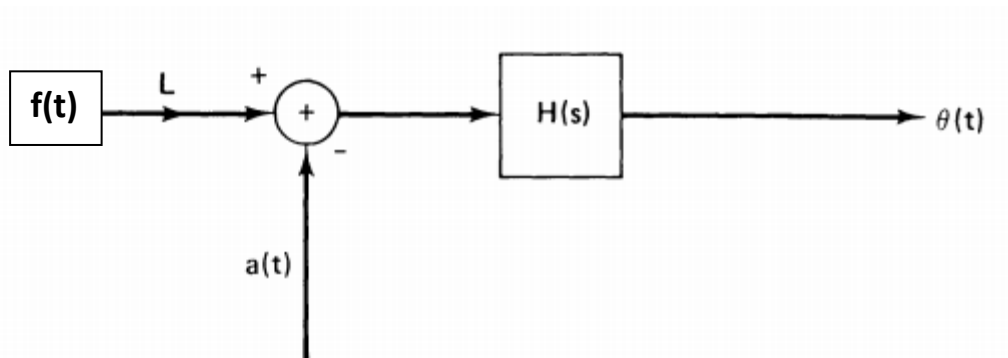


Figure 3.3 La représentation d'un système en boucle ouverte

❖ **Fonction système et schéma fonctionnel avec retour d'informations (feedback)**

$$\theta(s) = H(s) [L f(s) - a(s)] \quad (3.34)$$

$$H(s) = \frac{1}{Ls^2 - g} \quad (3.35)$$

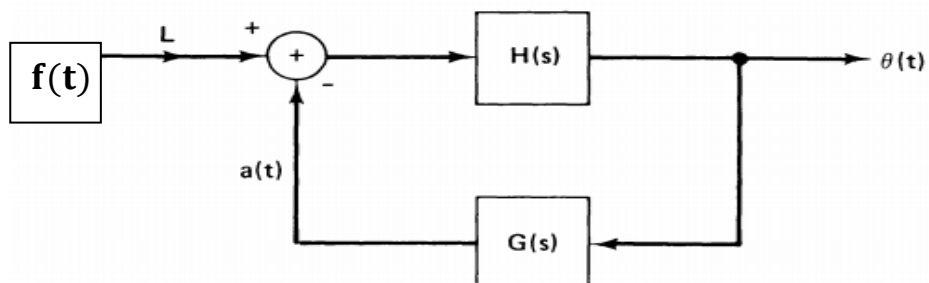


Figure 3.4 La représentation d'un svstème en boucle fermé

$$\theta(s) = \frac{LH(s)}{1+G(s)H(s)} f(s) \quad (3.36)$$

$$H(s) = \frac{1}{Ls^2 - g} \quad (3.37)$$

Proportionnelle feedback

$$a(t) = k_1 \theta(t) \quad (3.38)$$

$$G(s) = k_1 \quad (3.39)$$

$$\theta(s) = \frac{1}{s^2 - \left(\frac{g - k_1}{L}\right)} f(s) \quad (3.40)$$

pôles :

$$p = \pm \sqrt{\frac{g - k_1}{L}} \quad (3.41)$$

Lieu des pôles de fonction système pour feedback proportionnelle avec $k_1 < 0$

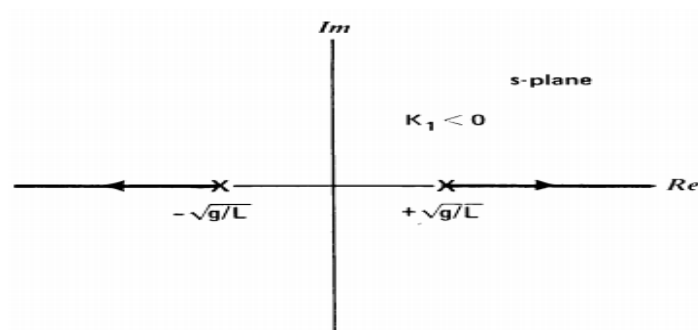


Figure 3.5 Représentation des pôles du système avec un retour proportionnel avec $k_1 < 0$

Lieu des pôles de fonction système pour feedback proportionnelle avec $k_1 < 0$

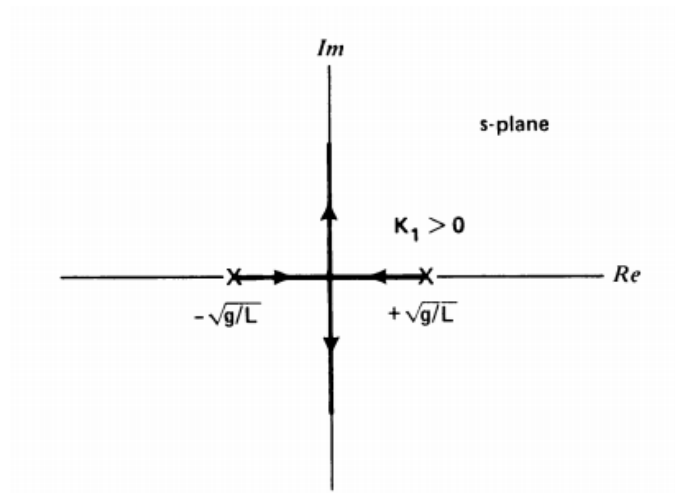


Figure 3.6 Représentation des pôles du système avec un retour proportionnel avec $k_1 > 0$

Fonction globale du système avec feedback dérivée

$$\theta(s) = \frac{L H(s)}{1 + G(s)H(s)} f(s) \quad (3.42)$$

$$H(s) = \frac{1}{Ls^2 - g} \quad (3.43)$$

Fonction globale du système avec feedback dérivée

$$a(t) = k_2 \frac{d\theta(t)}{dt} \quad (3.44)$$

$$G(s) = k_2 s \quad (3.45)$$

$$\theta(s) = \frac{1}{s^2 + s\left(\frac{k_2}{L}\right) - \frac{g}{L}} f(s) \quad (3.46)$$

Pôles :

$$s = -\frac{k_2}{2L} \pm \sqrt{\left(\frac{k_2}{2L}\right)^2 + (G/L)} \quad (3.45)$$

Lieu des pôles de fonctions du système pour la rétroaction dérivée avec $k_2 < 0$:

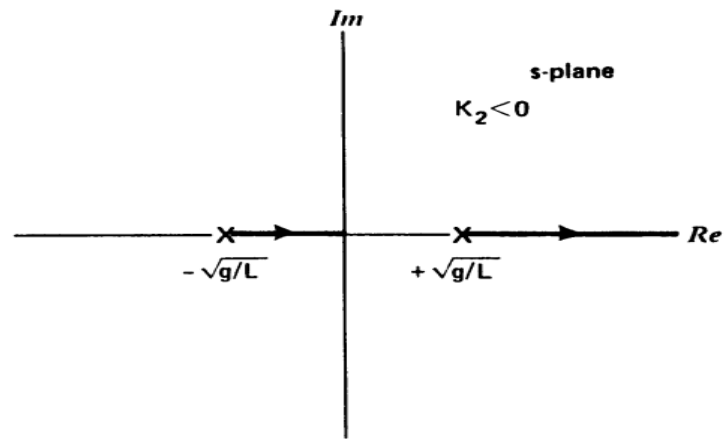


Figure 3.7 Représentation des pôles du système avec un retour dérivé avec $k_2 < 0$

lieu des pôles de fonctions du système pour la rétroaction dérivée avec $k_2 > 0$:

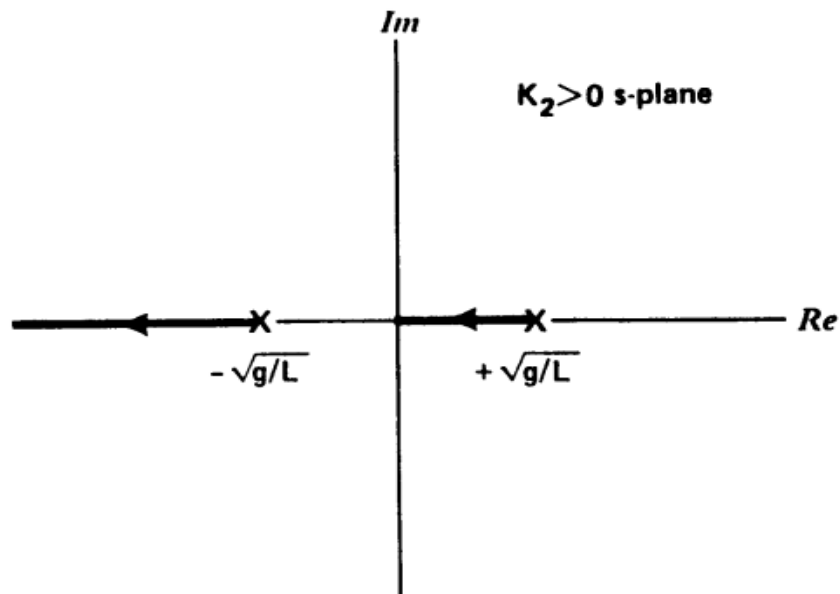


Figure 3.8 Représentation des pôles du svstème avec un retour dérivé avec $k_2 > 0$

Fonction globale du système avec feedback proportionnelle et dérivée

$$a(t) = k_1 \theta(t) + k_2 \frac{d\theta(t)}{dt} \quad (3.46)$$

$$G(s) = k_1 + k_2 s \quad (3.47)$$

$$H(s) = \frac{1}{s^2 + s\left(\frac{k_2}{L}\right) - \frac{g}{L} + k_1/L} \quad (3.48)$$

Pôles :

$$s = -\frac{k_2}{2L} \pm \sqrt{\left(\frac{k_2}{2L}\right)^2 - \left(\frac{k_1 - g}{L}\right)} \quad (3.49)$$

Lieu des pôles de fonction système pour rétroaction proportionnelle et dérivée

Avec $k_1 > 0$ et $k_2 > 0$:

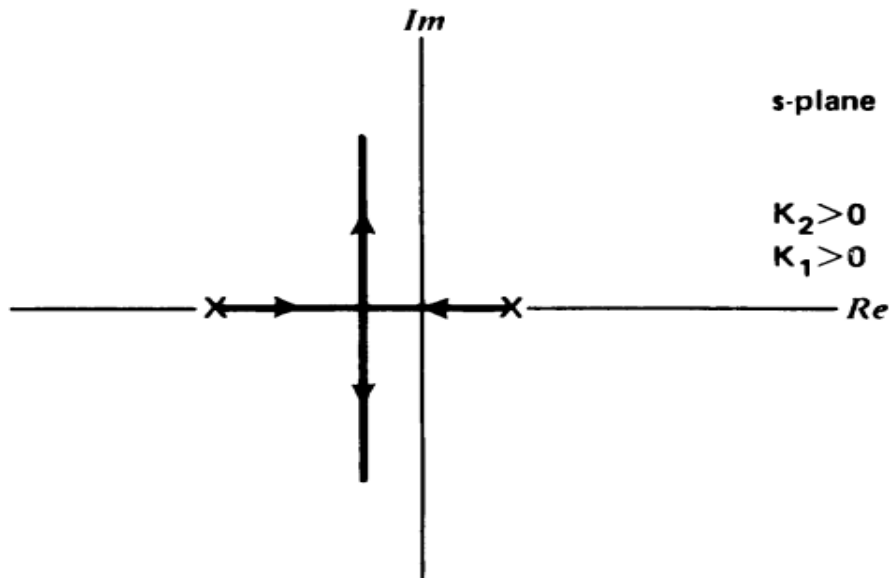


Figure 3.9 Représentation des pôles du système avec un retour dérivée et proportionnel avec $k_1 > 0$ et $K_2 > 0$

D'après les résultats obtenus le pendule inversé avec rétroaction proportionnelle et dérivés fonctionnent comme un système stable

3.2.3 Les coefficients du régulateur PID

D'après les résultats obtenue précédemment, on a conclue que le système est peut être un système stable avec certain conditions, qui représentes les coefficients k_1 et k_2 . Dans cette phase on travaille pour limité les intervalles de ces coefficients.

pour un système stable il faut que la partie réel des pôles soit négatif.

Dans notre système on utilise une rétroaction proportionnelle et dérivée, et les pôles obtenus est :

$$s = -\frac{k_2}{2L} \pm \sqrt{\left(\frac{k_2}{2L}\right)^2 - \left(\frac{k_1-g}{L}\right)} \quad (3.50)$$

avec des conditions initiales $k_1 > 0$ et $k_2 > 0$.

pour avoir des pôles qui ont la partie réelle négative il faut que soit :

- $\left(\frac{k_2}{2L}\right)^2 - \left(\frac{k_1-g}{L}\right) < 0$ pour avoir $s = -\frac{k_2}{2L} \pm i\sqrt{\left(\frac{k_2}{2L}\right)^2 - \left(\frac{k_1-g}{L}\right)}$ et la partie réelle $= -\frac{k_2}{2L}$.
- sinon $-\frac{k_2}{2L} \pm \sqrt{\left(\frac{k_2}{2L}\right)^2 - \left(\frac{k_1-g}{L}\right)} < 0$.

➤ on va commencer par le premier cas

$$\left(\frac{k_2}{2L}\right)^2 - \left(\frac{k_1-g}{L}\right) < 0 \text{ pour avoir } s = -\frac{k_2}{2L} \pm i\sqrt{\left(\frac{k_2}{2L}\right)^2 - \left(\frac{k_1-g}{L}\right)}$$

$$\left(\frac{k_2}{2L}\right)^2 < \frac{k_1-g}{L}$$

on obtient $k_2 < \sqrt{4 * L(k_1 - g)}$ avec $g = 9.81$ et $L = 0.2m$

alors il faut $k_1 > g$ et $0 < k_2 < \sqrt{4 * L(k_1 - g)}$ pour avoir des pôles avec une partie réelle négative

➤ ensuite le deuxième cas

$$-\frac{k_2}{2L} \pm \sqrt{\left(\frac{k_2}{2L}\right)^2 - \left(\frac{k_1-g}{L}\right)} < 0 \quad (3.51)$$

on vas étudié juste ce cas $-\frac{k_2}{2L} + \sqrt{\left(\frac{k_2}{2L}\right)^2 - \left(\frac{k_1-g}{L}\right)} < 0$

il faut Just que $\left(\frac{k_2}{2L}\right)^2 > \frac{k_1-g}{L} > 0$ donc $k_1 > g$ et $k_2 > \sqrt{4 * L(k_1 - g)}$ pour avoir des pôles négatif avec une partie imaginaire nulle .

3.3 Auto-équilibrage

3.3.1 PID

L'élément le plus important de tout robot est le contrôleur. Surtout pour un robot d'auto-équilibrage, le programme de contrôle est vital, car il interprète les données de capteurs et décide combien les moteurs doivent être déplacés pour que le robot reste debout. Le contrôleur le plus couramment utilisé pour les systèmes de stabilisation est le régulateur PID. Alors regardons comment cela fonctionne[7](voir la figure3.10):

Le contrôleur PID

PID signifie proportionnelle, intégrale et dérivée, en se référant aux équations mathématiques utilisées pour calculer la sortie.

Le P-coefficient prend simplement dans l'angle actuel du robot et rend les moteurs se déplacent dans la même direction que le robot est en baisse. Si le P-composant est utilisé seul, le robot pourrait se stabiliser pendant un certain temps, mais le système aura tendance à dépasser, osciller et finalement tomber.

I-coefficient est utilisé pour accumuler les erreurs. Par exemple, si le robot a tendance à tomber sur un côté, il sait qu'il a besoin de se déplacer dans la direction opposée, afin de maintenir la sur la cible et pour prévenir la dérive vers la gauche ou la droite.

Enfin, le coefficient D est chargé d'amortir toute oscillation et assure que le robot ne vibre pas trop. Il agit tout simplement contre tout mouvement.

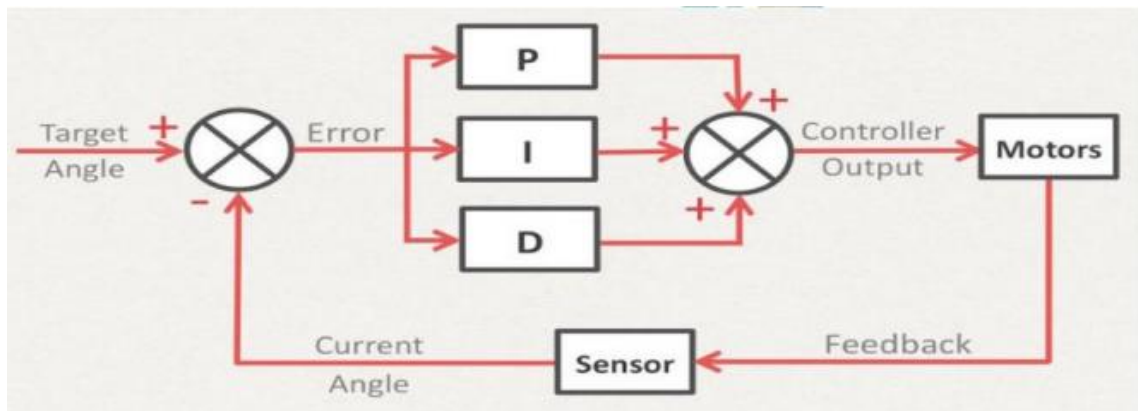


Figure 3.10 Schéma du contrôleur PID [7].

3.3.2 Calibrage contrôleur PID

- 1) Créer une certaine manière dans laquelle vous pouvez modifier la constante PID de votre robot pendant qu'il est en cours d'exécution. Une option est d'utiliser un potentiomètre ou une autre entrée analogique pour être en mesure d'augmenter ou de diminuer la constante PID. J'ai personnellement utilisé la connexion USB pour envoyer de nouvelles valeurs PID. Ceci est important car vous pouvez alors voir immédiatement le fonctionnement des nouvelles valeurs PID, et vous ne devrez pas re-télécharger plusieurs fois !
- 2) Pour commencer mettez toutes les constantes PID à zéro.
- 3) Lentement augmenter la valeur de P-constant. Alors que vous faites cela, maintenez le robot pour vous assurer qu'il ne tombe pas, Vous devez augmenter la P-constante jusqu'à ce que le robot répond rapidement à toute inclinaison, puis juste rendre le dépassement du robot dans l'autre sens.
- 4) Maintenant augmenter la I-constante. Ce coefficient est un peu difficile à trouver à droite. Vous devez garder cela relativement faible, car il peut accumuler des erreurs très rapidement.
- 5) Soulever le D-constant. Beaucoup. Les coefficient dérivés fonctionnent contre tout mouvement, de sorte qu'il contribue à amortir des oscillations et de réduire les

Dépassements Passez de nombreuses heures infructueuses modifiant légèrement les valeurs PID. Ceci est probablement la partie la plus longue de la procédure, car il n'y a pas beaucoup d'une méthode à elle. Vous avez juste à augmenter et diminuer les valeurs jusqu'à ce que vous atteignez le contrôle parfait pour votre robot !

3.4 Le branchement des outils

Dans notre robot on a besoin de (voir la figure 3.11) :

- deux moteur a courant continue 3-12v (partie actionneur) .
- deux driver moteur (pré-actionneur).
- une carte myrio-1900 (la partie commande) .
- un capteur mpu9250 .

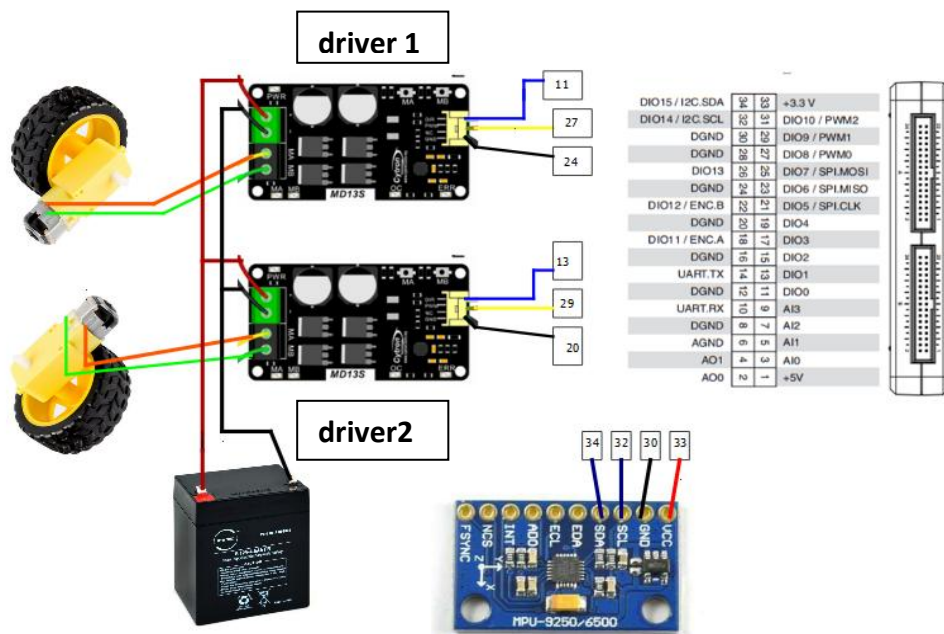


Figure 3.11 Le schéma électrique du système.

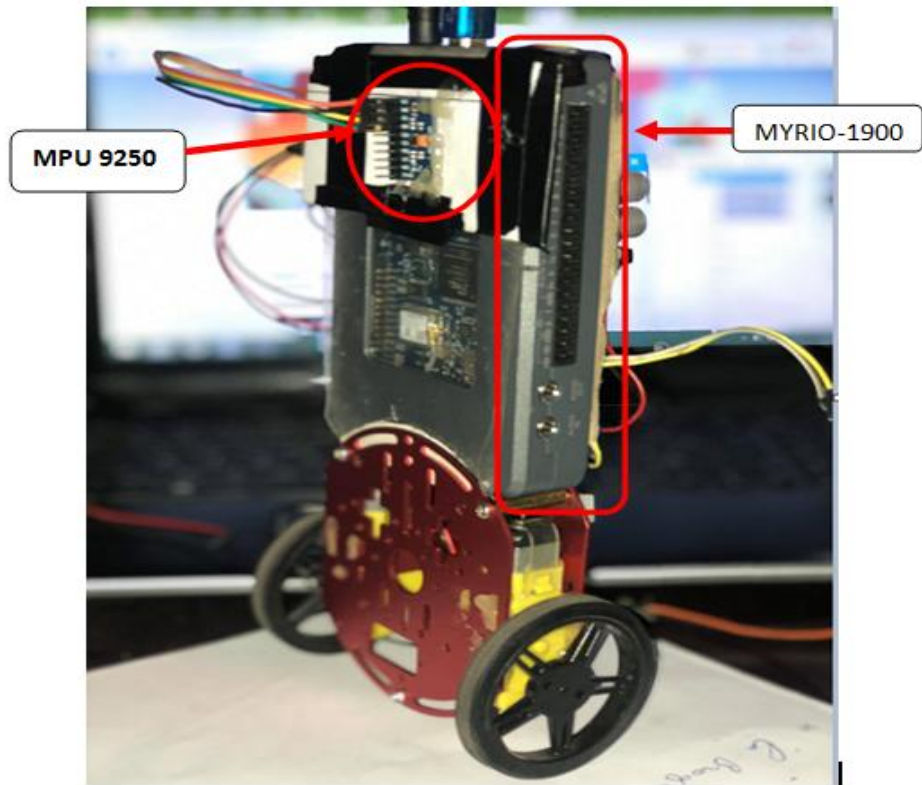


Figure 3.12 L'avant du robot.

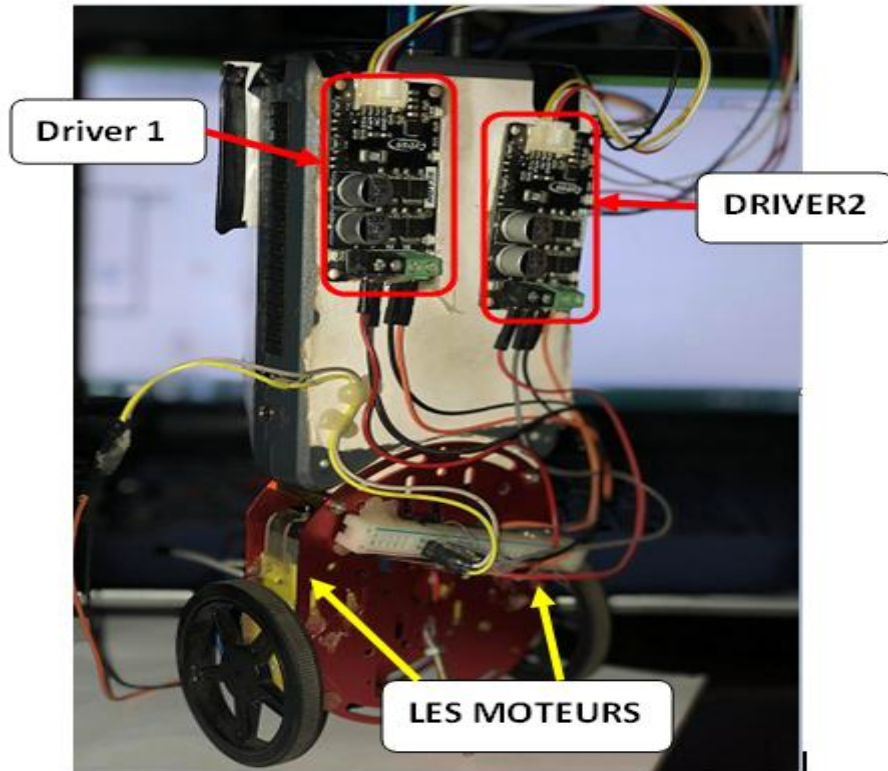


Figure 3.13 L'arrière du robot .

Les pins du driver 1	branchement avec la carte myrio-1900
PWM 1	PIN 27(PWM 0_A)
DIR 1	PIN 11(DIO 0_A)
GND 1	PIN 24(D_GND_A)

Tableau 9 Le branchement du driver 1 avec myrio-1900.

Les pins du driver 2	branchement avec la carte myrio-1900
PWM 2	PIN 29(PWM 1_A)
DIR 2	PIN 13(DIO 1_A)
GND 2	PIN 20(D_GND_A)

Tableau 10 Le branchement du driver 2 avec myrio-1900.

Les pins du capteur mpu9250	branchement avec la carte myrio-1900
SCL	PIN 32(I2C.SCL_A)
SDA	PIN 34(I2C.SDA_A)
VCC	PIN 33(3.3V_A)
GND	PIN 30(D-GND_A)

Tableau 11 Le branchement du capteur mpu9250 avec myrio-1900.

/	Driver 1	Driver 2
Moteur 1	MA-MB(1)	/
Moteur 2	/	MA-MB(2)
Batterie	VCC-GND	VCC-GND

Tableau 12 Le branchement des drivers avec les moteurs et la batterie

3.5 La programmation

Les systèmes embarqués sont composés d'une partie logiciel « Software » et une partie « Hardware ». Dans cette partie, nous allons discuter de la partie software (à base de LAVIEW). En outre, nous allons donner les différentes solutions aux contraintes rencontrées au cours de ce projet.

Pour que le robot reste en équilibré, 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 (MPU9250), qui fournit à la fois une accélération et une rotation dans les trois axes (bases du MPU9250 I2C). Le MYRIO-1900 utilise ce signal pour connaître l'orientation actuelle du robot. L'élément correcteur est la combinaison moteur / roue.

3.5.1 Le fonctionnement de mpu 9250

Pour mesurer l'inclinaison et l'accélération, on a utiliser le MPU92-50. 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 MYRIO-1900 via un bus de communication I2C.

Bus I2C :

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.

Pour l'instant en n'a qu'un capteur connecté en I2C à la carte MYRIO-1900 (comme on a vue dans la partie (3.4) branchement des outis).

maintenant pour lire les valeur on utilise une bibliothèque I2C dans logiciel LABVIEW (voire la figure 3.14):

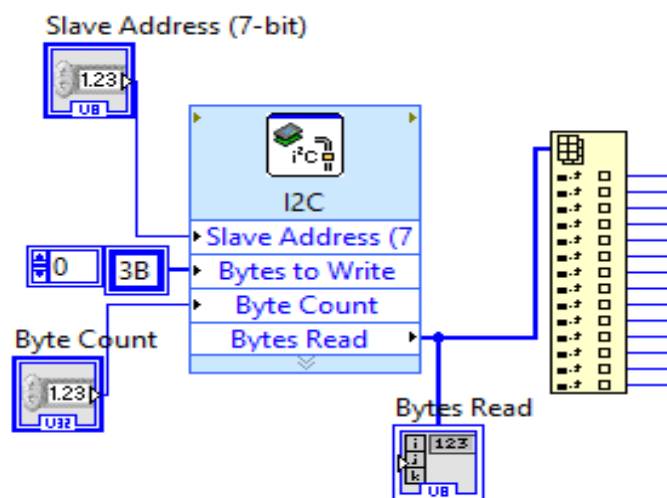


Figure3.14 Utilisation de la bibliothèque i2c

- **slave adress** : l'adresse par défaut d'un gyro MPU-6050 est **0x68**(en hexadécimale) ,

a partir de cette adresse le MYRIO-1900(maitre) peut communiquer avec le mpu92-50(esclave).

Le MYRIO 1900 écrira (write) pour envoyer des données au gyroscope, ou lira (read) pour en recevoir des données.

- **bytes to write** : Spécifie les octets des données que ce VI Express écrive sur le périphérique esclave I2C ,on a commencer par la lecture de l'adresse **0x3B** (voire la figure).

Addr (Hex)	Addr (Dec.)	Register Name
3B	59	ACCEL_XOUT_H
3C	60	ACCEL_XOUT_L
3D	61	ACCEL_YOUT_H
3E	62	ACCEL_YOUT_L
3F	63	ACCEL_ZOUT_H
40	64	ACCEL_ZOUT_L
41	65	TEMP_OUT_H
42	66	TEMP_OUT_L
43	67	GYRO_XOUT_H
44	68	GYRO_XOUT_L
45	69	GYRO_YOUT_H
46	70	GYRO_YOUT_L
47	71	GYRO_ZOUT_H
48	72	GYRO_ZOUT_L

Figure 3.15 L'adressage des données

- **byte count**: Spécifie le nombre d'octets que ce VI Express écrive sur le périphérique esclave I2C (on a donner une valeur 14 car on a besoins de lires 14 octets pour avoir les valeurs de l'accéléromètre(x,y,z) et le gyroscope(x,y,z) .
- **bytes read** : Renvoie les octets de données que ce VI Express lit à partir du périphérique esclave I2C(voire la figure 3.15).

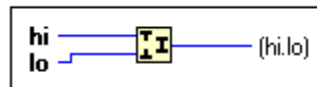
3.5.2 La lecture du capteur mpu 9250

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

AcX , AcY , AcZ , Tmp , GyX , GyY , GyZ ; définit que les variables mesurées sont codés sur 16 bits et sont signées (peuvent être négatives). Elles peuvent donc prendre une valeur allant de -32768 à 32767

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. 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 «**Join Numbers Function**» .



hi :peut être un nombre de 8, 16 ou 32 bits ou un tableau ou un cluster de cette représentation

lo: peut être un nombre de 8, 16 ou 32 bits ou un tableau ou un cluster de cette représentation

(hi.lo) :est un entier deux fois la largeur de **hi** et **lo** .

- pour convertir la valeur qu'on a lit en nombre entier on utilise «**To Word Integer Function**» .



le résultat obtenue peuvent prendre une valeur x allant de -32 768 à 32 767 donc on fait conversion pour avoir entre -1 et +1 .

l'équation utiliser de la conversion est :

il faut connais la valeur de H et L ,et bien sur x

nouveau $x = (x-L)/16000+L/16000$, alors avec cette méthode on a trouvé avec des donnés entre -1 et +1 (voir les figures 3.16 et 3.17).

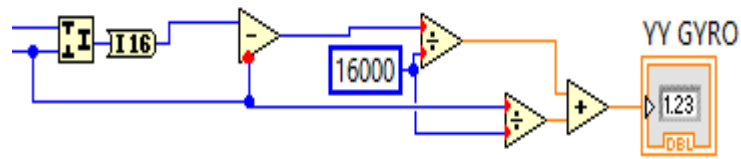


Figure 3.16 L'application de l'équation sur le programme

XX ACC	YY ACC	ZZ ACC	XX GYRO	YY GYRO	ZZ GYRO
0,03975	0,85625	0,56875	-0,0169E	-0,00437	-0,0106E

Figure3.17 L'affichage des valeurs de gyroscope.

3.5.3 Utilisation de l'accéléromètre :

d'après la lecture des données a partir du capteur mpu9250, on obtient les valeurs de l'accélération des axes X, Y et Z , et l'état désirer du robot c'est d'être vertical (voire la figure 3.18).

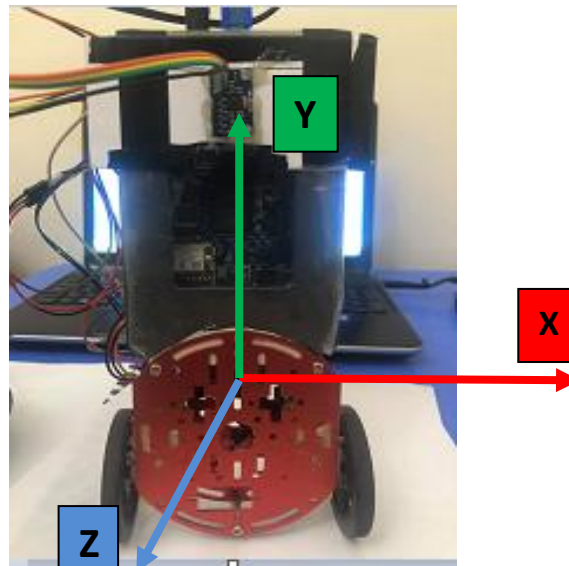


Figure 3.18 Le robot

alors d'après le positionnement du capteur mpu92-50 (l'orientation des axes X,Y et Z) , les valeurs du x ,y et z fournissent l'accélération nominal de la tombé ,

donc à l'état désirer il faut que $Y=1$, $X=0$ et $Z=0$.

si ($Y \neq 0$) alors le robot n'est pas verticale et automatiquement l'axe Z devient supérieur ou inférieur à 0 :

si $Z > 0$ alors le robot est incliné en avant (il faut que les moteur entraînent en avant, pour avoir la bonne position du robot) .

si $Z < 0$ alors le robot est incliné en arrière (il faut que les moteur entraînent en arrière , pour avoir la bonne position du robot).

pour avoir l'angle actuel du robot (la bonne position est 0°), il faut calculer l'arctangente du Z/Y ($\arctan(Z/Y)$) et on obtient l'angle de l'inclinaison (voir les figures 3.19 et 3.20) .

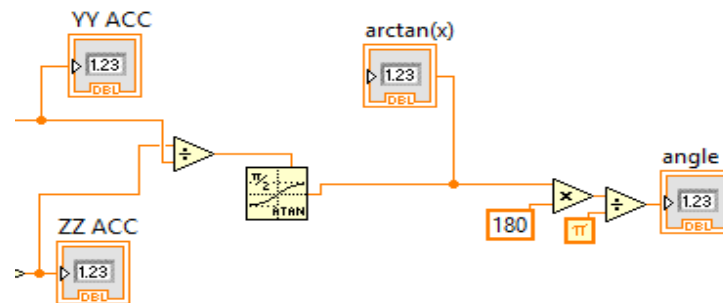


Figure 3.19 Exécution dans le Block Diagram

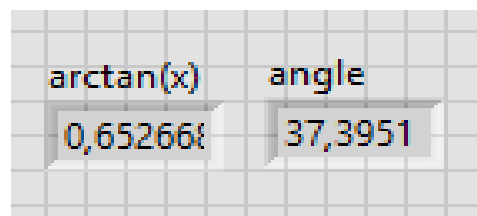


Figure3.20 L'affichage dans le Front panel

3.5.4 Utilisation de gyroscope

on a besoins de la valeur de **gyro(X)** pour déterminer la vitesse angulaire du pendule ,et à partir de la vitesse angulaire ,on peut déterminer l'accélération angulaire par une dérivée, et on peut trouvé l'angle de la variation à partir de l'intégrale de la vitesse angulaire .

3.5.5 Utilisation des intégrateurs et dérivateurs

avant tout , pour pouvoir utiliser les intégrateurs et les dérivateurs il faut utiliser une boucle qui s'appelle **control and simulation loop** , (voir la figure 3.21).

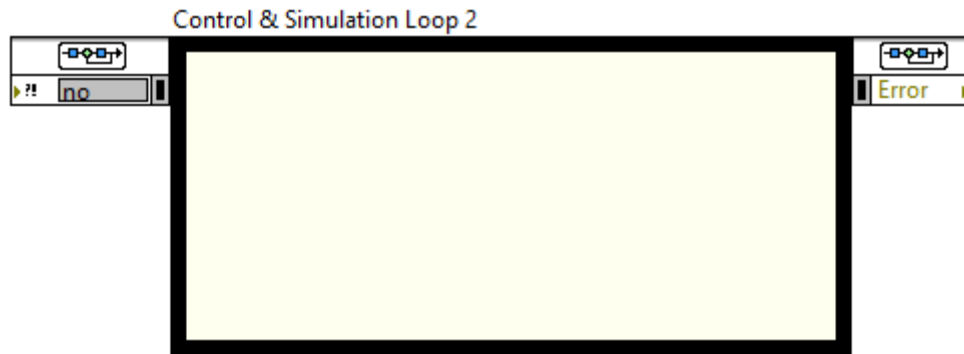


Figure 3.21 La boucle control and simulation .

- **dérivateur** : il est très simple(voir la figure 3.22) ,

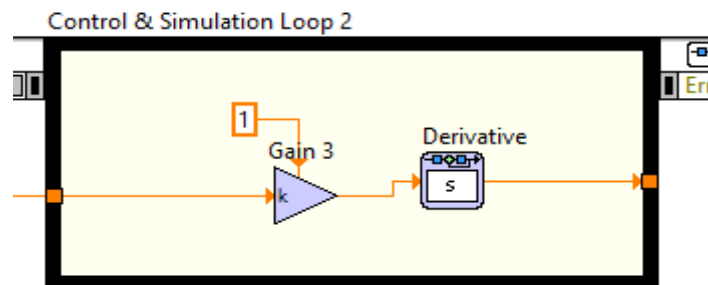


Figure 3.22 Utilisation de dérivateur.

On a :

input : c'est l'entrée du module .

output : c'est la sortie .

Gain : c'est le coefficient.

- **intégrateur** : il est presque comme le dérivateur (voir la figure 3.23) ,

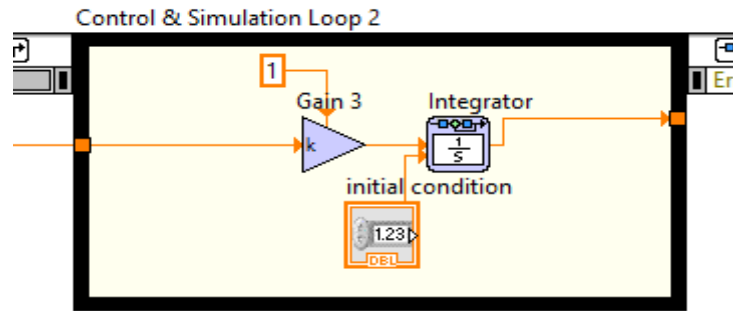


Figure 3.23 Utilisation de l' intégrateur

Input: c'est l'entrée du module.

Output : la sortie du module

Gain : c'est le coefficient.

condition initiale: est la valeur initial de l'entrée .

3.5.6 Le signal de commande

pour produire le signal de commande, il faut d'abord trouver l'erreur et on utilise un correcteur PID pour trouver la commande(voir la figure 3.24) .

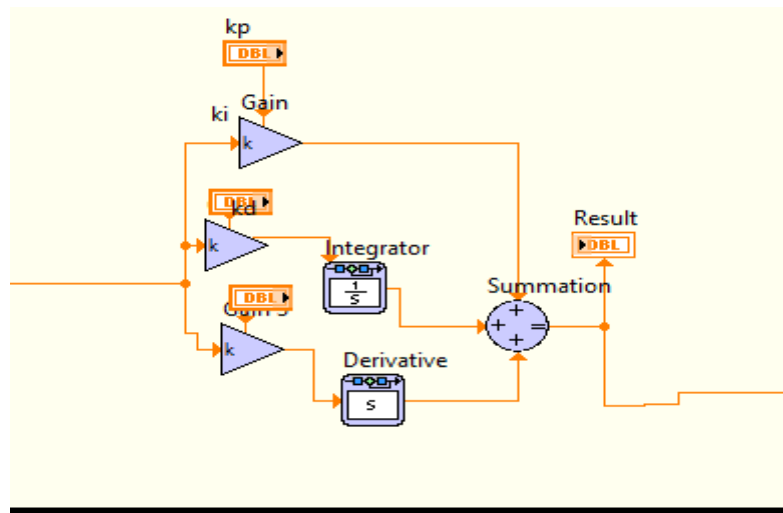


Figure 3.24 Le régulateur pid .

avec k_p, k_i, k_d des coefficients du régulateur, et pour les valeurs de chaque coefficient on a donner les intervalle de chaque une pour le système être stable .

la sortie du correcteur PID , elle devient le signal PWM après le calibrage , car le signal PWM prend des valeurs entre 0 et 1 ,dans notre cas après la fixation des coefficients

du régulateur , on fait un test de la plus grande variation de l'angle (l'accélération maximal) et on vas lire la sortie du PID , après la lecture de la sortie (la valeur maximal), alors on fait la division de la sortie du PID sur sa valeur maximale pour trouver un signal entre -1 et 1 .

le signal obtenue est le PWM , mais le PWM prend des valeurs juste entre 0 et 1,et notre signal prend des valeur entre -1 et +1 , et but du signe est de connaitre les direction d'entrainement des moteurs .

alors on fait la valeur absolu de signal et on obtient le signal PWM .

3.5.7 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, ce signal utilisé pour contrôler un moteur.

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 MD13S.Ce circuit offre un moyen de piloter un moteur CC.

pour lancer la commande a ce circuit , on a utilisé une bibliothèque PWM (on a besoin de deux bibliothèque car on a deux moteur) sur LABVIEW (voire la figure 3.25) . :

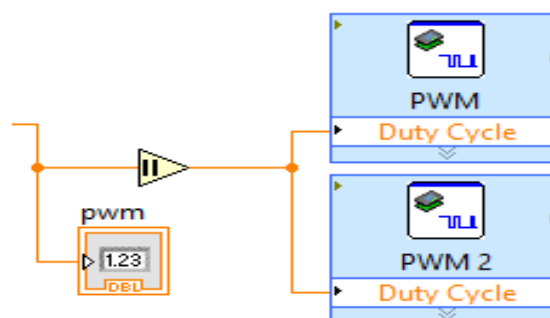


Figure3.25 La bibliothèque PWM sur LABVIEW.

Duty Cycle : est l'entrée du signal PWM .

la configuration de la bibliothèque est de donner le numéro du port et la fréquence de signal (voir la figure 3.26):

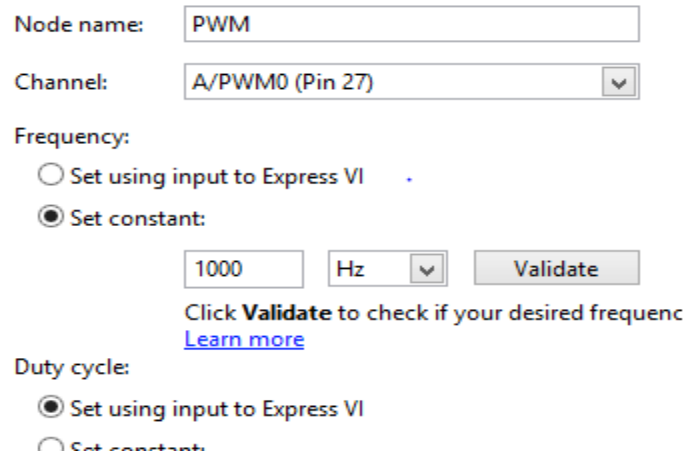


Figure 3.26 Configuration de la bibliothèque.

3.5.8 La direction de la Rotation des Moteur

pour contrôler la direction de la rotation des moteur à courant continu, il suffit juste d'inversé les pôles ,et le driver MD13S c'est lui qui est responsable de cette tache .

et pour lancer la commande a le circuit MD13S , on utilise une bibliothèque DIO sur LABVIEW (on a besoin de deux bibliothèque pour contrôler la direction de rotation de chaque moteur)(voire la figure 3.27).

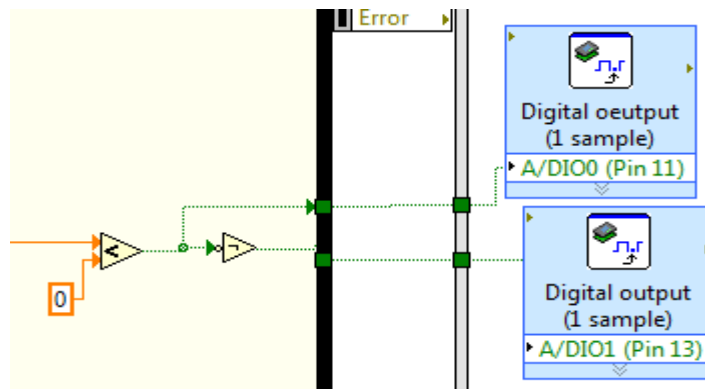


Figure3.27 La bibliothèque DIO sur LABVIEW.

la configuration de bibliothèque DIO est juste de donner le nom du port utiliser(figure3.28) :

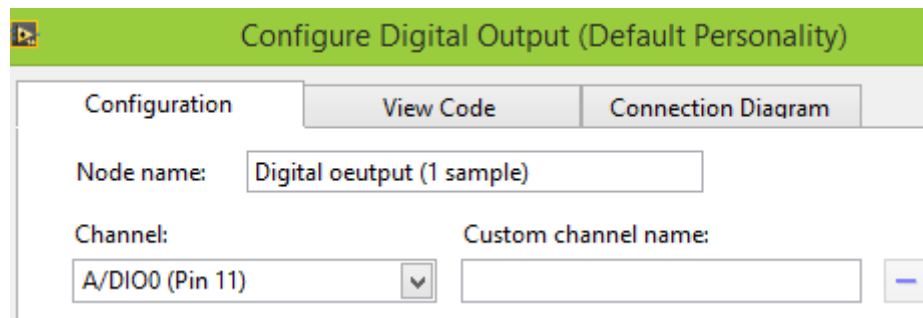


Figure 3.28 La configuration de la bibliothèque DIO.

pour la direction de la rotation des moteurs , on donne un signal différent pour avoir la marche en avant ou en arrière :

pour le robot marche en avant : donc le PWM >0 alors on fait le test si le PWM >0

on donne a **DIO0" HIGH "** et **DIO1" LOW "** , car la fixation des moteurs dans le châssis est différente

3.5.9 Flat Sequence Structure

après l'explication de tous les bibliothèques (utilisation , la configuration) , et pour avoir une bonne exécution du programme (l'ordre de l'exécution) on a utiliser une structure de séquence ,elle Se compose d'un ou plusieurs sous-diagrammes, ou cadres, qui s'exécutent séquentiellement. Utilisez la **structure Flat Sequence** pour vous assurer qu'un sous-diagramme s'exécute avant ou après un autre sous- diagramme .

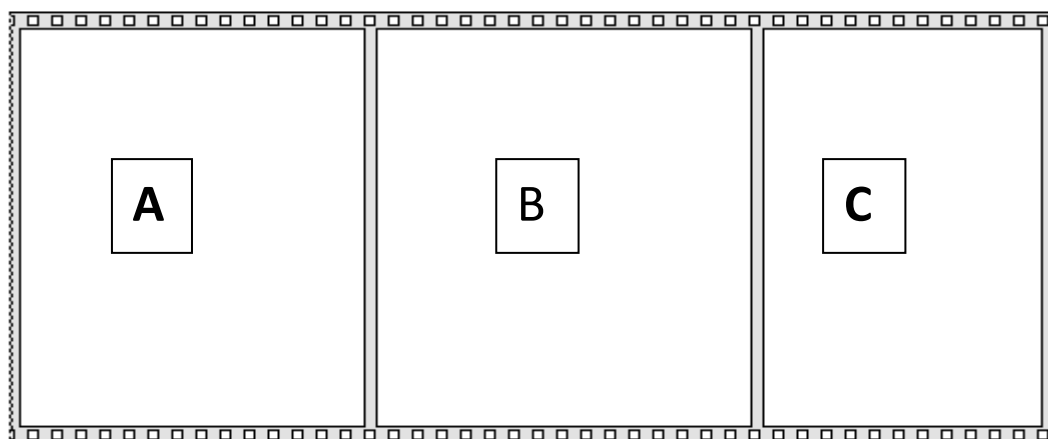


Figure 3.29 La structure flat sequence utilisé dans le programme.

dans notre programme on a utiliser une **structure Flat Sequence** de 3 parties(voir la figure 3.29) :

la partie A : c'est pour la lecture des données.

la partie B : c'est pour le traitement des données .

la partie C : c'est pour l'exécution de la commande .

3.6 Conclusion

Nous avons présenté dans ce chapitre les étapes de la modélisation du système chariot-pendule à deux degrés de liberté, ainsi nous avons cité l'étude et le développement du régulateur PID pour la stabilité du système. Dans ce chapitre nous avons également relevé toutes les utilisations et le mode de fonctionnement des bibliothèques utilisés dans la programmation en détail.

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.

Pour l'obtention de l'angle d'inclinaison nous avons intégré la valeur obtenue par le gyromètre ce qui nous a permis d'avoir une estimation relativement correcte.

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.

Bibliographie

- [1]: Robotique. In: Wikipédia [Internet]. 2020 [cité 10 sept 2020]. Disponible sur: <https://fr.wikipedia.org/w/index.php?title=Robotique&oldid=174491207>
- [2]: Historique de la robotique [Internet]. [cité 10 sept 2020]. Disponible sur: <http://www-igm.univ-mlv.fr/~dr/XPOSE2002/robotique/chapitres/Historique.htm>
- [3]: https://eduscol.education.fr/sti/sites/eduscol.education.fr.sti/files/ressources/techniques/5875/5875-la-robotique-industrielle-eduscol-2015_0.pdf
- [4]: ISN - Robotique - Constituants d'un robot [Internet]. [cité 10 sept 2020]. Disponible sur: http://silanus.fr/sin/formationISN/Parcours/Robotique/co/module_Robotique_9.html
- [5]: 4-Robot.pdf [Internet]. [cité 10 sept 2020]. Disponible sur: <https://eureka.uqam.ca/wp-content/uploads/sites/4/2018/05/4-Robot.pdf>
- [6]: Memoire Online - Conception et réalisation d'un robot mobile à base d'arduino - Hocine TAKHI [Internet]. Memoire Online. [cité 10 sept 2020]. Disponible sur: https://www.memoireonline.com/01/16/9368/m_Conception-et-realisation-d-un-robot-mobile--base-d-arduino4.html
- [7]: [lectures/lecture-26-feedback-example-the-inverted-pendulum/MITRES_6_007S11_lec26.pdfhttps://www.academia.edu/26268640/Projet_Self_balancing_robot](https://www.academia.edu/26268640/Projet_Self_balancing_robot)
- [8] : D. Choi and J.H. Oh. Human-friendly motion control of a wheeled inverted pendulum by reduced order disturbance observer. In IEEE International Conference on Robotics and Automation, Pasadena, CA, USA, 2008.
- [9] : A. Djoudi , T. Benabdallah, "synthèse et Application des lois de commande non linéaires sur un pendule inverse", Mémoire d'Ingénieur , Ecole Nationale Polytechnique, 2007.
- [10] : Jong H. Park and Kyoung, "Biped robot walking using gravity-compensated inverted pendulum mode and computed torque control ". International Conference on Robotics and Automation, Belgium, 1998.
- [11]: myRIO-1900 User Guide and Specifications - National Instruments. :32.
- [12] : Définition - Micromoteurs mdp motor [Internet]. [cité 10 sept 2020]. Disponible sur: <https://www.mdp.fr/documentation/lexique/courant-continu/definition.html>
- [13]: Board Layout - Cytron Technologies MD13S User Manual [Page 5] | ManualsLib [Internet]. [cité 10 sept 2020]. Disponible sur: <https://www.manualslib.com/manual/1447479/Cytron-Technologies-Md13s.html?page=5#manual>
- [14] : MPU-9250 - Invensense | DigiKey [Internet]. [cité 10 sept 2020]. Disponible sur: <https://www.digikey.fr/fr/product-highlight/i/invensense/mpu-9250-9-axis-gyro-accel->

magnet

[15] :<http://www.cim.mcgill.ca/~jer/courses/es/labs/lab1.pdf>

[16] : Bugeja M. « Non linear swing-up and stabilizing control of an inverted pendulum system ».EuroconLjubljana, Slovenia.2003

[17]: Lam J. “Control of an inverted pendulum”. Dept of Electrical &computer Engineering, University of California. U.S.A. 2004

[18]: <https://ocw.mit.edu/resources/res-6-007-signals-and-systems-spring-2011/video->